



Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE3.1.2H

Framework and Tools Specifications (Protection and Accounting Tools)

Version: 2.7

Date: 07/03/2005

Responsible: DSI

Project Number: IST-2-511299

Project Title: AXMEDIS

Deliverable Type: report

Visible to User Groups: NO

Visible to Affiliated: NO

Visible to the Public: NO

Deliverable Number: DE3.1.2 part H

Contractual Date of Delivery: January 2005

Actual Date of Delivery: 15 March 2005

Title of Deliverable: Document

Work-Package contributing to the Deliverable: WP3.1

Task contributing to the Deliverable: WP3, WP2

Nature of the Deliverable: report

Author(s): DSI, EXITECH, FUPF

Abstract: this part includes the major components for the protection system of AXMEDIS: AXMEDIS certifier and Supervisor, AXMEDIS Protection Manager Support, Protection editor, Accounting manager and collector, etc. Other protection tools are in Part A and B.

Keyword List: Protection, Digital Rights Management, certification, authentication, verification, black lists of objects, licensing, etc.

AXMEDIS Copyright Notice

The following terms (including future possible amendments) set out the rights and obligations licensee will be requested to accept on entering into possession of any official AXMEDIS document either by downloading it from the web site or by any other means.

Any relevant AXMEDIS document includes this license. PLEASE READ THE FOLLOWING TERMS CAREFULLY AS THEY HAVE TO BE ACCEPTED PRIOR TO READING/USE OF THE DOCUMENT.

1. DEFINITIONS

- i. **"Acceptance Date"** is the date on which these terms and conditions for entering into possession of the document have been accepted.
- ii. **"Copyright"** stands for any content, document or portion of it that is covered by the copyright disclaimer in a Document.
- iii. **"Licensor"** is AXMEDIS Consortium as a de-facto consortium of the EC project and any of its derivations in terms of companies and/or associations, see www.axmedis.org
- iv. **"Document"** means the information contained in any electronic file, which has been published by the Licensor's as AXMEDIS official document and listed in the web site mentioned above or available by any other means.
- v. **"Works"** means any works created by the licensee, which reproduce a Document or any of its part.

2. LICENCE

1. The Licensor grants a non-exclusive royalty free licence to reproduce and use the Documents subject to present terms and conditions (the **Licence**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.
2. In consideration of the Licensor granting the Licence, licensee agrees to adhere to the following terms and conditions.

3. TERM AND TERMINATION

1. Granted Licence shall commence on Acceptance Date.
2. Granted Licence will terminate automatically if licensee fails to comply with any of the terms and conditions of this Licence.
3. Termination of this Licence does not affect either party's accrued rights and obligations as at the date of termination.
4. Upon termination of this Licence for whatever reason, licensee shall cease to make any use of the accessed Copyright.
5. All provisions of this Licence, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this Licence and shall continue in full force and effect.
6. Notwithstanding License termination, confidentiality clauses related to any content, document or part of it as stated in the document itself will remain in force for a period of 5 years after license issue date or the period stated in the document whichever is the longer.

4. USE

1. Licensee shall not breach or denigrate the integrity of the Copyright Notice and in particular shall not:
 - i. remove this Copyright Notice on a Document or any of its reproduction in any form in which those may be achieved;
 - ii. change or remove the title of a Document;
 - iii. use all or any part of a Document as part of a specification or standard not emanating from the Licensor without the prior written consent of the Licensor; or
 - iv. do or permit others to do any act or omission in relation to a Document which is contrary to the rights and obligations as stated in the present license and agreed with the Licensor

5. COPYRIGHT NOTICES

1. All Works shall bear a clear notice asserting the Licensor's Copyright. The notice shall use the wording employed by the Licensor in its own copyright notice unless the Licensor otherwise instructs licensees.

6. WARRANTY

1. The Licensor warrants the licensee that the present licence is issued on the basis of full Copyright ownership or re-licensing agreements granting the Licensor full licensing and enforcement power.
2. For the avoidance of doubt the licensee should be aware that although the Copyright in the documents is given under warranty this warranty does not extend to the content of any document which may contain references or specifications or technologies that are covered by patents (also of third parties) or that refer to other standards. AXMEDIS is not responsible and does not guarantee that the information contained in the document is fully proprietary of AXMEDIS consortium and/or partners.
3. Licensee hereby undertakes to the Licensor that he will, without prejudice to any other right of action which the Licensor may have, at all times keep the Licensor fully and effectively indemnified against all and any liability (which liability shall include, without limitation, all losses, costs, claims, expenses, demands, actions, damages, legal and other professional fees and expenses on a full indemnity basis) which the Licensor may suffer or incur as a result of, or by reason of, any breach or non-fulfilment of any of his obligations in respect of this Licence.

7. INFRINGEMENT

1. Licensee undertakes to notify promptly the Licensor of any threatened or actual infringement of the Copyright which comes to licensee notice and shall, at the Licensor's request and expense, do all such things as are reasonably necessary to defend and enforce the Licensor's rights in the Copyright.

8. GOVERNING LAW AND JURISDICTION

1. This Licence shall be subject to, and construed and interpreted in accordance with Italian law.
2. The parties irrevocably submit to the exclusive jurisdiction of the Italian Courts.

Please note that:

- You can become affiliated with AXMEDIS. This will give you the access to a huge amount of knowledge, information and source code related to the AXMEDIS Framework. If you are interested please contact P. Nesi at nesi@dsi.unifi.it. Once affiliated with AXMEDIS you will have the possibility of using the AXMEDIS specification and technology for your business.
- You can contribute to the improvement of AXMEDIS documents and specification by sending the contribution to P. Nesi at nesi@dsi.unifi.it
- You can attend AXMEDIS meetings that are open to public, for additional information see WWW.axmedis.org or contact P. Nesi at nesi@dsi.unifi.it

Table of Content

| | | |
|----------|--|----------|
| 1 | EXECUTIVE SUMMARY AND REPORT SCOPE (DSI, ALL) | 8 |
| 2 | AXMEDIS CERTIFIER AND SUPERVISOR (WP5.6.1: DSI, EXITECH) | 9 |
| 2.1 | GENERAL SCENARIOS FOR THE CERTIFIED AND SUPERVISOR | 10 |
| 2.1.1 | Business Model (Distributor point of view) | 11 |
| 2.1.2 | Business Model (Collecting society or creator point of view) | 11 |
| 2.1.3 | Content Tracking and accounting | 13 |
| 2.1.4 | Single collection (single AXCS) | 14 |
| 2.1.5 | Hierarchical collection of information from the several AXCS, the Super AXCS | 15 |
| 2.1.6 | Reporting and accounting back to the Administration | 16 |
| 2.1.7 | ID generation and management | 17 |
| 2.2 | AXMEDIS REGISTRATION (DSI, EXITECH) | 18 |
| 2.2.1 | Application Scenario for registration | 18 |
| 2.2.2 | AXMEDIS Registration Logic | 21 |
| 2.2.3 | AXMEDIS Registration WEB Service interface formalization | 25 |
| 2.3 | AXMEDIS CERTIFICATION AND VERIFICATION (FUPF) | 28 |
| 2.3.1 | Architecture of the module | 28 |
| 2.3.2 | Certification and Verification Scenarios | 29 |
| 2.3.2.1 | Certification of Tool and User scenario | 30 |
| 2.3.2.2 | Certification of Tool and User scenario | 31 |
| 2.3.2.3 | Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption | 32 |
| 2.3.2.4 | Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption | 33 |
| 2.3.2.5 | Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption inside a Domain | 34 |
| 2.3.2.6 | Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption inside a Domain | 35 |
| 2.3.3 | AXMEDIS Certification and Verification Logics (FUPF) | 35 |
| 2.3.4 | AXMEDIS Certification and Verification WEB Service formalisation (FUPF) | 36 |
| 2.4 | AXMEDIS SUPERVISOR (FUPF) | 42 |
| 2.4.1 | Architecture of the module | 42 |
| 2.4.2 | Difference between Supervisor Input Data and Action Log | 44 |
| 2.4.3 | Supervisor Scenarios | 44 |
| 2.4.3.1 | Successful consumption of a protected AXMEDIS object | 44 |
| 2.4.3.2 | User not authorized scenario | 45 |
| 2.4.3.3 | Object key cached scenario | 46 |
| 2.4.4 | AXMEDIS Supervisor Logics (FUPF) | 46 |
| 2.4.5 | AXMEDIS Supervisor WEB Service formalisation (FUPF) | 47 |
| 2.5 | AXMEDIS REPORTING (DSI, EXITECH) | 62 |
| 2.5.1 | AXMEDIS Reporting Logic | 63 |
| 2.5.2 | AXMEDIS Reporting Web Service interface formalization | 65 |
| 2.6 | AXMEDIS STATISTIC ANALYSIS TOOL (DSI, EXITECH) | 68 |
| 2.6.1 | AXMEDIS Statistics Analysis Logic | 69 |
| 2.6.2 | AXMEDIS Statistics Analysis WEB Service interface formalisation | 71 |
| 2.7 | AXCS SYNCHRONIZER (DSI, EXITECH) | 74 |
| 2.7.1 | AXCS Synchronizer Logic | 75 |
| 2.8 | AXCS DATABASE INTERFACE (DSI, EXITECH) | 79 |
| 2.8.1 | AXCS Database Interface Logic | 79 |
| 2.9 | AXCS REGISTRATION AND CERTIFICATION DATABASE (DSI, EXITECH) | 83 |
| 2.9.1 | Entity-Relationship description | 83 |
| 2.9.2 | Relational database schema extended description | 85 |
| 2.10 | AXCS OBJECTS ID DATABASE (DSI, EXITECH) | 96 |
| 2.10.1 | Entity-Relationship description | 96 |
| 2.10.2 | Relational database schema extended description | 96 |
| 2.11 | AXCS ACCOUNTING DATABASE (DSI, EXITECH) | 103 |
| 2.11.1 | Action log received by AXCS (FUPF) | 103 |
| 2.11.2 | Entity-Relationship description (DSI) | 104 |

| | | |
|----------|--|------------|
| 2.11.3 | Relational database schema extended description (DSI)..... | 105 |
| 2.12 | AXCS MANAGER USER INTERFACE (DSI, EXITECH)..... | 112 |
| 2.12.1 | AXCS Manager User Interface logic..... | 113 |
| 3 | SUPER AXCS (DSI, EXITECH)..... | 115 |
| 3.1 | AXMEDIS SOFTWARE TOOL OFF LINE REGISTRATION (DSI, EXITECH)..... | 116 |
| 3.1.1 | AXMEDIS Software Tool Off-line Registration logic..... | 117 |
| 3.2 | AXMEDIS OID GENERATOR (WP5.6.2: DSI, EXITECH)..... | 120 |
| 3.2.1 | AXMEDIS OID Generator scenario..... | 121 |
| 3.2.2 | AXMEDIS OID Generator logic..... | 121 |
| 3.2.3 | AXMEDIS OID Generator algorithm..... | 123 |
| 3.2.4 | AXMEDIS OID Generator web service interface formalization..... | 124 |
| 3.3 | GLOBAL OBJECT LIST WEB SERVICE (DSI, EXITECH)..... | 126 |
| 3.3.1 | Global Object List logic..... | 127 |
| 3.3.2 | Global Object List WEB Service interface formalization..... | 129 |
| 3.4 | ACTIVE AXCS LIST/DATABASE (DSI, EXITECH)..... | 131 |
| 3.4.1 | Entity-Relationship description..... | 131 |
| 3.5 | AXMEDIS REGISTRATION OF AXCSs (DSI, EXITECH)..... | 133 |
| 3.5.1 | Application Scenario for AXCSs registration..... | 133 |
| 3.5.2 | AXMEDIS Registration of AXCSs logic..... | 134 |
| 3.5.3 | AXMEDIS Registration of AXCSs WEB Service interface formalization..... | 136 |
| 3.6 | AXCS COLLECTOR (DSI, EXITECH)..... | 138 |
| 3.6.1 | AXCS Collector scenario..... | 138 |
| 3.6.2 | AXCS Collector logic..... | 139 |
| 4 | PROTECTION TOOL AREA (WP4.5: FUPF, EXITECH, WP5.6.5: FHGIGD) | 142 |
| 4.1 | PROTECTION MANAGER SUPPORT, GENERAL..... | 143 |
| 4.1.1 | Scenarios related with Protection Manager Support..... | 143 |
| 4.1.1.1 | DRM Association for License Production, license references AXMEDIS object..... | 143 |
| 4.1.1.2 | DRM Association for License Production, license is stored inside the object..... | 144 |
| 4.1.1.3 | DRM Association for License Production, object stores license identifier..... | 145 |
| 4.1.1.4 | DRM Association for License Production, object stores reference to license server..... | 146 |
| 4.1.1.5 | Multiple license creation..... | 147 |
| 4.1.1.6 | License Acquisition (on-line use)..... | 148 |
| 4.1.1.7 | License Acquisition (off-line use)..... | 149 |
| 4.2 | A NETWORK OF AXCS AND PMS..... | 149 |
| 4.3 | PROTECTION MANAGER SUPPORT SERVER (FUPF)..... | 152 |
| 4.3.1 | Module profile definition..... | 152 |
| 4.3.2 | Architecture of PMS..... | 152 |
| 4.3.3 | WSDL Interface..... | 155 |
| 4.4 | PROTECTION MANAGER SUPPORT DOMAIN HOME (FUPF)..... | 177 |
| 4.4.1 | Module profile definition..... | 178 |
| 4.4.2 | Architecture of PMS..... | 178 |
| 4.4.3 | WSDL Interface..... | 180 |
| 4.5 | PROTECTION MANAGER SUPPORT DOMAIN FACTORY (FUPF)..... | 196 |
| 4.5.1 | Module profile definition..... | 197 |
| 4.5.2 | Architecture of PMS..... | 197 |
| 4.5.3 | WSDL Interface..... | 200 |
| 4.6 | PROTECTION MANAGER SUPPORT CLIENT (FUPF)..... | 222 |
| 4.6.1 | Module profile definition..... | 222 |
| 4.6.2 | Architecture of PMS..... | 223 |
| 4.6.3 | Sequence diagram..... | 225 |
| 4.6.4 | WSDL Interface..... | 225 |
| 4.7 | AXCS PROXY..... | 237 |
| 4.7.1 | Architecture of the module..... | 238 |
| 4.8 | DRM SUPPORT (FUPF)..... | 239 |
| 4.8.1 | Architecture of DRM Support module..... | 240 |
| 4.8.2 | WSDL Interface..... | 243 |
| 4.9 | AUTHORISATION SUPPORT (FUPF)..... | 260 |
| 4.9.1 | Architecture of Authorisation Support module..... | 261 |

| | | |
|----------|---|------------|
| 4.9.2 | WSDL Interface | 261 |
| 4.10 | RDD SERVER (FUPF) | 263 |
| 4.10.1 | Architecture of RDD Server module | 264 |
| 4.10.2 | WSDL Interface | 264 |
| 4.11 | SECURE COMMUNICATION SUPPORT (FUPF) | 266 |
| 4.11.1 | Description of the module | 267 |
| 4.11.1.1 | OpenSSL | 267 |
| 4.11.1.2 | Windows Version of OpenSSL | 267 |
| 4.11.1.3 | Secure session functionalities offered by OpenSSL | 267 |
| 4.11.2 | Architecture of the Secure communication support module | 268 |
| 4.12 | CONTENT CONSUMPTION STATUS (FUPF) | 271 |
| 4.12.1 | Content Consumption Scenario | 271 |
| 4.12.2 | Architecture of the module | 272 |
| 4.13 | LICENSE MANAGER (FUPF) | 278 |
| 4.13.1 | Architecture of License Manager module | 279 |
| 4.13.2 | WSDL Interface | 280 |
| 4.14 | LICENSE VERIFICATOR (FUPF) | 287 |
| 4.14.1 | License Verificator scenario | 287 |
| 4.14.2 | Architecture of License Verificator module | 288 |
| 4.14.3 | WSDL Interface | 288 |
| 4.15 | PROTECTION INFO MANAGER(FUPF) | 293 |
| 4.15.1 | Protection Info Manager Scenarios | 293 |
| 4.15.2 | Protection Info Manager module architecture | 295 |
| 4.16 | KEY GENERATOR (FUPF) | 300 |
| 4.16.1 | Key Generator module architecture | 301 |
| 4.17 | DOMAIN MANAGER (FUPF) | 302 |
| 4.17.1 | Domain Manager scenarios | 302 |
| 4.17.2 | Module Profile definition | 303 |
| 4.17.3 | Architecture of Domain Manager and Domain Registration Manager Module | 304 |
| 4.17.4 | Domain Manager WEB Service formalisation | 304 |
| 4.17.5 | Domain registration client | 311 |
| 4.17.5.1 | Architecture of Domain Registration Client Module | 311 |
| 4.17.5.2 | WSDL Description | 312 |
| 4.18 | SECURE CACHE MANAGER (FUPF) | 315 |
| 4.18.1 | Description of the module | 316 |
| 4.18.1.1 | OpenSSL | 316 |
| 4.18.1.2 | Windows Version of OpenSSL | 316 |
| 4.18.1.3 | Cryptographic functions provided by OpenSSL | 316 |
| 4.18.2 | Architecture of the module | 317 |
| 4.19 | LOCAL CACHE INFORMATION FORMAT (FUPF) | 318 |
| 4.19.1 | File format (FUPF) | 319 |
| 4.20 | LICENSE GENERATOR (FUPF) | 322 |
| 4.20.1 | Architecture of the License Generator | 323 |
| 4.20.2 | WSDL Interface | 324 |
| 4.21 | RIGHTS EXPRESSION TRANSLATOR (FUPF) | 329 |
| 4.21.1 | Architecture of the Rights Expression Translator | 330 |
| 4.21.2 | Rights Expression Translator WEB Service formalisation | 330 |
| 5 | PROTECTION TOOL ENGINE (FHGIGD) | 333 |
| 5.1 | SCENARIOS ON CONTENT PROTECTION AREA | 334 |
| 5.2 | PROTECTION RULE EDITOR (FHGIGD) | 335 |
| 5.3 | PROTECTION INFORMATION EDITOR AND VIEWER (FHGIGD) | 337 |
| 6 | AXMEDIS ACCOUNTING AREA (EXITECH) | 340 |
| 6.1 | CORE ACCOUNTING MANAGER AND REPORTING TOOL (EXITECH) | 342 |
| 6.1.1 | Core Accounting Manager and Reporting Tools interface toward the system | 342 |
| 6.1.2 | Core Accounting Manager and Reporting Tools interface toward the user | 343 |
| 6.1.3 | CAMART interface with AXCS | 346 |
| 6.1.4 | CAMART as a public service | 349 |
| 6.1.5 | CAMART user interface | 353 |

| | | |
|-------|--|-----|
| 6.2 | ADMINISTRATIVE INFORMATION INTEGRATOR (WP9.1: EXITECH) | 354 |
| 6.2.1 | Administrative Information Integrator in polling mode | 359 |
| 6.2.2 | Administrative Information Integrator in push mode | 359 |
| 6.2.3 | Administrative Information Integrator User Interface | 362 |
| 6.2.4 | Administrative Information Integrator and CAMART integration | 362 |

1 Executive Summary and Report Scope (DSI, all)

The full AXMEDIS specification document has been decomposed in the following parts:

- A. general aspects up to the description of the content model
- B. Viewers and players, including plug ins, etc.
- C. Content Production tools and algorithms
- D. Fingerprint and descriptors algorithms and tools
- E. Database area, query support and Content Crawling from CMS
- F. AXEPTool area, for B2B distribution and Programme and Publication for B2C distribution
- G. Workflow aspects and tools
- H. Protection tools and support, Certification and Supervision and Accounting tools
- I. Distribution tools and AXMEDIS Portal
- J. Definitions, tables, terminology, acronyms, lists, references, links and Appendixes

This document contains Part H concerning:

AXCS (AXMEDIS Certifier and Supervisor) and Super AXCS. AXCS is the authority put in charge of supervising the certification process over all its phases. Its own proper tasks concern about user and tool registration, certification and managements, object identifier generation and manipulation, object metadata collection, object usage registration and managements. All the data collected, managed and elaborated by AXCS have to be available for other subjects in the system entitled to get pertinent information.

Since AXCS is not unique (every Distributor could place his own AXCS) a coordinator subject has been introduced in order to supervising the activities of supervisors. This higher entity is called SuperAXCS. It has been projected so as being composed by most AXCS components with some other modules performing specific coordinating and data collecting tasks. Both AXCSs and Super AXCS don't contain detailed data about users: they know only users ID and all related data about objects usage. Users registration data is collected exclusively by Distributors.

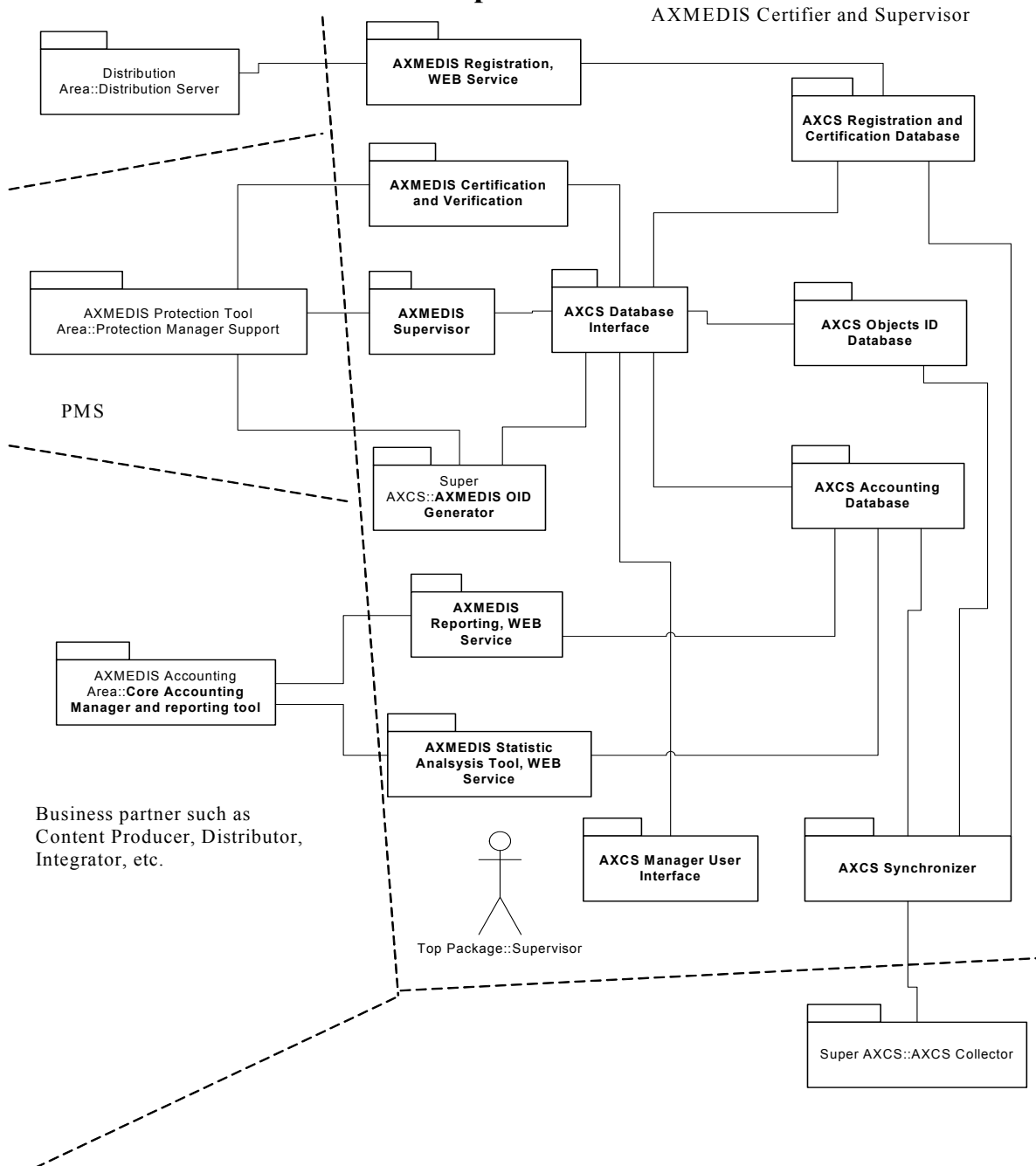
Protection Tool Area. The protection tool are include the PMS (Protection Manager Support) that is the tools for manaing the Digital Rights Management. The PMS allow estimating the grant locally by using a cached version of the license and protection information. In alternative it is capable of recovering the license and protection information via a connection with different instances of similar PMS Servers. The PMS exploit functionalities of the DRM support for the processing of REL/RDD or MPEG21. The PMS work in strictly agreement with the Protection Processor described in Part A.

Protection Tool Engine. The protection tools engine allows protecting objects in a systematic manner. It is mainly described as used interface in Part C, while in this document it is included only the aspects and relationships with protection tools of this document.

Accounting. The accounting area contain tools for collecting accounting information which are collected in the AXCS during its day by day work for instance the Action Log, which are the log of Events and actions performed on the objects in the player around the world. In this case different tools are used to get these information and move them from the AXCS to the administrative database of the content providers, distributors, integrators, collecting societies, etc.

2 AXMEDIS Certifier and Supervisor (WP5.6.1: DSI, EXITECH)

AXMEDIS Certifier and Supervisor



It includes:

- AXMEDIS Registration
- AXMEDIS Certification and Verification
- AXMEDIS Supervisor

AXMEDIS Project

CONFIDENTIAL

- AXCS registration and certification database
- AXCS Objects ID Database
- AXCS Accounting Database
- AXCS Database Interface
- AXMEDIS Reporting
- AXMEDIS Statistic Analysis Tool
- AXCS Synchronizer
- AXMEDIS SW Tool OffLine Registration
- Global Object List WEB Service
- AXCS Manager User Interface
- Etc...

AXMEDIS Certifier and Supervisor is the AXMEDIS certification authority that provides services for Content Providers and Distributors and verify the correctness of the Clients (as “Clients” are intended also software agents, not only physics or legal people).

AXMEDIS Certifier and Supervisor database structure is designed considering the distribution of services provided with the aim of scalable architecture capable of supporting a huge amount of transactions per second. These transactions can be of various kinds:

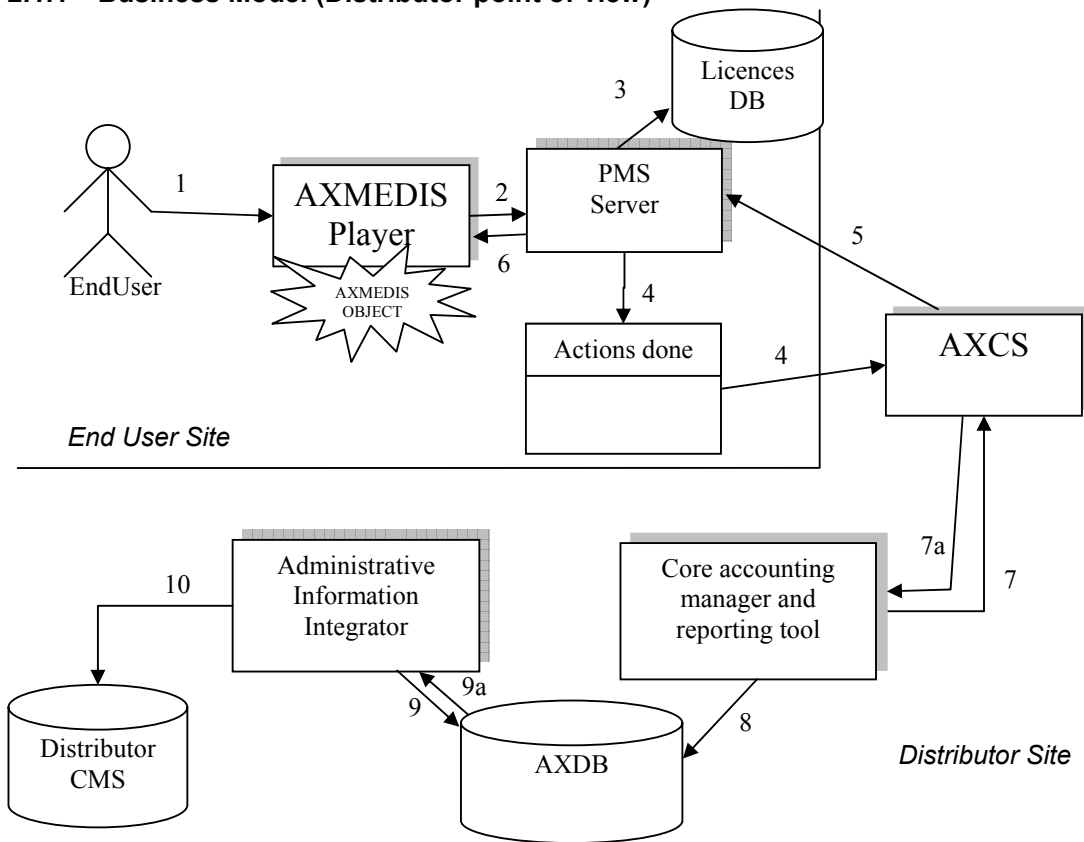
- requests of key and/or protection information,
- requests of verification,
- requests of logs,
- Registrations, etc...

The architecture of AXMEDIS Certifier and Supervisor has to be flexible enough to support centralised Certification and Supervision as well as distributed. In the centralised version only one AXMEDIS Certifier and Supervisor is set up for the whole network, in the other case each distributor and P2P network may have a distinct Certifier and Supervisors. They could be hierarchically organised or stand alone, limiting in this case the navigation of content.

The architecture of the AXMEDIS Certifier and Supervisor has to be scalable and the internal services should be well separable to cope with large traffic for the certification and supervision and to allow the decentralisation of some of the services in an easy and reconfigurable manner.

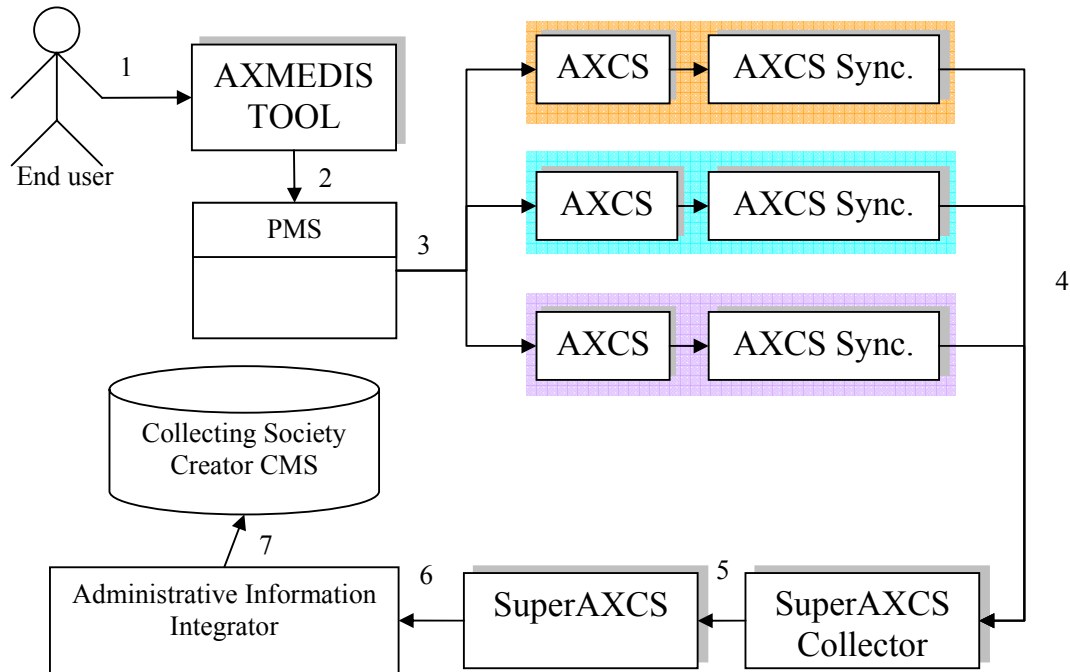
2.1 General Scenarios for the Certified and Supervisor

2.1.1 Business Model (Distributor point of view)



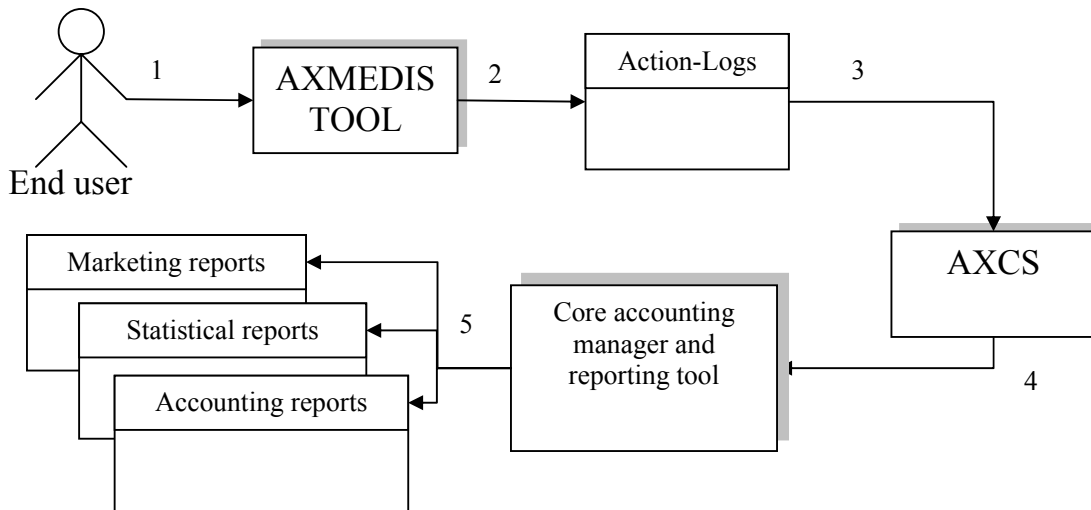
1. End User requests to perform an action on an AXMEDIS Protected Object
2. AXMEDIS Player asks PMS to perform an Action (assuming client has been already certified)
3. PMS checks in the LicenceDB if the Action is allowed (assuming OK)
4. PMS sends AXCS the action performed
5. AXCS gives back the key to access the content (if necessary)
6. PMS gives the grant to access the content and possibly the key to the AXMEDIS Player
7. Accounting & Rep. Tool retrieves from AXCS the actions performed by all the End Users on objects distributed by the distributor
8. A&R Tool stores the transactions in the AXDB
9. Adm. Integrator gets transactions performed from the DB
10. Administrative information are mapped into the Distributor CMS

2.1.2 Business Model (Collecting society or creator point of view)



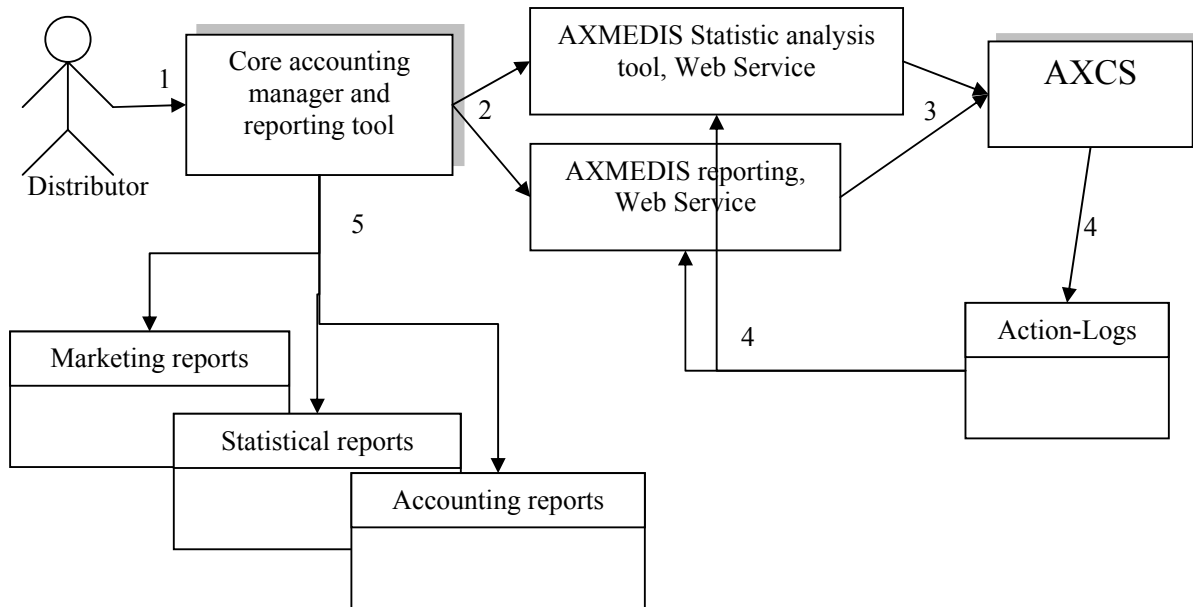
1. End User uses an AXMEDIS tool to operate on an AXMEDIS Protected Objects that are on different distribution channels
2. Protection Manager Support allow only authorized operations on the object
3. Objects are accessed on different channels and each AXCS stores its Action-Logs
4. Via the AXCS sync general information on objects or information that allow SuperAXCS to recover Action-Logs from the different AXCSs are transferred to the SuperAXCS Collector
5. SuperAXCS collects information
6. Administrative reports are created
7. Administrative Information Integrator transfer Action-Logs on CMS

2.1.3 Content Tracking and accounting



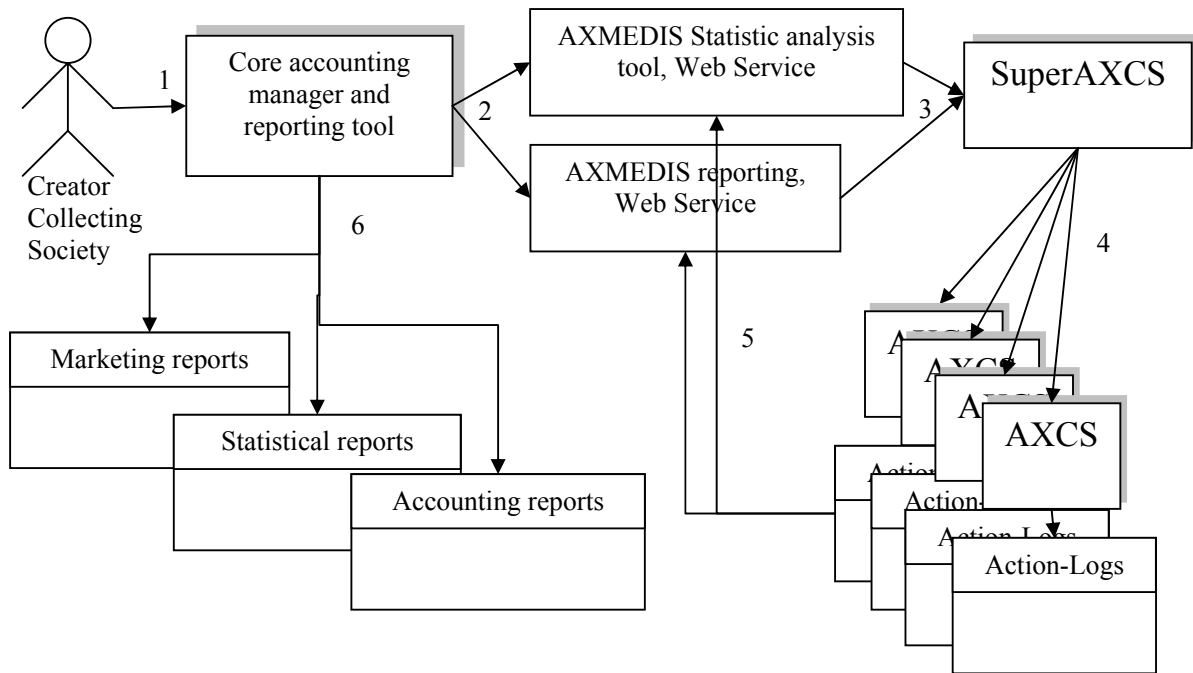
1. Distributor performs actions on objects
2. Action-Logs are generated (both on line and off line) reporting actions performed on objects
3. Action-Logs are stored by the AXCS
4. Core accounting manager and reporting tool extract information from AXCS allowing the generation of different reports type
5. Marketing reports, Statistical reports and accounting reports can be generated on demand

2.1.4 Single collection (single AXCS)



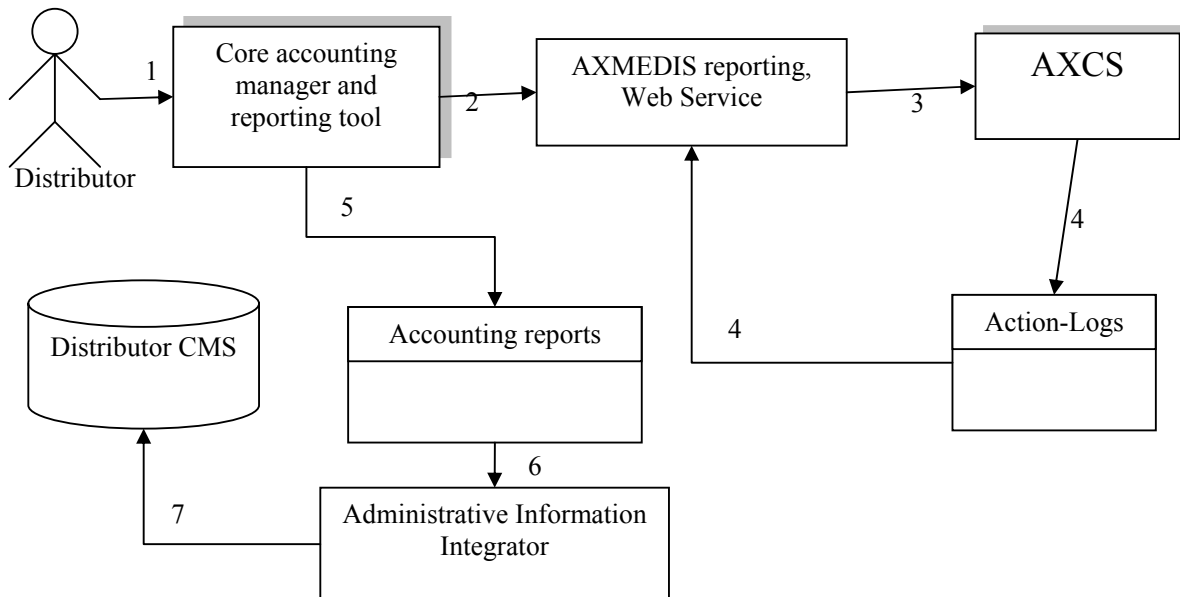
1. A Distributor wants to recover information on actions performed on the objects he has rights.
2. Core accounting manager and reporting tool query the correct tool for obtaining the Action-Logs in the correct form (anonymous or not, aggregated or not, etc)
3. AXMEDIS Statistic or reporting tools query AXCS
4. AXCS extracts the required Action-Logs and communicate them to the tools that perform actions to return results in the desired form
5. Different reports are generated on the basis of the information collected.

2.1.5 Hierarchical collection of information from the several AXCS, the Super AXCS



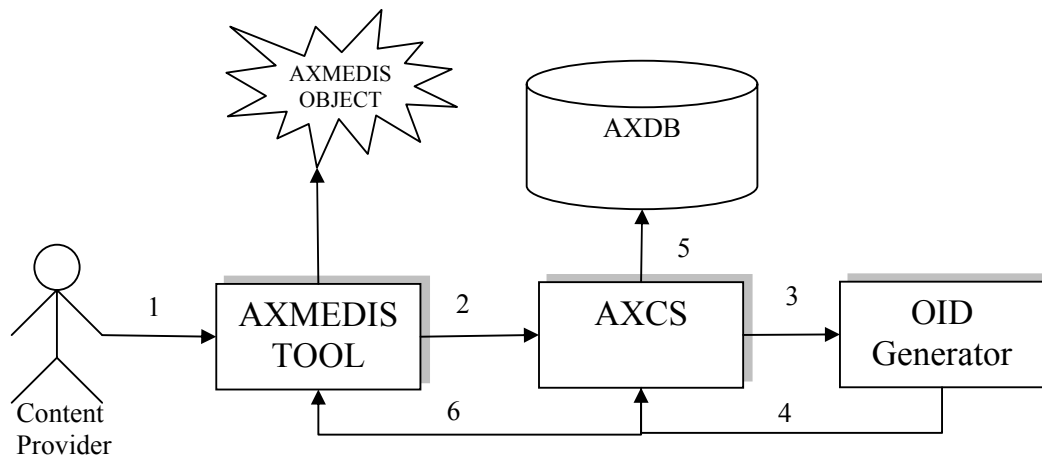
1. An Actor, that is collecting society or creator, wants to recover information on actions performed on the objects he has rights.
2. Core accounting manager and reporting tool query the correct tool for obtaining the Action-Logs in the correct form (anonymous or not, aggregated or not, etc)
3. AXMEDIS Statistic or reporting tools query the SuperAXCS
4. SuperAXCS recover information from the different AXCSs
5. The different AXCSs extract the required Action-Logs and communicate them to the tools that perform actions to return results in the desired form
6. Different reports are generated on the basis of the information collected.

2.1.6 Reporting and accounting back to the Administration



1. A Distributor wants to recover information on actions performed on the objects he has rights.
2. Core accounting manager and reporting tool query the reporting webservice for obtaining the Action-Logs
3. AXMEDIS Statistic or reporting tools query AXCS
4. AXCS extracts the required Action-Logs and communicate them to the reporting tool
5. Accounting report is generated.
6. Accounting report is passed to the Administrative Information Integrator
7. Data are loaded in the Distributor CMS

2.1.7 ID generation and management



1. A Creator, Distributor or a Content Provider, in more general sense, wants to create a new AXMEDIS Object.
2. The tool with that is creating the object requests to the AXCS an object ID
3. AXCS interact with Object ID generator to ask an ID
4. OID generator returns back to AXCS the generated ID
5. AXCS stores ID and related info in the AXDB
6. Generated ID is communicated to the tool
7. A new AXMEDIS object is created with the assigned ID

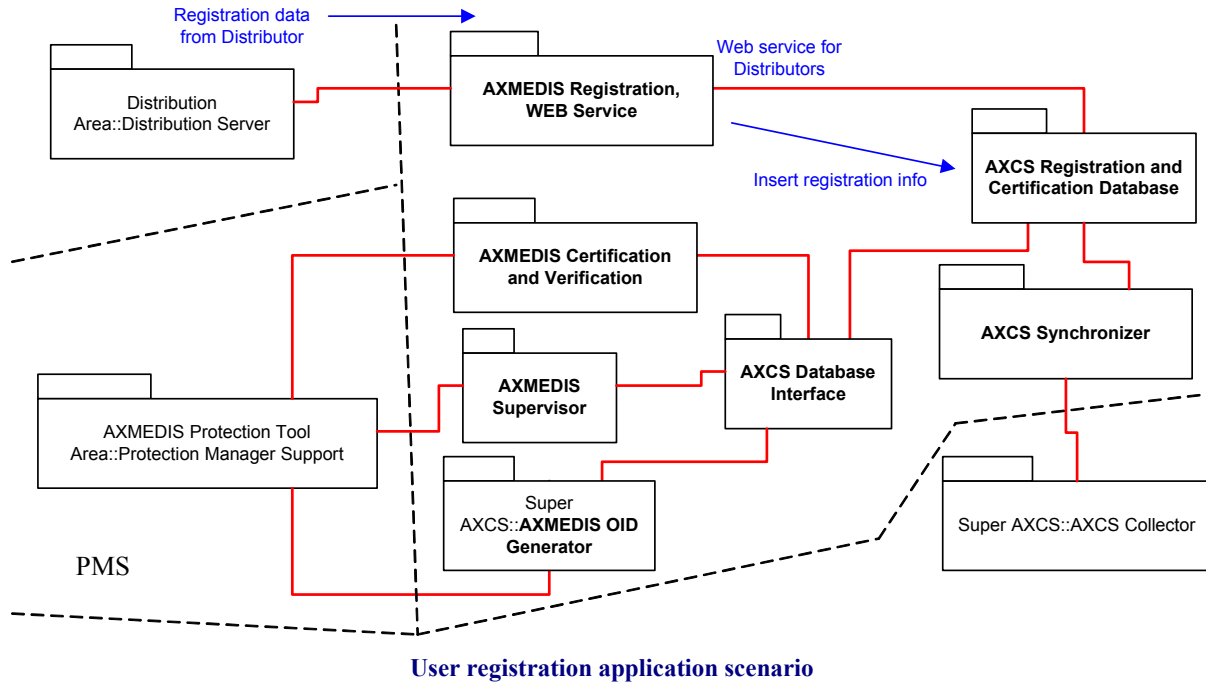
2.2 AXMEDIS Registration (DSI, EXITECH)

| Module Profile | | |
|---|-----------------------------------|--|
| AXMEDIS Registration | | |
| Executable or Library(Support) | Web application | |
| Single Thread or Multithread | | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Distribution Area:Distribution Server | Web service based on (TBD) | Protected (SSL) |
| AXCS: Registration and Certification Database | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No user interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

All AXMEDIS users must be registered by AXCS. Several information collected by Distributors in the registration phase has to be transferred to AXCS. The AXMEDIS Registration Web Services is the AXCS module that receives this information and store it in the AXCS Registration and Certification Database. Once the user is registered in the system and the related data is stored in the AXCS Registration and Certification Database, other AXCS modules can access this database to retrieve user information to perform their work.

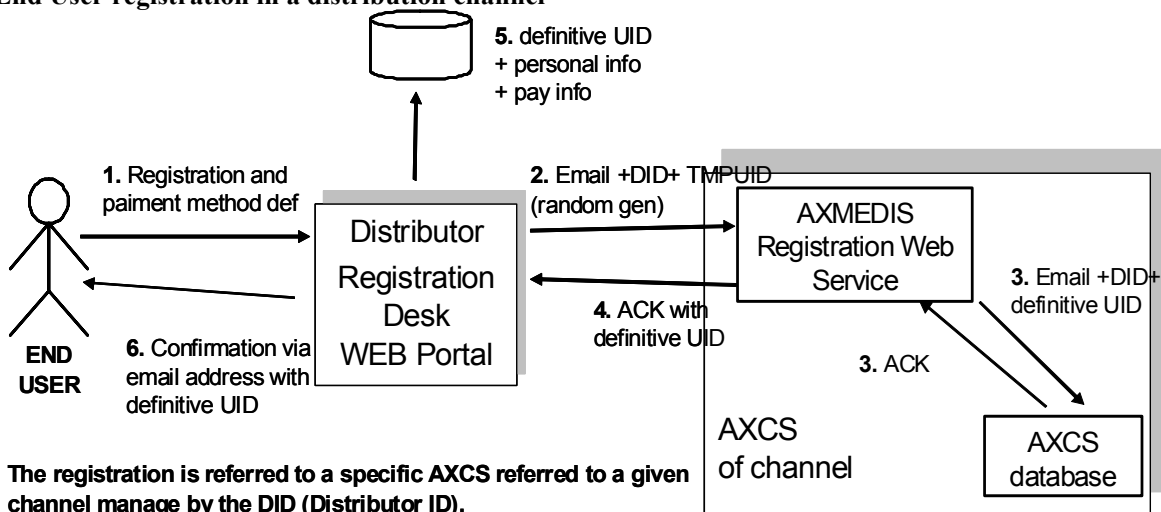
2.2.1 Application Scenario for registration

The scenario shows the relationship between the AXMEDIS Registration module and other modules inside AXCS.



Here are reported some registration scenarios to better understand the role of the AXCS Registration Web Service.

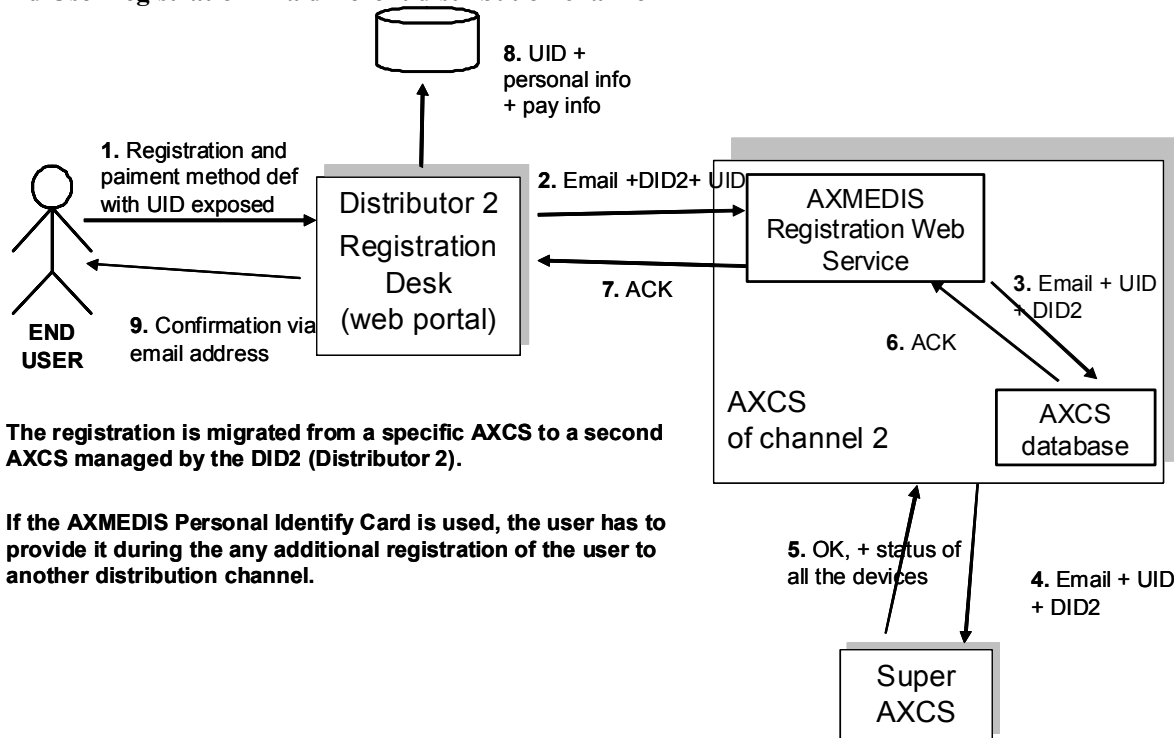
End User registration in a distribution channel



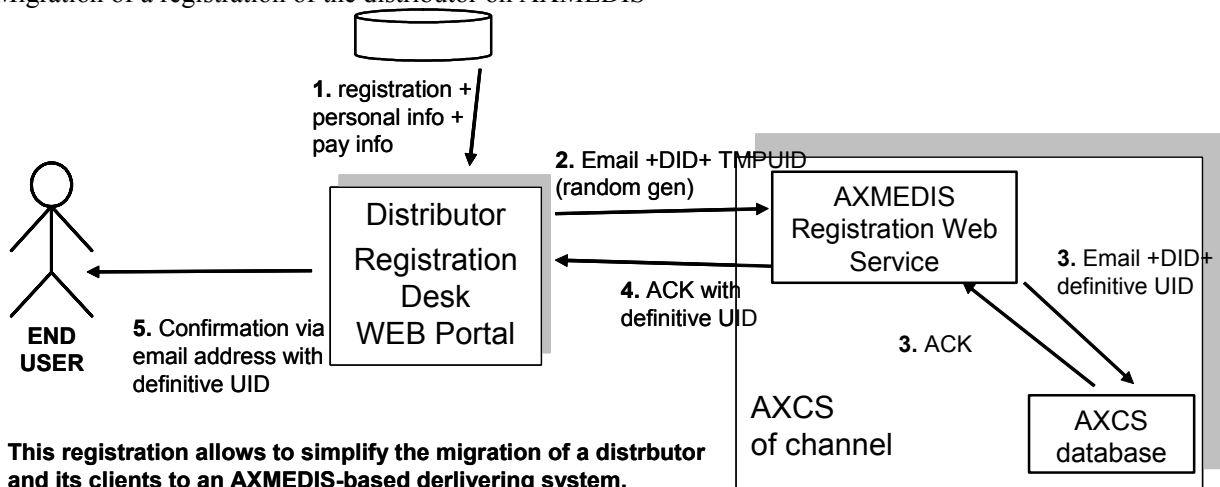
The registration is referred to a specific AXCS referred to a given channel managed by the DID (Distributor ID).

Instead of a definitive UID we can use a "Certificate" or what we can call the AXMEDIS Personal Identity Card (AXPIC). It can be a certificate that one can exhibit to authenticate himself/herself in the AXMEDIS circuit, a check is typically done with that ID and the email, etc...

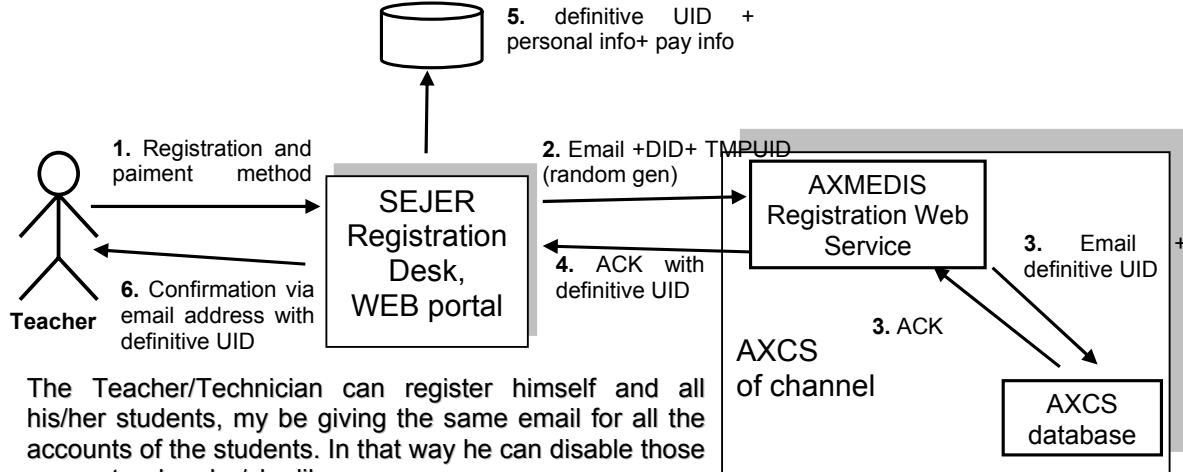
End User registration in a different distribution channel



Migration of a registration of the distributor on AXMEDIS



Instead of a definitive UID we can use a “Certificate” or what we can call the AXMEDIS Personal Identity Card (XPIC). It can be a certificate that one can exhibit to authenticate himself/herself in the AXMEDIS circuit, a check is typically done with that ID and the email ,etc...

Registration of a new Teacher/School or Student

The Teacher/Technician can register himself and all his/her students, by giving the same email for all the accounts of the students. In that way he can disable those accounts when he/she likes.

Teacher: <UID354135>

Student sdhfg: <UID134514>

Student afsdhKLFH: <UID675737>

Student dgag: <UID437673>

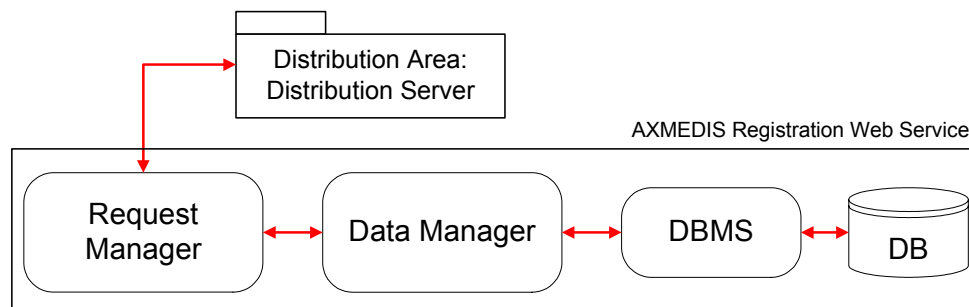
Student rtywuyert: <UID36773673>

Different UID (or AXMEDIS Personal Identify Cards) will be received by the Teacher, that has to save all the info (emails)

2.2.2 AXMEDIS Registration Logic

The AXMEDIS Registration Web Service is a web application running on a web server (perhaps Apache or IIS) implemented as a set of web scripts. We can identify the following logical decomposition:

- **Request Manager**: this component receives registration requests from Distributor, and prepares them to be processed by the Data Manager. The communication channel connecting Distributor and Request Manager is protected using a secure protocol (for instance SSL). It implements the interface with requestors and manage the whole application.
- **Data Manager**: this component receives data from Request Manager and inserts it in the AXCS Registration and Certification Database. If necessary elaborates and fits data before insert it in the database. The database management is performed using the related API (AXDB-API).



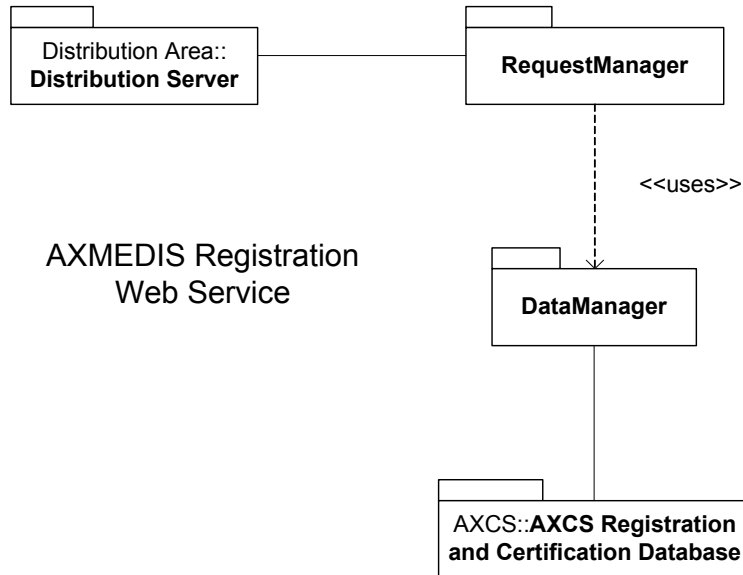
AXMEDIS Registration Web Service architecture.

The **Request Manager** should be composed by some modules implementing the following functions:

- Accepting authentication data, needful to verify Distributors credentials and make it access the system
- Accepting Users registration data and preparing it to be transferred to Data Manager
- Reply to requestor with the most appropriate message (on the basis of Data Manager responses)

The **Data Manager** should be composed by some modules implementing the following functions:

- Receiving verification requests and data from the Request Manager
- Accessing the database (using the AXDB-API) to verify Distributors Credential information
- Accessing the database to store received Users registration data
- Reply to Request Manager according to the performed actions



The following table describes methods thought to be used in AXMEDIS Registration Web Service.

| Method name and parameters | Output | Description |
|---|--|--|
| scope: public | | |
| RequestManager::Registration(string NickName, string Password, usrDatatype regdata) | string result string definitive_UID | It is the only public method of this web service. It collects Distributor credentials needful to access the system and uses the VerifyLogin() (a DataManager method) to verify requestor credentials. It collects also registration data (regdata) provided by the requesting distributor and uses the other methods (described below) to insert them in database and to provide the result to the requesting distributor. The result output parameter is set to 0 if the registration is successful otherwise is set to 1. This method returns also the definitive AXUID or B2BUserID in definitive_UID |
| scope: private | | |
| RequestManager::DataPrepare() | All user registration fields. See | It prepares the User registration |

| | | |
|---|---------------------------------------|--|
| | below for details. | data received from the Distributor by the web service to be stored in DB. It distinguishes between Users and B2B Users (like Creators, Distributors and so on) on the basis of received data. Is also uses UIDGenerator() to get the definitive UID (AXUID or B2BUserID) to be inserted in the database and to be sent to requestor |
| RequestManager::UIDGenerator(string TempUID) | String DefinitiveUID | It provides the definitive User ID, AXUID or B2BUserID. |
| RequestManager::EncryptComm(string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| RequestManager::DecryptComm(string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| DataManager::VerifyLogin(string NickName, string Password) | boolean IsTrusted string PublicKey | It accesses AXCS Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process. It also retrieves the Distributor public key stored in the DB. Note: the password is encrypted both the one received as input from the method and the one stored in DB. The equality check is made between encrypted strings. |
| DataManager::StoreData() (the needful input data can be stored in the class attributes or sent as method parameters) | | Store received data in the AXCS Registration and Certification Database |

Note that it has been introduced two encrypting/decrypting methods (DecryptComm, EncryptComm) to enforce the encrypting robustness. In fact we can suppose to use an encrypted protocol (like SSL), but we can enforce encryption robustness (and therefore security) encrypting our self data too using a Public/Private key paradigm. You have to remember that the Distributor public key is stored in AXCS Registration and Certification Database.

Here is a list of attributes containing data about Users sent by Distributors to AXMEDIS Registration Web Services to be stored in AXMEDIS Registration and Certification database. The registration attribute set is changeable on the basis of the registering user role in the AXMEDIS system. First we have to distinguish between Users and B2BUsers. The firsts are the final Users of the AXMEDIS system. The others are the user involved (in some way) in the production, creation or distribution process.

Case of Users attributes

AXUID (PK)
AXDOM (FK)
email
NickName
Password
nationality
PubKey
RegDate
RegDeadline
Status

Case of B2BUsers attributes

AXMEDIS Project

CONFIDENTIAL

B2BUserID (PK)**Website****RefName****Phone****Company****CmpAddress****CmpPhone1****CmpPhone2****CmpFax****Email****NickName****Password****Location****Nationality****PubKey**

In the case of B2BUsers, we have to insert additional information about the B2BUser role in the AXMEDIS system. Here we have to distinguish between Creators, Distributors, B2BAxepTool, Collecting Societies or Tool Producers.

Case of Creators attributes**AXCID (PK)****AXDOM (FK)****B2BUserID (FK)****RegDate****RegDeadline****Status****Case of Distributors attributes****AXDID (PK)****AXDOM (FK)****B2BUserID (FK)****RegDate****RegDeadline****Status****Case of B2BAxepTools attributes****AXAID (PK)****AXDOM (FK)****B2BUserID (FK)****PubKey****RegDate****RegDeadline****Status****Case of Collecting Societies attributes****AXSID (PK)****AXDOM (FK)****B2BUserID (FK)****NationDomain****RegDate****RegDeadline****Status****Case of Tool Producers attributes****AXPID (PK)****AXDOM (FK)****B2BUserID (FK)****RegDate****RegDeadline****Status**

This attributes list has to be complaint with the pertinent database tables respect to data types.

2.2.3 AXMEDIS Registration WEB Service interface formalization

In the present paragraph is explained the Registration Web Service interface using the WSDL formalism.

| | |
|------|---|
| WSDL | <pre> <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:rm="http://new.webservice.namespace" targetNamespace="http://new.webservice.namespace"> <types> <xs:schema targetNamespace="http://new.webservice.namespace" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="usrDatatype"> <sequence> <!-- case of single user --> <element name="AXUID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="email" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="NickName" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Password" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="nationality" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="PubKey" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDate" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDeadline" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Status" type="xsd:string" minOccurs="0" maxOccurs="1"/> <!-- case of B2B users --> <element name="B2BUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Website" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RefName" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Phone" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Company" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="CmpAddress" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="CmpPhone1" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="CmpPhone2" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="CmpFax" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Email" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="NickName" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Password" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Location" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Nationality" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="PubKey" type="xsd:string" minOccurs="0" maxOccurs="1"/> <!-- case of Creators --> <element name="AXCID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="B2BUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDate" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDeadline" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Status" type="xsd:string" minOccurs="0" maxOccurs="1"/> <!-- case of Distributors --> <element name="AXDID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="B2BUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDate" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDeadline" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Status" type="xsd:string" minOccurs="0" maxOccurs="1"/> <!-- case of B2BAxepTools --> <element name="AXAID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="B2BUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="PubKey" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDate" type="xsd:string" minOccurs="0" maxOccurs="1"/> </pre> |
|------|---|

| | |
|------------------------------|---|
| | <pre> <element name="RegDeadline" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Status" type="xsd:string" minOccurs="0" maxOccurs="1"/> <!-- case of Collecting Societies --> <element name="AXSID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="B2BUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="NationDomain" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDate" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDeadline" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Status" type="xsd:string" minOccurs="0" maxOccurs="1"/> <!-- case of Tool Producers --> <element name="AXPID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="B2BUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDate" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegDeadline" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Status" type="xsd:string" minOccurs="0" maxOccurs="1"/> </sequence> </complexType> </xs:schema> </types> <message name="RegistrationRequest"> <part name="NickName" type="xs:string"/> <part name="Password" type="xs:string"/> <part name="Regdata" type="rm:usrDatatype"/> </message> <message name="RegistrationResponse"> <part name="Result" type="xs:string"/> <part name="definitive_UID" element="" type="xs:string"/> </message> <portType name="Registration_PortType"> <operation name="Registration"> <input message="rm:RegistrationRequest"/> <output message="rm:RegistrationResponse"/> </operation> </portType> <binding name="Registration" type="rm:Registration_PortType"> <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="Registration"> <soap:operation soapAction="urn:#Registration"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <soap:operation soapAction="urn:#Registration"/> </operation> </binding> <service name="RegistrationWebService"> <port name="Registration" binding="rm:Registration"> <soap:address location="No Target Adress"/> </port> </service> </definitions> </pre> |
| Request Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Registration xmlns:m="http://new.webservice.namespace"> <NickName>...distributor user...</NickName> <Password>...password...</Password> <Regdata> <AXUID>0A2Z4X678B0124456X89W12345B7P901</AXUID> <AXDOM>0V2X4XV75BD1SS456XC5P98345BD4L05</AXDOM> <email>luxor@unknown.com</email> <NickName>luxor27</NickName> <Password>...password...</Password> <nationality>French</nationality> <PubKey>...public key...</PubKey> </Regdata> </m:Registration> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |

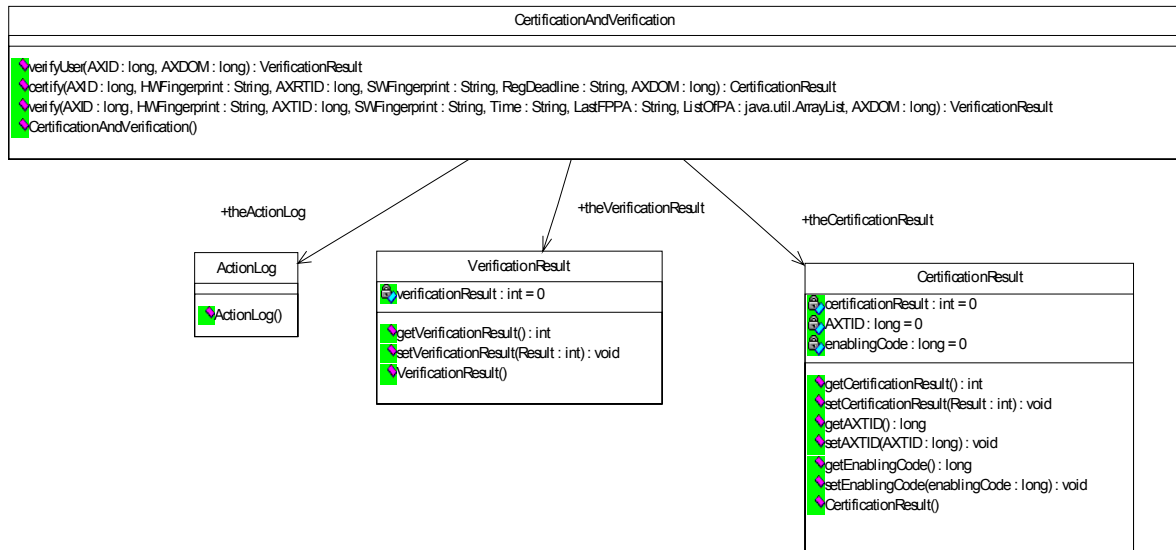
| | |
|-------------------------------|--|
| | <pre> <RegDate>2005-02-05T14:03:10</RegDate> <RegDeadline>2007-02-05T14:03:10</RegDeadline> <Status>U</Status> <B2BUserID></B2BUserID> <Website></Website> <RefName></RefName> <Phone></Phone> <Company></Company> <CmpAddress></CmpAddress> <CmpPhone1></CmpPhone1> <CmpPhone2></CmpPhone2> <CmpFax></CmpFax> <Email></Email> <NickName></NickName> <Password></Password> <Location></Location> <Nationality></Nationality> <PubKey></PubKey> <AXCID></AXCID> <AXDOM></AXDOM> <B2BUserID></B2BUserID> <RegDate></RegDate> <RegDeadline></RegDeadline> <Status></Status> <AXDID></AXDID> <AXDOM></AXDOM> <B2BUserID></B2BUserID> <RegDate></RegDate> <RegDeadline></RegDeadline> <Status></Status> <AXAID></AXAID> <AXDOM></AXDOM> <B2BUserID></B2BUserID> <PubKey></PubKey> <RegDate></RegDate> <RegDeadline></RegDeadline> <Status></Status> <AXSID></AXSID> <AXDOM></AXDOM> <B2BUserID></B2BUserID> <NationDomain></NationDomain> <RegDate></RegDate> <RegDeadline></RegDeadline> <Status></Status> <AXPID></AXPID> <AXDOM></AXDOM> <B2BUserID></B2BUserID> <RegDate></RegDate> <RegDeadline></RegDeadline> <Status></Status> </Regdata> </m:Registration> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
| Response Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Registration xmlns:m="http://new.webservice.namespace"> <Result>0</Result> <definitive_UID>0A2Z4X678B0124C56X89W123452CV478</definitive_UID> </m:Registration> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |

2.3 AXMEDIS Certification and Verification (FUPF)

| Module Profile | | |
|--|-----------------------------------|--|
| AXMEDIS Certification and Verification | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| PMS (Protection Processor) | | |
| AXCS Database Interface | | |
| AXMEDIS Supervisor | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

2.3.1 Architecture of the module

The following figure shows the UML diagram of this module, together with the definition of the public operations. The ActionLog class is defined in the AXMEDIS Supervisor.

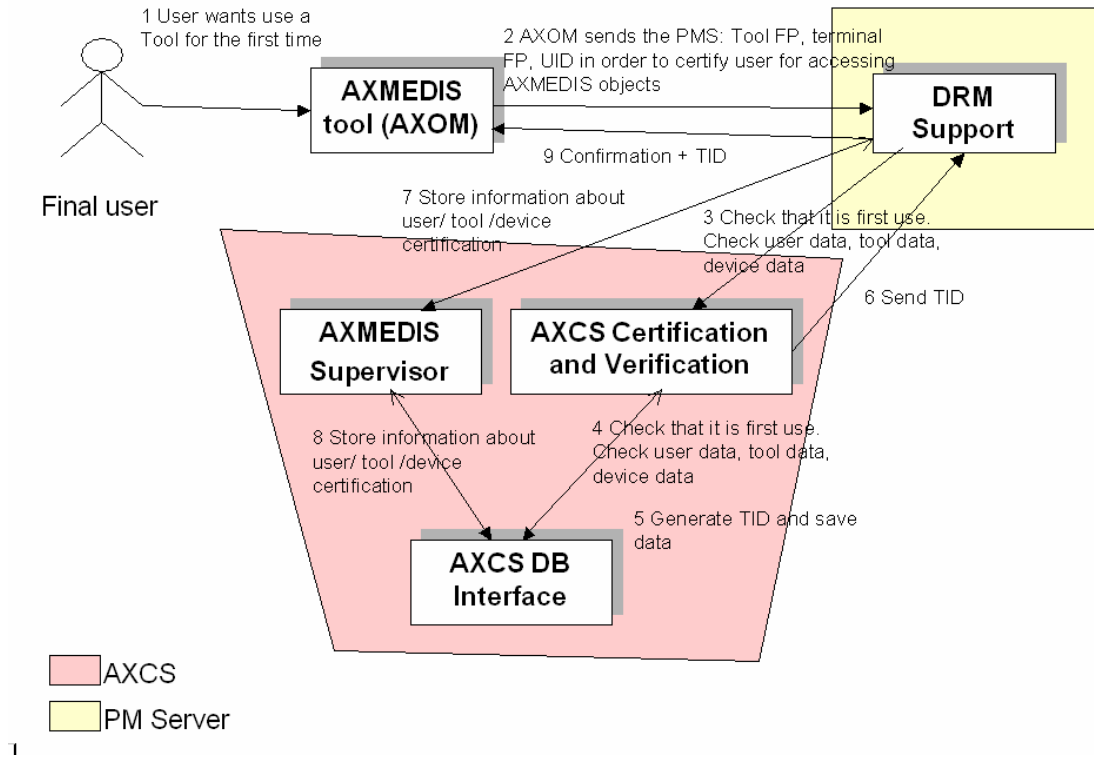


2.3.2 Certification and Verification Scenarios

The following scenarios show the relationship between the AXMEDIS Certification and Verification module and other modules inside AXCS.

2.3.2.1 Certification of Tool and User scenario

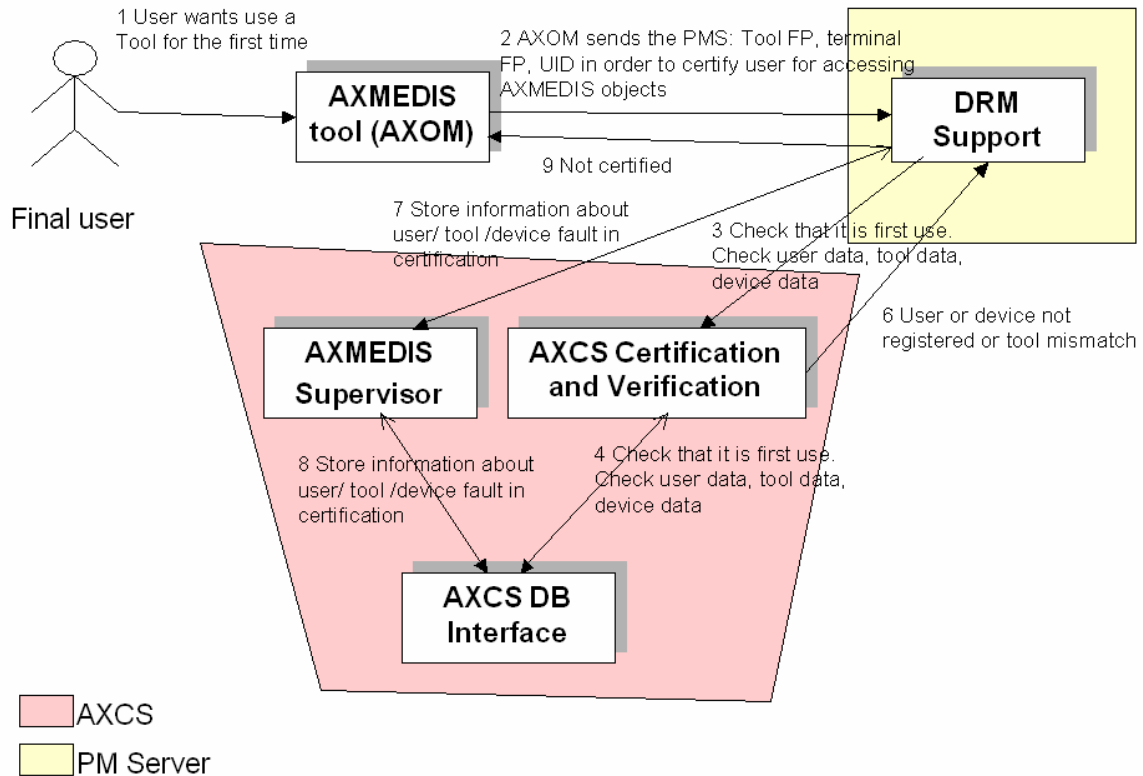
The scenario in then next figure represents the certification of a user and the corresponding AXMEDIS tool on a device, when the user accesses the system for the first time after registration.



Certification of Tool and User scenario

2.3.2.2 Certification of Tool and User scenario

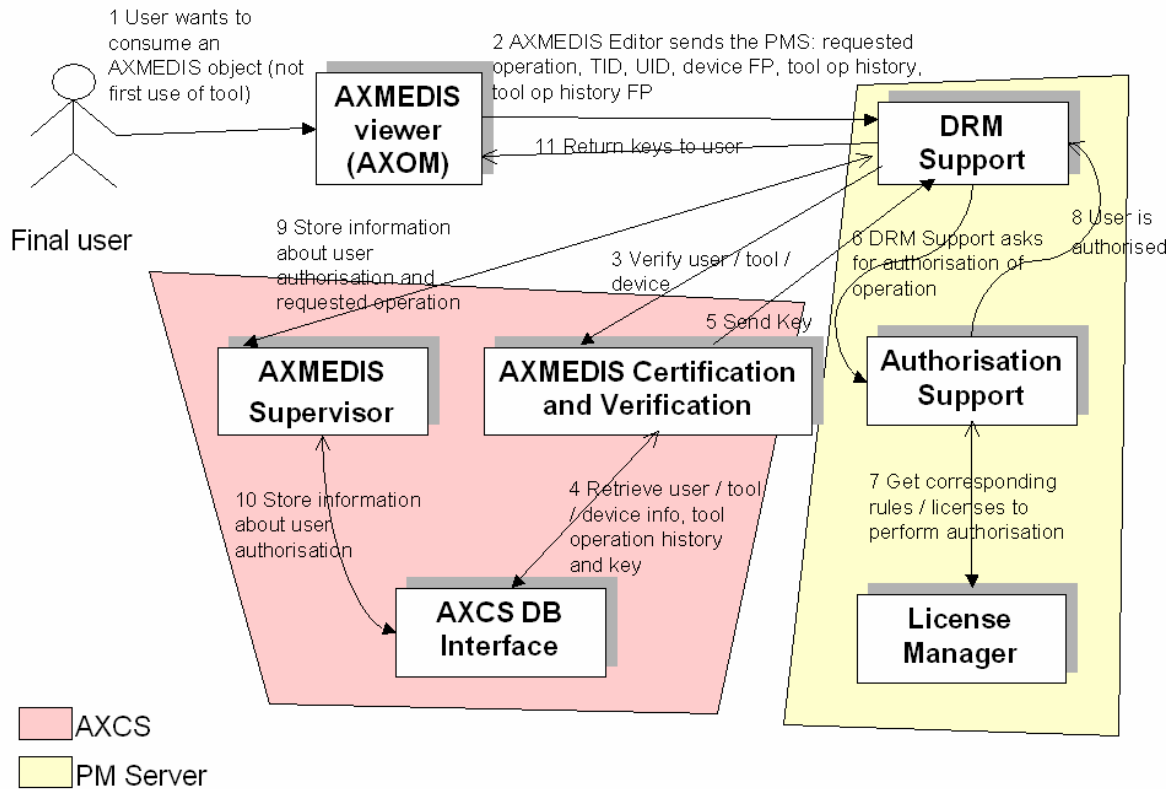
The scenario represents the relationship among the different components of AXCS when a user or device are not verified or tool is not certified. This scenario also shows the AXCS Supervisor module functionality in this case.



Certification of tool and user scenario. User or device not registered or Tool mismatch

2.3.2.3 Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption

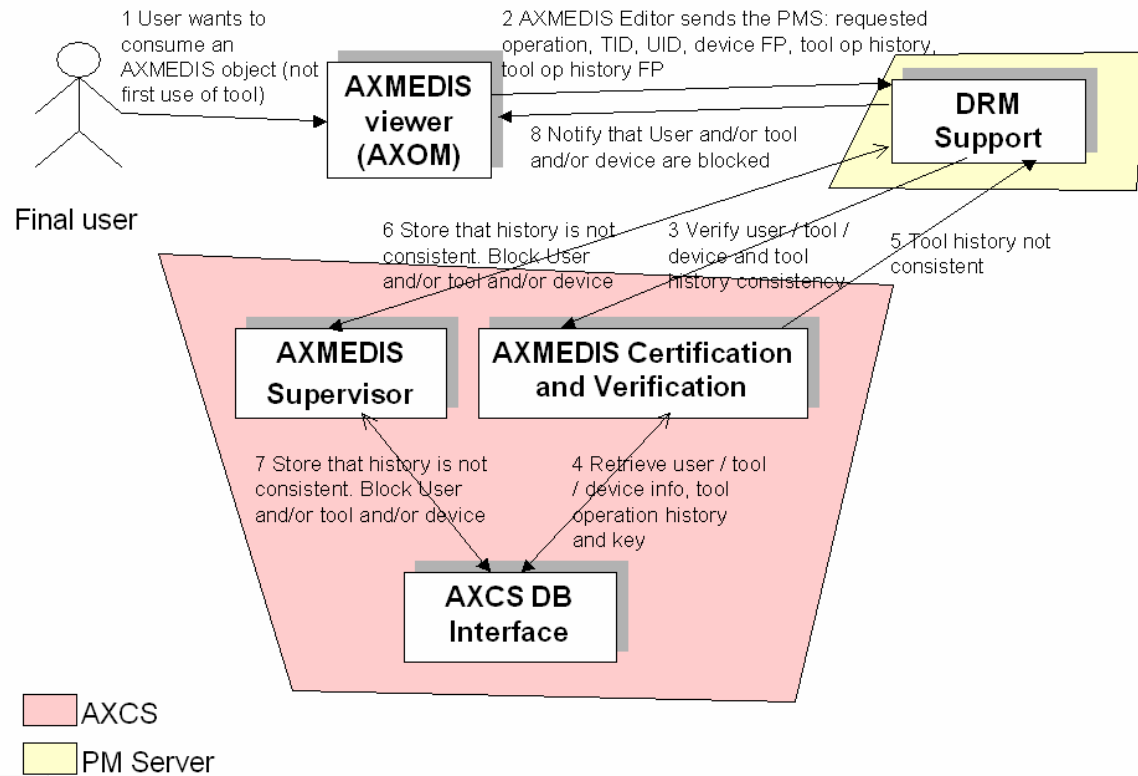
The scenario represents the relationship among the different components of AXCS during successful content consumption.



Successful verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption

2.3.2.4 Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption

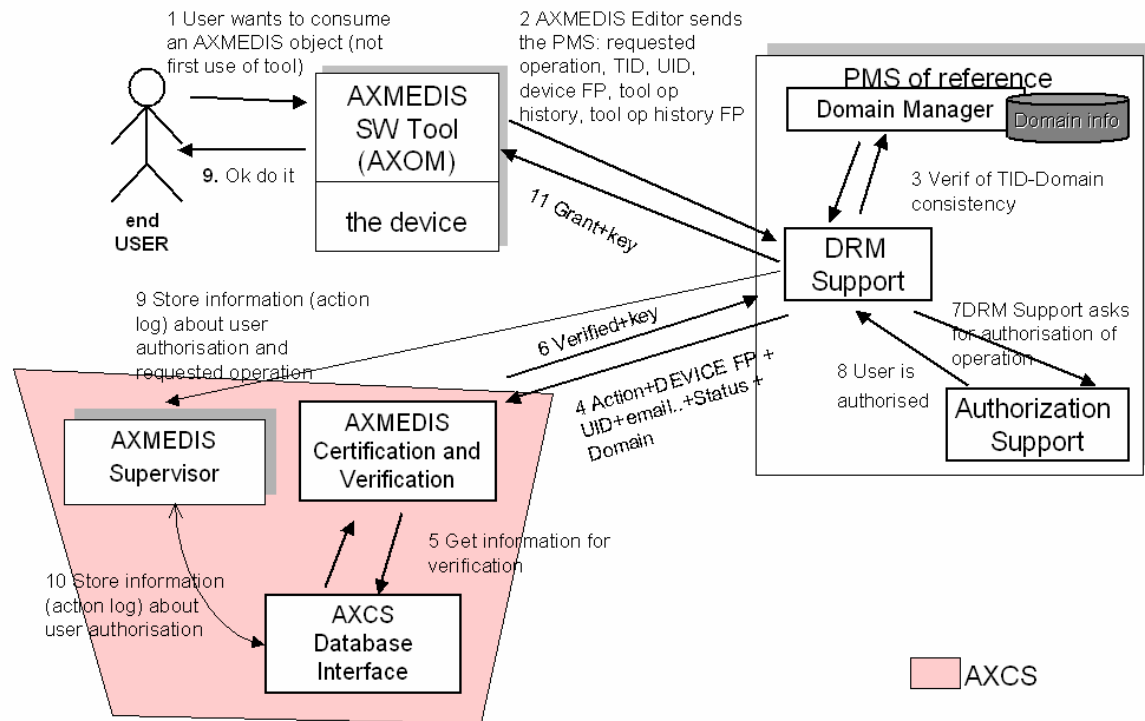
The scenario represents the relationship among the different components of AXCS during unsuccessful content consumption due to Tool FP mismatch/user mismatch/device mismatch.



Unsuccessful verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption due to Tool operation inconsistency

2.3.2.5 Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption inside a Domain

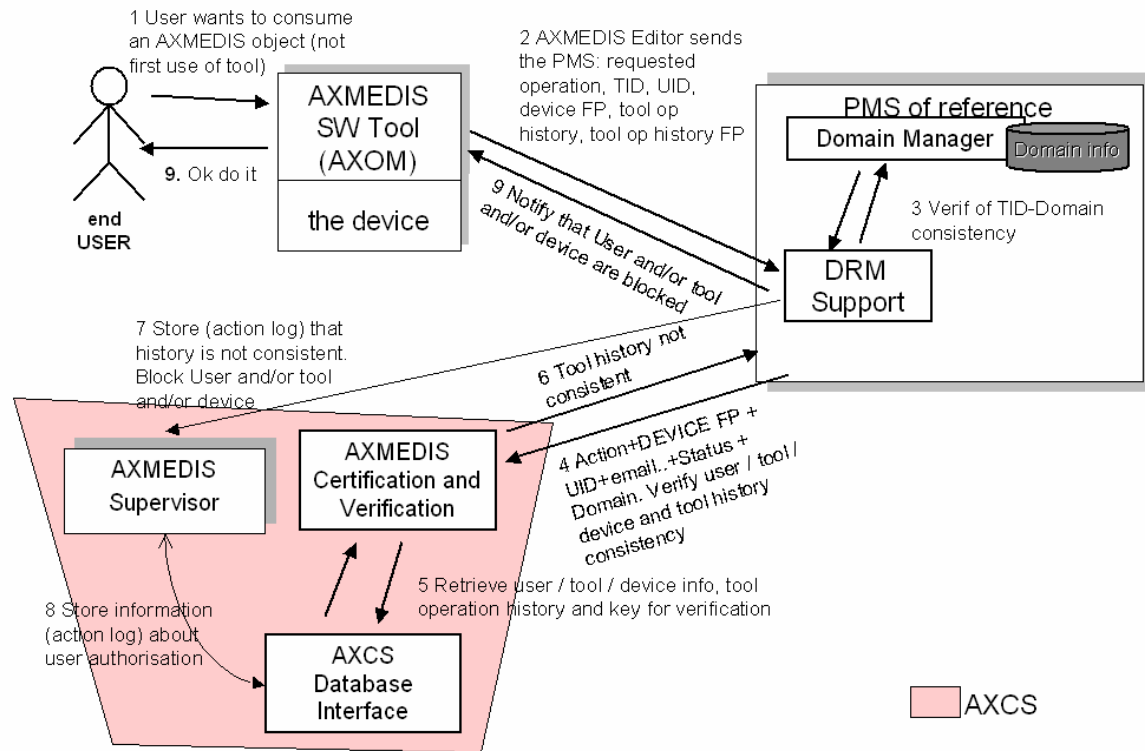
The scenario represents the relationship among the different components of AXCS during successful content consumption inside a Domain



Successful verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption inside a Domain

2.3.2.6 Verification of AXMEDIS Users using AXMEDIS Tools on a Device during content consumption inside a Domain

The scenario represents the relationship among the different components of AXCS during unsuccessful content consumption inside a Domain due to Tool operation inconsistency



Unsuccessful content consumption inside a Domain due to Tool operation inconsistency

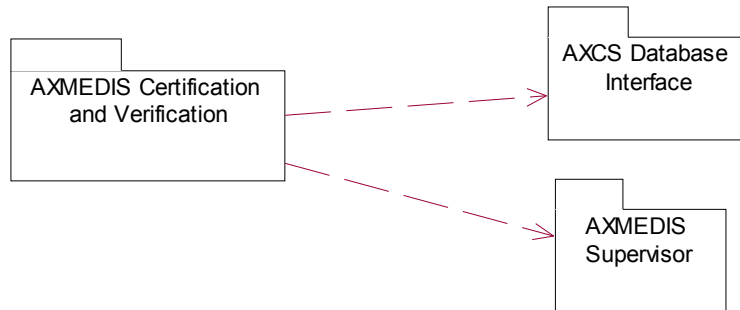
2.3.3 AXMEDIS Certification and Verification Logics (FUPF)

The AXMEDIS Certification and Verification is a web service running on a server and it is implemented in Java.

It communicates with the following modules:

- Protection Processor. AXCv receives requests from the Protection Processor through the interfaces provided as web services. The communication between the Protection Processor and AXCv is made through axcsProxy provided by PMS Client and over a secure channel
- AXCS Database Interface. AXCv stores, retrieves and updates the information related to the certification and verification of users, devices and tools. The communication between AXCv and AXCS Database Interface is made the interfaces provided as web services and over a secure channel

- AXMEDIS Supervisor. AXCv requests AXS to resynchronize the actions performed during the offline operation. The communication between AXCv and AXS is made through a web service provided by AXS and over a secure channel.



The following table describes the methods offered by the AXMEDIS Certification and Verification Web Service.

| CertificationAndVerification | |
|------------------------------|---|
| Method | Description |
| verifyUser | This method can be used to verify the status of a user, optionally inside a domain |
| certify | This method is called by the Protection Processor, through the axcsProxy. It is used to certify that the original tool has not been modified and to activate it. It creates a new entry in the CerTools table of the AXCS database associated to the tool user and returns an activation code and tool identifier to the Protection Processor. |
| verify | This method is called by the Protection Processor, through the axcsProxy. It is used to verify that the tool installed on a device has neither been modified nor blocked, that the user is not blocked and that the registered tool is not blocked. It is also responsible for resynchronizing the offline tool operation through the AXSupervisor. |

2.3.4 AXMEDIS Certification and Verification WEB Service formalisation (FUPF)

This section contains the formalisation of the AXMEDIS Certification and Verification Web service.

| CertificationAndVerification | |
|------------------------------|---|
| WSDL | <pre> <?xml version="1.0"?> <wsdl:definitions name="CertificationAndVerification" targetNamespace="urn:CertificationAndVerification" xmlns:tns="urn:CertificationAndVerification" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="ActionLog"> <!-- Defined in AXMEDIS Supervisor --> </xsd:complexType> <xsd:complexType name="VerificationResult"> <xsd:sequence> <xsd:element name="verificationResult" type="xsd:int"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="CertificationResult"> <xsd:sequence> </pre> |

```

        <xsd:element name="AXTID" type="xsd:long"/>
        <xsd:element name="certificationResult" type="xsd:int"/>
        <xsd:element name="enablingCode" type="xsd:long"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/containers/literal/ms.net"
xmlns:xns5="http://systinet.com/wsd/default"/>
    <xsd:import namespace="http://systinet.com/wsd/default"/>
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:element name="AXID" type="xsd:long"/>
    <xsd:element name="HWFingerprint" type="xsd:string" nillable="true"/>
    <xsd:element name="AXTID" type="xsd:long"/>
    <xsd:element name="SWFingerprint" type="xsd:string" nillable="true"/>
    <xsd:element name="Time" type="xsd:string" nillable="true"/>
    <xsd:element name="LastFPPA" type="xsd:string" nillable="true"/>
    <xsd:element name="ListOfPA" type="xns4:ArrayList" nillable="true"/>
    <xsd:element name="AXDOM" type="xsd:long"/>
    <xsd:element name="VerificationResult_Response" type="xns5:VerificationResult" nillable="true"/>
    <xsd:element name="HWFingerprint_1" type="xsd:long"/>
    <xsd:element name="AXRTID" type="xsd:long"/>
    <xsd:element name="RegDeadline" type="xsd:string" nillable="true"/>
    <xsd:element name="CertificationResult_Response" type="xns5:CertificationResult"
nillable="true"/>
</xsd:schema>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
        <xsd:sequence>
            <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
        <xsd:complexContent mixed="false">
            <xsd:extension base="tns:List"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="ArrayOfAnyType">
        <xsd:sequence>
            <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>

</wsdl:types>

<wsdl:message name="CertificationAndVerification_verify_1_Request">
    <wsdl:part name="AXID" element="ns0:AXID"/>
    <wsdl:part name="HWFingerprint" element="ns0:HWFingerprint"/>
    <wsdl:part name="AXTID" element="ns0:AXTID"/>
    <wsdl:part name="SWFingerprint" element="ns0:SWFingerprint"/>
    <wsdl:part name="Time" element="ns0:Time"/>
    <wsdl:part name="LastFPPA" element="ns0:LastFPPA"/>
    <wsdl:part name="ListOfPA" element="ns0:ListOfPA"/>
    <wsdl:part name="AXDOM" element="ns0:AXDOM"/>
</wsdl:message>

<wsdl:message name="CertificationAndVerification_verifyUser_Response">
    <wsdl:part name="response" element="ns0:VerificationResult_Response"/>
</wsdl:message>

<wsdl:message name="CertificationAndVerification_verifyUser_1_Request">
    <wsdl:part name="AXID" element="ns0:AXID"/>
    <wsdl:part name="AXDOM" element="ns0:AXDOM"/>
</wsdl:message>

```

| | |
|--|---|
| | <pre> <wsdl:message name="CertificationAndVerification_verify_Response"> <wsdl:part name="response" element="ns0:VerificationResult_Response"/> </wsdl:message> <wsdl:message name="CertificationAndVerification_certify_Response"> <wsdl:part name="response" element="ns0:CertificationResult_Response"/> </wsdl:message> <wsdl:message name="CertificationAndVerification_certify_1_Request"> <wsdl:part name="AXID" element="ns0:AXID"/> <wsdl:part name="HWFingerprint" element="ns0:HWFingerprint"/> <wsdl:part name="AXRTID" element="ns0:AXRTID"/> <wsdl:part name="SWFingerprint" element="ns0:SWFingerprint"/> <wsdl:part name="RegDeadline" element="ns0:RegDeadline"/> <wsdl:part name="AXDOM" element="ns0:AXDOM"/> </wsdl:message> <wsdl:portType name="CertificationAndVerification"> <wsdl:operation name="verify" parameterOrder="AXID HWFingerprint AXRTID SWFingerprint Time LastFPPA ListOfPA AXDOM"> <wsdl:input message="tns:CertificationAndVerification_verify_1_Request"/> <wsdl:output message="tns:CertificationAndVerification_verify_Response"/> </wsdl:operation> <wsdl:operation name="verifyUser" parameterOrder="AXID AXDOM"> <wsdl:input message="tns:CertificationAndVerification_verifyUser_1_Request"/> <wsdl:output message="tns:CertificationAndVerification_verifyUser_Response"/> </wsdl:operation> <wsdl:operation name="certify" parameterOrder="AXID HWFingerprint AXRTID SWFingerprint RegDeadline AXDOM"> <wsdl:input message="tns:CertificationAndVerification_certify_1_Request"/> <wsdl:output message="tns:CertificationAndVerification_certify_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="CertificationAndVerification" type="tns:CertificationAndVerification"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/> <wsdl:operation name="verify"> <map:java-operation name="verify" signature="KEpMamF2YS9sYW5nL1N0cmZtKTGphdmEvdGFuZy9TdHJpbmc7TGphdmEvdGFuZy9TdHJp bmc7TGphdmEvdGFuZy9TdHJpbmc7TGphdmEvdXRpbC9BcnJheUxpc3Q7SiIMVmVyaWZpY2F0aW9uUmVz dWx0Ow==" /> <soap:operation soapAction="urn:CertificationAndVerificationCertificationAndVerification#verify#KEpMamF2YS9sYW5nL1N0c mluZtKTGphdmEvdGFuZy9TdHJpbmc7TGphdmEvdGFuZy9TdHJpbmc7TGphdmEvdGFuZy9TdHJpbmc7TG phdmEvdXRpbC9BcnJheUxpc3Q7SiIMVmVyaWZpY2F0aW9uUmVzdWx0Ow==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyUser"> <map:java-operation name="verifyUser" signature="KEpKKUxWZXJpZmljYXRpb25SZXN1bHQ7" /> <soap:operation soapAction="urn:CertificationAndVerificationCertificationAndVerification#verifyUser#KEpKKUxWZXJpZmljYXR pb25SZXN1bHQ7" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="certify"> <map:java-operation name="certify" signature="KEpMamF2YS9sYW5nL1N0cmZtKTGphdmEvdGFuZy9TdHJpbmc7TGphdmEvdGFuZy9TdHJp bmc7SiIMQ2VydGhmaWNhdGlvbJlJlc3VsdDs=" /> </pre> |
|--|---|

| | <pre> <soap:operation soapAction="urn:CertificationAndVerificationCertificationAndVerification#certify#KEpMamF2YS9sYW5nL1N0c mluZztKTGphdmEvbGFuZy9TdHJpbmc7TGphdmEvbGFuZy9TdHJpbmc7SiIMQ2VydGhmaWNhdGlvbJlcl3Vsd Ds=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="CertificationAndVerification"> <wsdl:port name="CertificationAndVerification" binding="tns:CertificationAndVerification"> <soap:address location="http://efpc081:6060/CertificationAndVerification"/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|-------------------------|--|
| Method Description | |
| Method | verifyUser |
| Description | This method can be used to verify the status of a user, optionally inside a domain |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:ax.CertificationAndVerification" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXID i:type="d:long">1000</wn0:AXID> <wn0:AXDOM i:type="d:long">1010</wn0:AXDOM> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:ax.CertificationAndVerification" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:VerificationResult_Response i:type="wn1:VerificationResult"> <wn1:verificationResult i:type="d:int">0</wn1:verificationResult> </wn0:VerificationResult_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:long AXID: identifier of the AXMEDIS user</p> <p>xsd:long AXDOM: AXMEDIS domain of certified user (if any)</p> |
| Output parameters | <p>VerificationResult complex type formed by sequence of:</p> <p>xsd:int verificationResult, which indicates the result of the verification, according to the following numeration:</p> <ul style="list-style-type: none"> 0: Verification OK -1: User is not registered (in Domain, if present) -2: User is blocked |
| Method | Certify |
| Description | This method is called by the Protection Processor, through the axcsProxy. It is used to certify that the original tool has not been modified and to activate it. It creates a new entry in the CerTools table of the AXCS database associated to the tool user. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:ax.CertificationAndVerification" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXID i:type="d:long">1000</wn0:AXID> <wn0:HWFingerprint i:type="d:string">myHWFingerprint</wn0:HWFingerprint> <wn0:AXRTID i:type="d:long">343434</wn0:AXRTID> <wn0:SWFingerprint i:type="d:string">mySWFingerprint</wn0:SWFingerprint> </e:Body> </e:Envelope> </pre> |

| | |
|-------------------------|--|
| | <pre> <wn0:RegDeadline i:type="d:string">date</wn0:RegDeadline> <wn0:AXDOM i:type="d:long">1010</wn0:AXDOM> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:ax.CertificationAndVerification" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:CertificationResult_Response i:type="wn1:CertificationResult"> <wn1:AXTID i:type="d:long">0</wn1:AXTID> <wn1:certificationResult i:type="d:int">0</wn1:certificationResult> <wn1:enablingCode i:type="d:long">0</wn1:enablingCode> </wn0:CertificationResult_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:long AXID: identifier of the AXMEDIS user</p> <p>xsd:string HWFingerprint: hardware fingerprint of the device</p> <p>xsd:long AXRTID: identifier of the related registered tool (ID of the class of the registered tool given by SuperAXCS)</p> <p>xsd:string SWFingerprint: software fingerprint of the current state of a tool installed on a device.</p> <p>xsd:string RegDeadline: timestamp of certified tool registration end</p> <p>xsd:long AXDOM: AXMEDIS domain of certified tool (if any)</p> |
| Output parameters | <p>CertificationResult complex type formed by sequence of:</p> <ul style="list-style-type: none"> xsd:long AXTID, the identifier of the installed tool associated to a user and device xsd:int certificationResult, which indicates the result of the certification, according to the following numeration: <ul style="list-style-type: none"> 0: Certification OK -1: User is not registered (in Domain, if present) -2: User is blocked -3: Registered Tool is blocked -4: SWFingerprint mismatch. Original Tool has been manipulated -5: RegDeadline has expired xsd:long enablingCode, the tool activation code sent to the Protection Processor |
| Method | verify |
| Description | This method is called by the Protection Processor, through the axcsProxy. It is used to verify that the tool installed on a device has not been modified and that the user is not blocked to use that tool on the device. It is also responsible for resynchronizing the offline tool operation through the AXSupervisor. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:ax.CertificationAndVerification" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXID i:type="d:long">1000</wn0:AXID> <wn0:HWFingerprint i:type="d:string">myHWFingerprint</wn0:HWFingerprint> <wn0:AXTID i:type="d:long">34343434</wn0:AXTID> <wn0:SWFingerprint i:type="d:string">mySWFingerprint</wn0:SWFingerprint> <wn0:Time i:type="d:string">time</wn0:Time> <wn0:LastFPPA i:type="d:string">LastFPPA</wn0:LastFPPA> <wn0:ListOfPA i:type="wn2:ArrayList"> <wn2:Items> <wn2:Item i:type="wn0:ActionLog"> <wn0:AXCID i:nil="true"/> </pre> |

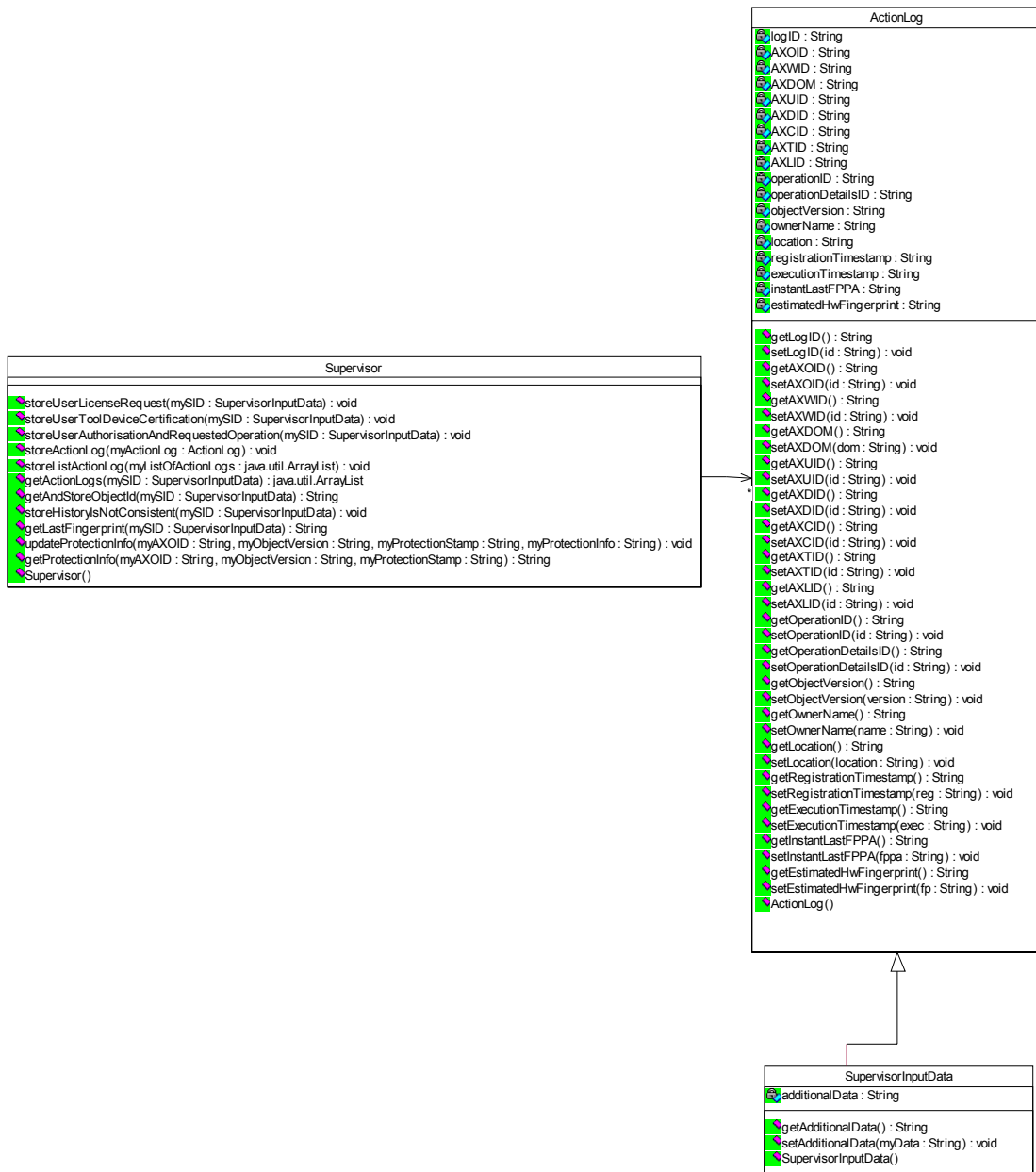
| | |
|-------------------------|---|
| | <pre> <wn0:AXDID i:nil="true"/> <wn0:AXDOM i:nil="true"/> <wn0:AXLID i:nil="true"/> <wn0:AXOID i:nil="true"/> <wn0:AXTID i:nil="true"/> <wn0:AXUID i:nil="true"/> <wn0:AXWID i:nil="true"/> <wn0:estimatedHwFingerprint i:nil="true"/> <wn0:executionTimestamp i:nil="true"/> <wn0:instantLastFPPA i:nil="true"/> <wn0:location i:nil="true"/> <wn0:logID i:nil="true"/> <wn0:objectVersion i:nil="true"/> <wn0:operationDetailsID i:nil="true"/> <wn0:operationID i:nil="true"/> <wn0:ownerName i:nil="true"/> <wn0:registrationTimestamp i:nil="true"/> </wn2:Item> <wn2:Item i:type="wn0:ActionLog"> <wn0:AXCID i:nil="true"/> <wn0:AXDID i:nil="true"/> <wn0:AXDOM i:nil="true"/> <wn0:AXLID i:nil="true"/> <wn0:AXOID i:nil="true"/> <wn0:AXTID i:nil="true"/> <wn0:AXUID i:nil="true"/> <wn0:AXWID i:nil="true"/> <wn0:estimatedHwFingerprint i:nil="true"/> <wn0:executionTimestamp i:nil="true"/> <wn0:instantLastFPPA i:nil="true"/> <wn0:location i:nil="true"/> <wn0:logID i:nil="true"/> <wn0:objectVersion i:nil="true"/> <wn0:operationDetailsID i:nil="true"/> <wn0:operationID i:nil="true"/> <wn0:ownerName i:nil="true"/> <wn0:registrationTimestamp i:nil="true"/> </wn2:Item> </wn2:Items> </wn0:ListOfPA> <wn0:AXDOM i:type="d:long">1010</wn0:AXDOM> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes" xmlns:wn1="urn:ax.CertificationAndVerification" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:VerificationResult_Response i:type="wn1:VerificationResult"> <wn1:verificationResult i:type="d:int">0</wn1:verificationResult> </wn0:VerificationResult_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:long AXID: identifier of the AXMEDIS user</p> <p>xsd:string HWFingerprint: hardware fingerprint of the device</p> <p>xsd:long AXTID: identifier of the certified tool (the single instance of the tool installed on a device)</p> <p>xsd:string SWFingerprint: software fingerprint of the current state of a tool installed on a device.</p> <p>xsd:string Time: time and date when the verification request is made</p> <p>xsd:string LastFPPA: last fingerprint of performed actions (last fingerprint of action log history)</p> <p>xns4:ArrayList ListOfPA: complex type formed by sequence of: ActionLog, which is a complex type defined in AXMEDIS Supervisor</p> |

| | |
|-------------------|---|
| | xsd:long AXDOM: AXMEDIS domain of certified tool (if any) |
| Output parameters | <p>VerificationResult complex type formed by sequence of:</p> <p>xsd:int verificationResult, which indicates the result of the verification, according to the following numeration:</p> <ul style="list-style-type: none"> 0: Verification OK -1: User is not registered (in Domain, if present) -2: User is blocked -3: AXTID does not exist -4: Installed Tool is blocked -5: SWFingerprint mismatch. Installed Tool has been manipulated -6: HWFingerprint (device) mismatch -7: LastFPPA not consistent |

2.4 AXMEDIS Supervisor (FUPF)

2.4.1 Architecture of the module

Next figure shows the architecture of the AXMEDIS supervisor module.



AXMEDIS Supervisor module class diagram

2.4.2 Difference between Supervisor Input Data and Action Log

The difference between Action Log and Supervisor Input Data is that the first one is directly related with the AXMEDIS Object (it is created when any action over the object is done) and Supervisor Input Data is not directly related with the object (it is not created when an action over the object is done, but it can refer to the object). The Supervisor Input Data is created to notify to Supervisor any event in (or between) modules.

Both of them mostly have the structure of the MPEG-21 Event Report, but in Supervisor Input Data there are some empty (or deleted for unnecessary) fields.

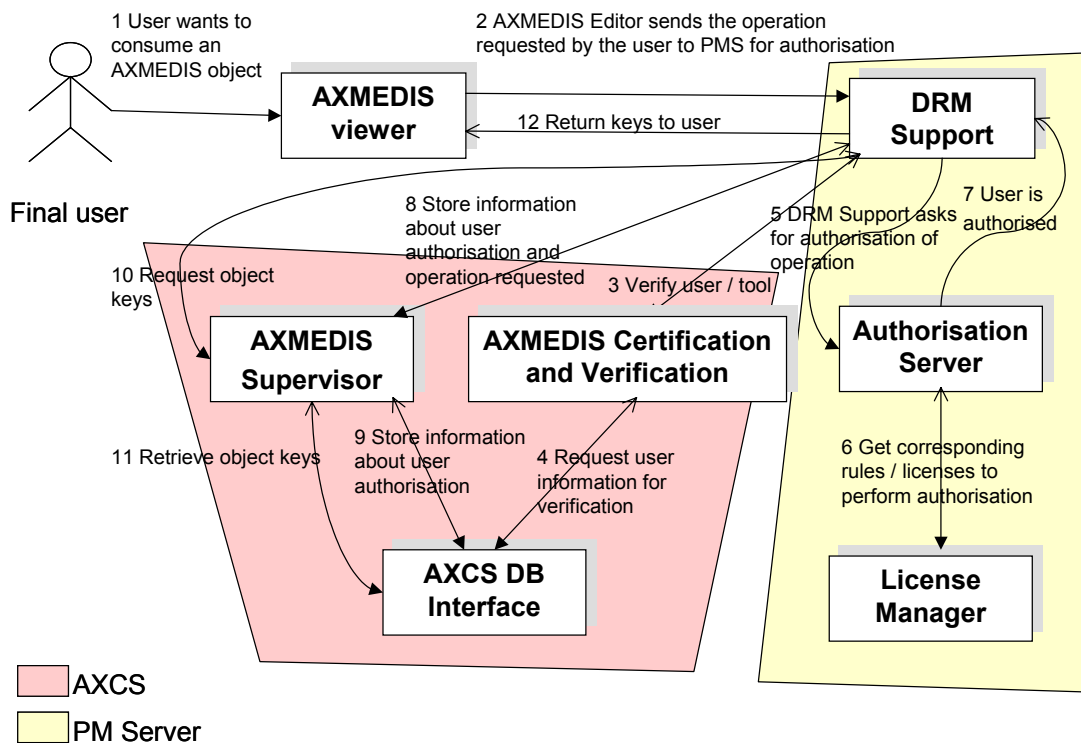
The objective of to distinguish between both of them is to clarify the notifications that comes from the object (actions done over the object) and the notifications between modules.

2.4.3 Supervisor Scenarios

The following scenarios show the relationship between the AXMEDIS Supervisor module and other modules inside AXCS.

2.4.3.1 Successful consumption of a protected AXMEDIS object

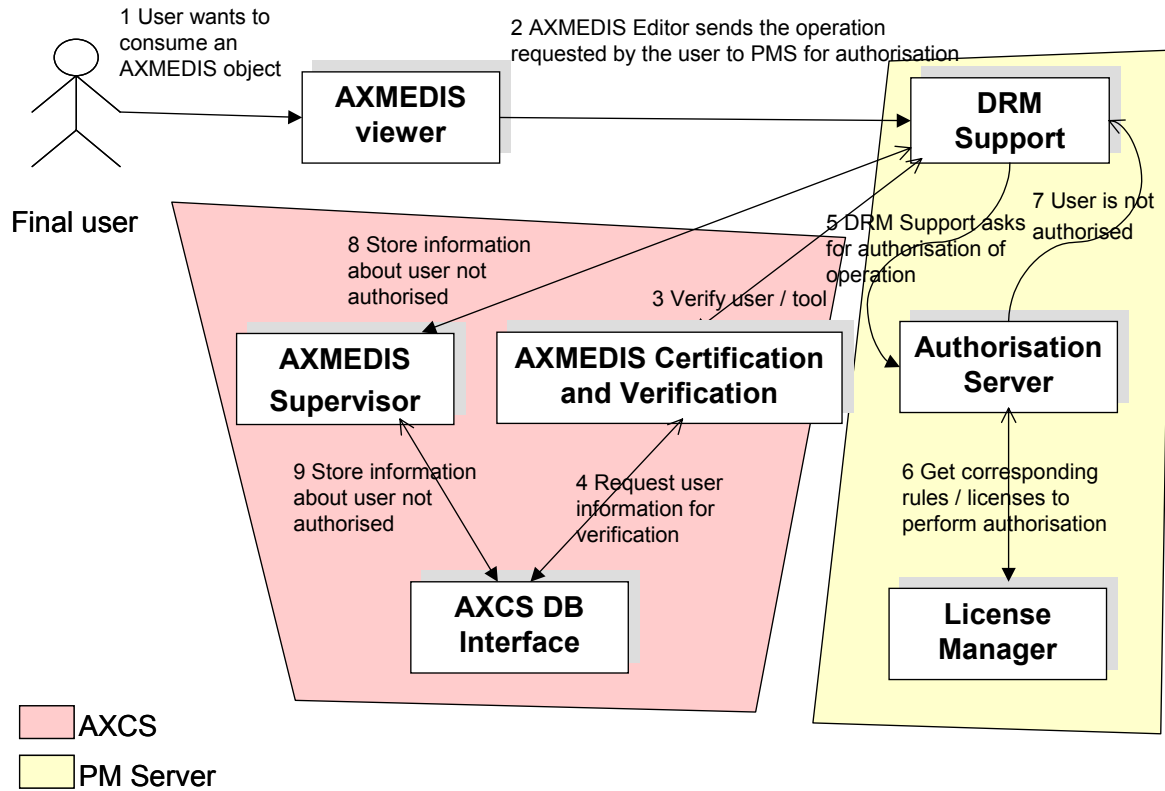
The scenario represents a successful consumption of a protected AXMEDIS object from the AXCS Supervisor module point of view. This scenario also involves the verification of the user and the tool by the AXMEDIS Certification and Verification module.



Successful consumption of a protected AXMEDIS object

2.4.3.2 User not authorized scenario

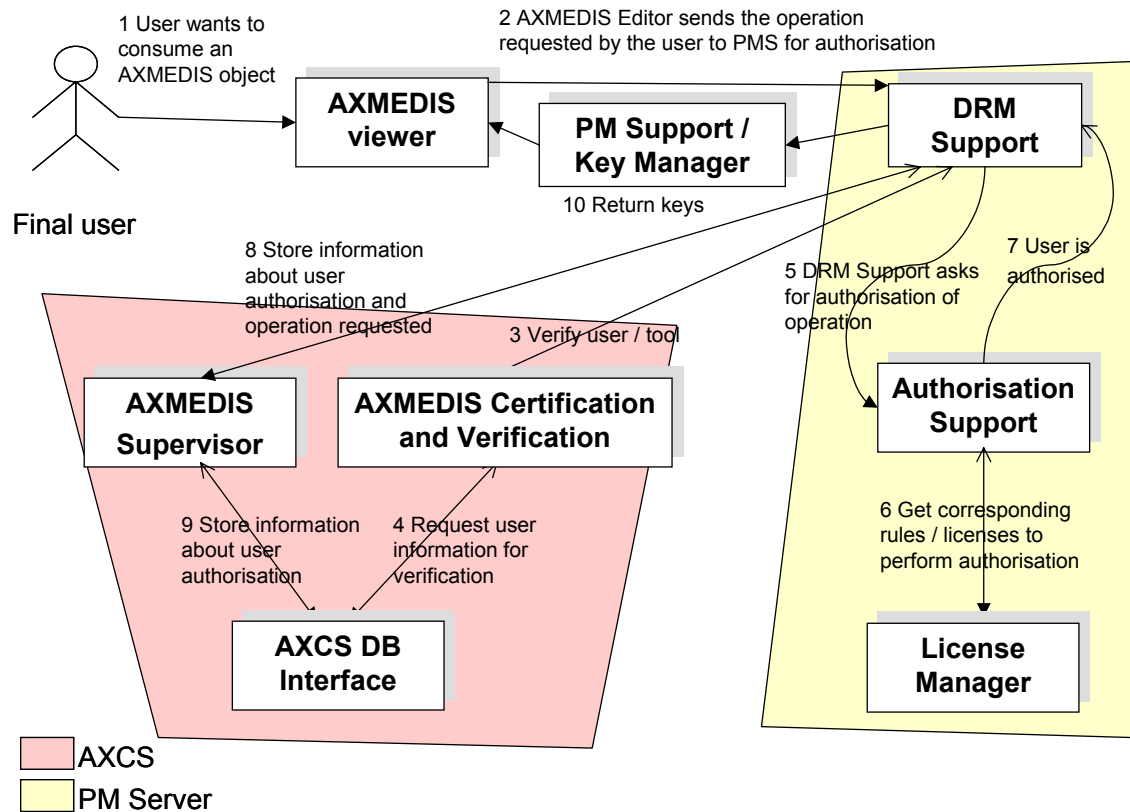
The scenario represents a user not authorized scenario from the AXCS Supervisor module point of view. This scenario also involves the verification of the user and the tool by the AXMEDIS Certification and Verification module.



User not authorized scenario

2.4.3.3 Object key cached scenario

The scenario represents a successful content consumption of a protected AXMEDIS object where the AXMEDIS object keys are cached in the client application. This scenario also involves the verification of the user and the tool by the AXMEDIS Certification and Verification module.



Object keys cached scenario

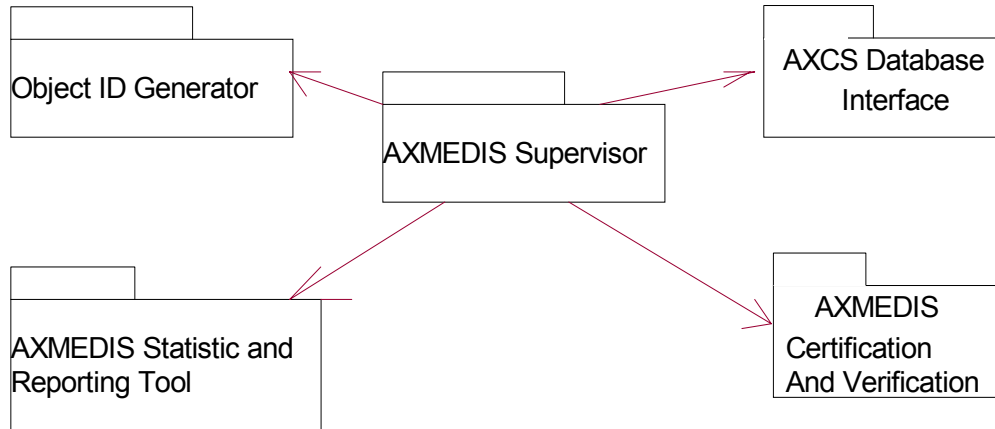
2.4.4 AXMEDIS Supervisor Logics (FUPF)

The AXMEDIS Supervisor is a web service running on a server and it is implemented in Java.

It communicates with the following modules:

- AXCS Database Interface. AXS stores, retrieves and updates the information related to any action made by the users, devices and tools. The communication between AXS and AXCS Database Interface is made the interfaces provided as web services
- AXMEDIS Certification and Verification. AXS is requested by AXCv to resynchronize the actions performed during the offline operation. The communication between AXS and AXCv is made through a web service provided by AXS and over a secure channel.
- AXMEDIS Statistic and Reporting Tool. A Distributor wants to recover information on actions performed on the objects he has rights. AXMEDIS Statistic and reporting tools queries AXS. AXS extracts the required Action-Logs from the AXCS DB Interface and communicates them to the AXMEDIS Statistic and Reporting Tool.

- Object ID Generator. A Content Provider wants to create a new AXMEDIS Object and the tool with that is creating the object requests to the AXS an object ID. AXS interact with Object ID generator to ask an ID. OID generator returns back to AXS the generated ID and AXS stores Object ID and related info in the AXDB and communicates to the tool the generated ID.



2.4.5 AXMEDIS Supervisor WEB Service formalisation (FUPF)

| Supervisor | |
|---|--|
| Methods | Description |
| storeUserLicenseRequest | This function is used to store information about user license when an AXMEDIS user wants to buy a license to consume an AXMEDIS object (on-line or off-line) and the information about license request is sent to Supervisor. |
| storeUserToolDeviceCertification | This function is used to receive the information about user/tool/device certification, to store it and to send it to the AXCS DB. When an end User requests to perform an action on an AXMEDIS Protected Object the DMS Support certifies the user/tool/device (positively or negatively) and sends to Supervisor the information (positive or negative) about user/tool/device certification. Once this information is received by Supervisor, this sends it to the AXCS DB Interface. |
| storeUserAuthorisationAndRequestedOperation | This function is used to store the information about user authorisation and requested operation (received from DMS Support) and to send the user authorisation to the AXCS DB Interface. When an end user requests to perform an action on an AXMEDIS Object, DMS Support authorises the user and sends to Supervisor the information about user authorisation. Then the Supervisor stores information about user authorisation and requested operation and sends the user authorisation to the AXCS DB Interface. |
| storeActionLog | This function is used to receive the Action Logs and to store them. When an end User requests to perform an action on an AXMEDIS Protected Object and has been certified, verified and authorised, PMS sends to AXCS the action performed (Action Log) and Action Logs are stored by Supervisor. An AXMEDIS user wants to consume an AXMEDIS Object and requests to perform an action on an AXMEDIS Protected Object. Then the user is certified and authorised and the PMS sends to AXCS the action performed. |
| storeListActionLog | This function is a particular case of the function before. In this case the PMS sends to Supervisor a list of Action Logs, not only one. It happens when an user has been off-line for a period of time and has done some actions over an object off-line. When he connects again with the system, the system and the cache of |

| | |
|-----------------------------|--|
| | the user have to be synchronised, for what the list of Action Logs is sent to Supervisor. |
| getActionLogs | This function is used to extract Action Logs (from the AXCS DB Interface) and to send it to the AXMEDIS Accounting or reporting tools. When a Distributor wants to recover information on actions performed on the objects he has rights, AXMEDIS Statistic or reporting tools queries AXCS (or SuperAXCS, who recover information from the different AXCSs). Then AXCS extracts the required Action-Logs from the AXCS DB Interface and communicates them to the tools that perform actions to return results in the desired form (These are AXMEDIS Statistic or reporting tools). |
| getObjectId | This function is used to communicate an Object ID when a Content Provider wants to create a new AXMEDIS Object. When a Content Provider wants to create a new AXMEDIS Object, the tool with that is creating the object requests to the AXCS an object ID. Then AXCS interact with Object ID generator to ask an ID and OID generator returns back to AXCS the generated ID. Once the AXCS has the information communicates to the tool the generated ID. |
| storeHistoryIsNotConsistent | This function is used to store that history is not consistent and to block user and/or tool and/or device due to an unsuccessful verification of an AXMEDIS user. When an end User requests to perform an action on an AXMEDIS Object and the verification of the user/tool/device is unsuccessful, the information about unsuccessful verification is sent to Supervisor. The Supervisor stores that history is not consistent and controls each unsuccessful verification and then blocks user and/or tool and/or device. |
| getLastFingerprint | This function is used to ask to Supervisor for the Last Fingerprint of an user or an object or a tool in order to certify or verify any user. |
| getProtectionInfo | This function is used to extract the protection information related to an object from the Objects Table in the AXCS Objects ID Database. |
| updateProtectionInfo | This function is used to update the protection information related to an object in the Objects Table in the AXCS Objects ID Database. |

| Supervisor | |
|------------|---|
| WSDL | <pre> <?xml version="1.0" encoding="UTF-8"?> <wsdl:definitions name="AxmedisSupervisor.Supervisor" targetNamespace="urn:AxmedisSupervisor.Supervisor" xmlns:tns="urn:AxmedisSupervisor.Supervisor" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayList"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="ArrayOfAnyType"> <xsd:sequence> <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" </pre> |


```

xmlns:xns4="urn:AxmedisSupervisor.Supervisor" xmlns:xns5="http://systinet.com/containers/literal/ms.net">
  <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
  <xsd:import namespace="urn:AxmedisSupervisor.Supervisor"/>
  <xsd:element name="mySID" type="xns4:SupervisorInputData" nillable="true"/>
  <xsd:element name="myActionLog" type="xns4:ActionLog" nillable="true"/>
  <xsd:element name="myListOfActionLogs" type="xns5:ArrayList" nillable="true"/>
  <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
  <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
  <xsd:element name="myAXOID" type="xsd:string" nillable="true"/>
  <xsd:element name="myObjectVersion" type="xsd:string" nillable="true"/>
  <xsd:element name="myProtectionStamp" type="xsd:string" nillable="true"/>
  <xsd:element name="myProtectionInfo" type="xsd:string" nillable="true"/>
</xsd:schema>
<xsd:schema targetNamespace="urn:AxmedisSupervisor.Supervisor" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:AxmedisSupervisor.Supervisor"
xmlns:map="http://systinet.com/mapping/">
  <xsd:complexType name="SupervisorInputData">
    <xsd:annotation>
      <xsd:appinfo>
        <map:java-type name="AxmedisSupervisor.SupervisorInputData"/>
      </xsd:appinfo>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="tns:ActionLog">
        <xsd:sequence>
          <xsd:element name="additionalData" type="xsd:string" nillable="true"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="ActionLog">
    <xsd:annotation>
      <xsd:appinfo>
        <map:java-type name="AxmedisSupervisor.ActionLog"/>
      </xsd:appinfo>
    </xsd:annotation>
    <xsd:sequence/>
  </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="Supervisor_getLastFingerprint_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeListActionLog_Response"/>
<wsdl:message name="Supervisor_storeUserToolDeviceCertification_1_Request">
  <wsdl:part name="mySID" element="ns0:mySID"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeUserAuthorisationAndRequestedOperation_1_Request">
  <wsdl:part name="mySID" element="ns0:mySID"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeHistoryIsNotConsistent_1_Request">
  <wsdl:part name="mySID" element="ns0:mySID"/>
</wsdl:message>
<wsdl:message name="Supervisor_getActionLogs_1_Request">
  <wsdl:part name="mySID" element="ns0:mySID"/>
</wsdl:message>
<wsdl:message name="Supervisor_updateProtectionInfo_Response"/>
<wsdl:message name="Supervisor_storeActionLog_1_Request">
  <wsdl:part name="myActionLog" element="ns0:myActionLog"/>
</wsdl:message>
<wsdl:message name="Supervisor_getAndStoreObjectId_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="Supervisor_getProtectionInfo_1_Request">
  <wsdl:part name="myAXOID" element="ns0:myAXOID"/>
  <wsdl:part name="myObjectVersion" element="ns0:myObjectVersion"/>
  <wsdl:part name="myProtectionStamp" element="ns0:myProtectionStamp"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeActionLog_Response"/>
<wsdl:message name="Supervisor_getLastFingerprint_1_Request">

```

```

<wsdl:part name="mySID" element="ns0:mySID"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeUserLicenseRequest_1_Request">
  <wsdl:part name="mySID" element="ns0:mySID"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeUserToolDeviceCertification_Response"/>
<wsdl:message name="Supervisor_updateProtectionInfo_1_Request">
  <wsdl:part name="myAXOID" element="ns0:myAXOID"/>
  <wsdl:part name="myObjectVersion" element="ns0:myObjectVersion"/>
  <wsdl:part name="myProtectionStamp" element="ns0:myProtectionStamp"/>
  <wsdl:part name="myProtectionInfo" element="ns0:myProtectionInfo"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeListActionLog_1_Request">
  <wsdl:part name="myListOfActionLogs" element="ns0:myListOfActionLogs"/>
</wsdl:message>
<wsdl:message name="Supervisor_getProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeHistoryIsNotConsistent_Response"/>
<wsdl:message name="Supervisor_storeUserAuthorisationAndRequestedOperation_Response"/>
<wsdl:message name="Supervisor_getActionLogs_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="Supervisor_getAndStoreObjectId_1_Request">
  <wsdl:part name="mySID" element="ns0:mySID"/>
</wsdl:message>
<wsdl:message name="Supervisor_storeUserLicenseRequest_Response"/>
<wsdl:portType name="Supervisor">
  <wsdl:operation name="storeUserLicenseRequest" parameterOrder="mySID">
    <wsdl:input message="tns:Supervisor_storeUserLicenseRequest_1_Request"/>
    <wsdl:output message="tns:Supervisor_storeUserLicenseRequest_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeUserToolDeviceCertification" parameterOrder="mySID">
    <wsdl:input message="tns:Supervisor_storeUserToolDeviceCertification_1_Request"/>
    <wsdl:output message="tns:Supervisor_storeUserToolDeviceCertification_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeUserAuthorisationAndRequestedOperation" parameterOrder="mySID">
    <wsdl:input message="tns:Supervisor_storeUserAuthorisationAndRequestedOperation_1_Request"/>
    <wsdl:output message="tns:Supervisor_storeUserAuthorisationAndRequestedOperation_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeActionLog" parameterOrder="myActionLog">
    <wsdl:input message="tns:Supervisor_storeActionLog_1_Request"/>
    <wsdl:output message="tns:Supervisor_storeActionLog_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeListActionLog" parameterOrder="myListOfActionLogs">
    <wsdl:input message="tns:Supervisor_storeListActionLog_1_Request"/>
    <wsdl:output message="tns:Supervisor_storeListActionLog_Response"/>
  </wsdl:operation>
  <wsdl:operation name="getActionLogs" parameterOrder="mySID">
    <wsdl:input message="tns:Supervisor_getActionLogs_1_Request"/>
    <wsdl:output message="tns:Supervisor_getActionLogs_Response"/>
  </wsdl:operation>
  <wsdl:operation name="getAndStoreObjectId" parameterOrder="mySID">
    <wsdl:input message="tns:Supervisor_getAndStoreObjectId_1_Request"/>
    <wsdl:output message="tns:Supervisor_getAndStoreObjectId_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeHistoryIsNotConsistent" parameterOrder="mySID">
    <wsdl:input message="tns:Supervisor_storeHistoryIsNotConsistent_1_Request"/>
    <wsdl:output message="tns:Supervisor_storeHistoryIsNotConsistent_Response"/>
  </wsdl:operation>
  <wsdl:operation name="getLastFingerprint" parameterOrder="mySID">
    <wsdl:input message="tns:Supervisor_getLastFingerprint_1_Request"/>
    <wsdl:output message="tns:Supervisor_getLastFingerprint_Response"/>
  </wsdl:operation>
  <wsdl:operation name="updateProtectionInfo" parameterOrder="myAXOID myObjectVersion myProtectionStamp myProtectionInfo">
    <wsdl:input message="tns:Supervisor_updateProtectionInfo_1_Request"/>
    <wsdl:output message="tns:Supervisor_updateProtectionInfo_Response"/>
  </wsdl:operation>
  <wsdl:operation name="getProtectionInfo" parameterOrder="myAXOID myObjectVersion

```

| | |
|--|--|
| | <pre> myProtectionStamp"> <wsdl:input message="tns:Supervisor_getProtectionInfo_1_Request"/> <wsdl:output message="tns:Supervisor_getProtectionInfo_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="Supervisor" type="tns:Supervisor"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/> <wsdl:operation name="storeUserLicenseRequest"> <map:java-operation name="storeUserLicenseRequest" signature="KExBeG1IZGizU3VwZXJ2aXNvci9TdXBicnZpc29ySW5wdXREYXRhOylW"/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#storeUserLicenseRequest#KExBeG1IZGizU3VwZXJ2aXNvci9TdXBicnZpc29ySW5wdXREYXRhOylW" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeUserToolDeviceCertification"> <map:java-operation name="storeUserToolDeviceCertification" signature="KExBeG1IZGizU3VwZXJ2aXNvci9TdXBicnZpc29ySW5wdXREYXRhOylW"/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#storeUserToolDeviceCertification#KExBeG1IZGizU3VwZXJ2aXNvci9TdXBicnZpc29ySW5wdXREYXRhOylW" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeUserAuthorisationAndRequestedOperation"> <map:java-operation name="storeUserAuthorisationAndRequestedOperation" signature="KExBeG1IZGizU3VwZXJ2aXNvci9TdXBicnZpc29ySW5wdXREYXRhOylW"/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#storeUserAuthorisationAndRequestedOperation#KExBeG1IZGizU3VwZXJ2aXNvci9TdXBicnZpc29ySW5wdXREYXRhOylW" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeActionLog"> <map:java-operation name="storeActionLog" signature="KExBeG1IZGizU3VwZXJ2aXNvci9BY3Rpb25Mb2c7KVY="/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#storeActionLog#KExBeG1IZGizU3VwZXJ2aXNvci9BY3Rpb25Mb2c7KVY=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeListActionLog"> <map:java-operation name="storeListActionLog" signature="KExqYXZhL3V0aWwvQXJyYXlMaXN0OylW"/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#storeListActionLog#KExqYXZhL3V0aWwvQXJyYXlMaXN0OylW" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </pre> |
|--|--|

| | |
|--|---|
| | <pre></wsdl:output> </wsdl:operation> <wsdl:operation name="getActionLogs"> <map.java-operation name="getActionLogs" signature="KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylMamF2YS91dGlSL0FycmF5TGldDs="/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#getActionLogs#KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylMamF2YS91dGlSL0FycmF5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="getAndStoreObjectId"> <map.java-operation name="getAndStoreObjectId" signature="KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylMamF2YS9sYW5nL1N0cmlyZzs="/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#getAndStoreObjectId#KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylMamF2YS9sYW5nL1N0cmlyZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeHistoryIsNotConsistent"> <map.java-operation name="storeHistoryIsNotConsistent" signature="KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylW"/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#storeHistoryIsNotConsistent#KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylW" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="getLastFingerprint"> <map.java-operation name="getLastFingerprint" signature="KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylMamF2YS9sYW5nL1N0cmlyZzs="/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#getLastFingerprint#KExBeG1IZGlzU3VwZXJ2aXNvci9TdXBlnZpc29ySW5wdXREYXRhOylMamF2YS9sYW5nL1N0cmlyZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="updateProtectionInfo"> <map.java-operation name="updateProtectionInfo" signature="KEXqYYZhL2xbmcvU3RyaW5nO0qxYXXzhL2xbmcvU3RyaW5nO0qxYXXzhL2xbmcvU3RyaW5nO0qxYZHl2xbmcvU3RyaW5nOylW"/> <soap:operation soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#updateProtectionInfo#KEXqYYZhL2xbmcvU3RyaW5nO0qxYXXzhL2xbmcvU3RyaW5nO0qxYXXzhL2xbmcvU3RyaW5nOylW" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output></pre> |
|--|---|

```

        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getProtectionInfo">
        <map:java-operation name="getProtectionInfo"
signature="KEXqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyIMa
mF2YS9sYW5nL1N0cmLuZzs="/>
        <soap:operation
soapAction="urn:AxmedisSupervisor.SupervisorSupervisor#getProtectionInfo#KEXqYXZhL2xhbmcvU3RyaW5nO0xq
YXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyIMamF2YS9sYW5nL1N0cmLuZzs="
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="Supervisor">
    <wsdl:port name="Supervisor" binding="tns:Supervisor">
        <soap:address location="http://EFPC016:6060/Supervisor"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

| | |
|-------------------------------|--|
| Method | storeUserLicenseRequest |
| Description | This function is used to store information about user license when an AXMEDIS user wants to buy a license to consume an AXMEDIS object (on-line or off-line) and the information about license request is sent to Supervisor |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/wsdl/default/" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:mySID i:type="wn0:SupervisorInputData"> <wn0:AXCID i:type="String">12</wn0:AXCID> <wn0:AXDID i:type="String">013</wn0:AXDID> <wn0:AXDOM i:type="String">myDomain </wn0:AXDOM> <wn0:AXLID i:type="String">013</wn0:AXLID> <wn0:AXOID i:type="String">987</wn0:AXOID> <wn0:AXTID i:type="String">mytool</wn0:AXTID> <wn0:AXUID i:type="String">887</wn0:AXUID> <wn0:AXWID i:type="String">mywork</wn0:AXWID> <wn0:estimatedHwFingerprint i:type="String">var1</wn0:estimatedHwFingerprint> <wn0:executionTimestamp i:type="String">08:00am</wn0:executionTimestamp> <wn0:instantLastFPPA i:type="String">05:00am</wn0:instantLastFPPA> <wn0:location i:type="String">Spain</wn0:location> <wn0:logID i:type="String">454</wn0:logID> <wn0:objectVersion i:type="String">val1</wn0:objectVersion> <wn0:operationDetailsID i:type="String">myoperationdetails</wn0:operationDetailsID> <wn0:operationID i:type="String">001</wn0:operationID> <wn0:ownerName i:type="String">JohnSmith</wn0:ownerName> <wn0:registrationTimestamp i:type="String">07:00am</wn0:registrationTimestamp> <wn0:additionalData i:type="String">licenserequest</wn0:additionalData> </wn1:mySID> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope></pre> |
| Input parameters | ax:SupervisorInputData mySID that is a complex type formed by (private attribute and get and set public operations): |

| | |
|--|---|
| | <p> xsd:string getAdditionalData setAdditionalData xsd:string data additionalData type="xsd:string" </p> <p>ax:SupervisorInputData extends ax:ActionLog</p> <p>ax:ActionLog is a complex type formed by (private attributes and public get and set public operations):</p> <p> AXCID type="xsd:string" AXDID type="xsd:string" AXDOM type="xsd:string" AXLID type="xsd:string" AXOID type="xsd:string" AXTID type="xsd:string" AXUID type="xsd:string" AXWID type="xsd:string" estimatedHwFingerprint type="xsd:string" executionTimestamp type="xsd:string" instantLastFPPA type="xsd:string" location type="xsd:string" logID type="xsd:string" objectVersion type="xsd:string" operationDetailsID type="xsd:string" operationID type="xsd:string" ownerName type="xsd:string" registrationTimestamp type="xsd:string" </p> <p> xsd:string getAXCID setAXCID xsd:string data xsd:string getAXDID setAXDID xsd:string data xsd:string getAXDOM setAXDOM xsd:string data xsd:string getAXLID setAXLID xsd:string data xsd:string getAXTCID setAXTID xsd:string data xsd:string getAXUID setAXUID xsd:string data xsd:string getAXWID setAXWID xsd:string data xsd:string getEstimatedHwFingerprint setAXEstimatedHwFingerprint xsd:string data xsd:string getExecutionTimestamp setAXExecutionTimestamp xsd:string data xsd:string getInstantLastFPPA setAXInstantLastFPPA xsd:string data xsd:string getLocation setAXLocation xsd:string data xsd:string getLogID setAXLogID xsd:string data xsd:string getObjectVersion </p> |
|--|---|

| | |
|-------------------------|---|
| | setAXObjectVersion xsd:string data xsd:string getOperationDetailsID setAXOperationDetailsID xsd:string data xsd:string getOperationID setAXOperationID xsd:string data xsd:string getOwnerName setAXOwnerName xsd:string data xsd:string getRegistrationTimestamp setAX xsd:string dataRegistrationTimestamp |
| Output parameters | None |
| Method | storeUserToolDeviceCertification |
| Description | This function is used to receive the information about user/tool/device certification, to store it and to send it to the AXCS DB. When an end User requests to perform an action on an AXMEDIS Protected Object the DMS Support certifies the user/tool/device (positively or negatively) and sends to Supervisor the information (positive or negative) about user/tool/device certification. Once this information is received by Supervisor, this sends it to the AXCS DB Interface. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/wsd/default/" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:mySID i:type="wn0:SupervisorInputData"> <wn0:AXCID>45</wn0:AXCID> <wn0:AXDID i:type="String">684</wn0:AXDID> <wn0:AXDOM i:type="String">myDomain</wn0:AXDOM> <wn0:AXLID i:type="String">665</wn0:AXLID> <wn0:AXOID i:type="String">145</wn0:AXOID> <wn0:AXTID>mytool</wn0:AXTID> <wn0:AXUID i:type="String">946</wn0:AXUID> <wn0:AXWID i:type="String">mywork</wn0:AXWID> <wn0:estimatedHwFingerprint i:type="String">var2</wn0:estimatedHwFingerprint> <wn0:executionTimestamp i:type="String">09:00am</wn0:executionTimestamp> <wn0:instantLastFPPA i:type="String">02:00am</wn0:instantLastFPPA> <wn0:location i:type="String">Spain</wn0:location> <wn0:logID i:type="String">645</wn0:logID> <wn0:objectVersion i:type="String">val2</wn0:objectVersion> <wn0:operationDetailsID i:type="String">myoperationdetails</wn0:operationDetailsID> <wn0:operationID i:type="String">002</wn0:operationID> <wn0:ownerName i:type="String">JohnSmith</wn0:ownerName> <wn0:registrationTimestamp i:type="String">08:00am</wn0:registrationTimestamp> <wn0:additionalData i:type="String">toolcertification</wn0:additionalData> </wn1:mySID> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope> </pre> |
| Input parameters | ax:SupervisorInputData mySID previously defined. |
| Output parameters | None |
| Method | storeUserAuthorisationAndRequestedOperation |
| Description | This function is used to store the information about user authorisation and requested operation (received from DMS Support) and to send the user authorisation to the AXCS DB Interface. When an end user requests to perform an action on an AXMEDIS Object, DMS Support authorises the user and sends to Supervisor the information about user authorisation. Then the Supervisor stores information about user authorisation and requested operation and sends the user authorisation to the |

| | |
|-------------------------|--|
| | AXCS DB Interface. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/wsdl/default/" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:mySID i:type="wn0:SupervisorInputData"> <wn0:AXCID i:type="String">192</wn0:AXCID> <wn0:AXDID i:type="String">023</wn0:AXDID> <wn0:AXDOM i:type="String">myDomain </wn0:AXDOM> <wn0:AXLID i:type="String">094</wn0:AXLID> <wn0:AXOID i:type="String">124</wn0:AXOID> <wn0:AXTID i:type="String">mytool</wn0:AXTID> <wn0:AXUID i:type="String">875</wn0:AXUID> <wn0:AXWID i:type="String">mywork</wn0:AXWID> <wn0:estimatedHwFingerprint i:type="String">var3</wn0:estimatedHwFingerprint> <wn0:executionTimestamp i:type="String">15:00am</wn0:executionTimestamp> <wn0:instantLastFPPA i:type="String">05:00am</wn0:instantLastFPPA> <wn0:location i:type="String">Spain</wn0:location> <wn0:logID i:type="String">997</wn0:logID> <wn0:objectVersion i:type="String">val3</wn0:objectVersion> <wn0:operationDetailsID i:type="String">myoperationdetails</wn0:operationDetailsID> <wn0:operationID i:type="String">008</wn0:operationID> <wn0:ownerName i:type="String">JohnSmith</wn0:ownerName> <wn0:registrationTimestamp i:type="String">10:00am</wn0:registrationTimestamp> <wn0:additionalData i:type="String">userauthorisation</wn0:additionalData> </wn1:mySID> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope> </pre> |
| Input parameters | ax:SupervisorInputData mySID previously defined. |
| Output parameters | None |
| Method | storeActionLog |
| Description | <p>This function is used to receive the Action Logs and to store them. When an end User requests to perform an action on an AXMEDIS Protected Object and has been certified, verified and authorised, PMS sends to AXCS the action performed (Action Log) and Action Logs are stored by Supervisor. This function is also used to retrieve the object keys from the AXCS DB Interface and to send them to DRM Support. An AXMEDIS user wants to consume an AXMEDIS Object and requests to perform an action on an AXMEDIS Protected Object. Then the user is certified and authorised, the PMS sends AXCS the action performed and DRM Support requests object keys to Supervisor. Supervisor retrieves the object keys from the AXCS DB Interface and gives back the keys to access the content (if necessary). This function allows to send the action log to Supervisor and asks for the object key to Supervisor in the same call of the function.</p> |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/wsdl/default/" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myActionLog i:type="wn0:ActionLog"> <wn0:AXCID i:type="String">812</wn0:AXCID> <wn0:AXDID i:type="String">213</wn0:AXDID> <wn0:AXDOM i:type="String">myDomain </wn0:AXDOM> <wn0:AXLID i:type="String">097</wn0:AXLID> <wn0:AXOID i:type="String">064</wn0:AXOID> <wn0:AXTID i:type="String">mytool</wn0:AXTID> <wn0:AXUID i:type="String">587</wn0:AXUID> <wn0:AXWID i:type="String">mywork</wn0:AXWID> <wn0:estimatedHwFingerprint i:type="String">var4</wn0:estimatedHwFingerprint> <wn0:executionTimestamp i:type="String">07:00am</wn0:executionTimestamp> </wn1:myActionLog> </e:Body> </e:Envelope> </pre> |

| | |
|-------------------------|--|
| | <pre> <wn0:instantLastFPPA i:type="String">01:00am</wn0:instantLastFPPA> <wn0:location i:type="String">France</wn0:location> <wn0:logID i:type="String">494</wn0:logID> <wn0:objectVersion i:type="String">val4</wn0:objectVersion> <wn0:operationDetailsID i:type="String">myoperationdetails</wn0:operationDetailsID> <wn0:operationID i:type="String">006</wn0:operationID> <wn0:ownerName i:type="String">JohnSmith</wn0:ownerName> <wn0:registrationTimestamp i:type="String">07:00am</wn0:registrationTimestamp> </wn1:myActionLog> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope> </pre> |
| Input parameters | ax:ActionLog myActionLog previously defined. |
| Output parameters | None |
| Method | storeListActionLog |
| Description | This function is a particular case of the function before. In this case the PMS sends to Supervisor a list of Action Logs, not only one. It happens when an user has been off-line for a period of time and has done some actions over an object off-line. When he connects again with the system, the system and the cache of the user have to be synchronised, for what the list of Action Logs is sent to Supervisor. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/wsdl/default/" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myListOfActionLogs i:type="wn2:ArrayList"> <wn2:Items> <wn2:Item i:type="wn0:ActionLog"> <wn0:AXCID i:type="String">087</wn0:AXCID> <wn0:AXDID i:type="String">099</wn0:AXDID> <wn0:AXDOM i:type="String">myDomain </wn0:AXDOM> <wn0:AXLID i:type="String">111</wn0:AXLID> <wn0:AXOID i:type="String">668</wn0:AXOID> <wn0:AXTID i:type="String">mytool</wn0:AXTID> <wn0:AXUID i:type="String">946</wn0:AXUID> <wn0:AXWID i:type="String">mywork</wn0:AXWID> <wn0:estimatedHwFingerprint i:type="String">var5</wn0:estimatedHwFingerprint> <wn0:executionTimestamp i:type="String">06:00am</wn0:executionTimestamp> <wn0:instantLastFPPA i:type="String">07:00am</wn0:instantLastFPPA> <wn0:location i:type="String">Spain</wn0:location> <wn0:logID i:type="String">458</wn0:logID> <wn0:objectVersion i:type="String">val5</wn0:objectVersion> <wn0:operationDetailsID i:type="String">myoperationdetails</wn0:operationDetailsID> <wn0:operationID i:type="String">008</wn0:operationID> <wn0:ownerName i:type="String">JohnSmith</wn0:ownerName> <wn0:registrationTimestamp i:type="String">07:00am</wn0:registrationTimestamp> </wn2:Item> <wn2:Item i:type="wn0:ActionLog"> <wn0:AXCID i:type="String">12</wn0:AXCID> <wn0:AXDID i:type="String">013</wn0:AXDID> <wn0:AXDOM i:type="String">myDomain </wn0:AXDOM> <wn0:AXLID i:type="String">018</wn0:AXLID> <wn0:AXOID i:type="String">97</wn0:AXOID> <wn0:AXTID i:type="String">mytool</wn0:AXTID> <wn0:AXUID i:type="String">87</wn0:AXUID> <wn0:AXWID i:type="String">mywork</wn0:AXWID> <wn0:estimatedHwFingerprint i:type="String">var6</wn0:estimatedHwFingerprint> <wn0:executionTimestamp i:type="String">18:00am</wn0:executionTimestamp> <wn0:instantLastFPPA i:type="String">15:00am</wn0:instantLastFPPA> <wn0:location i:type="String">Spain</wn0:location> <wn0:logID i:type="String">474</wn0:logID> </pre> |

| | |
|-------------------------|--|
| | <pre> <wn0:objectVersion i:type="String">val6</wn0:objectVersion> <wn0:operationDetailsID i:type="String">myoperationdetails</wn0:operationDetailsID> <wn0:operationID i:type="String">010</wn0:operationID> <wn0:ownerName i:type="String">JohnSmith</wn0:ownerName> <wn0:registrationTimestamp i:type="String">09:00am</wn0:registrationTimestamp> </wn2:Item> </wn2:Items> </wn1:myListOfActionLogs> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope> </pre> |
| Input parameters | Sequence of ActionLog that is a complex type previously defined. |
| Output parameters | None |
| Method | getActionLogs |
| Description | This function is used to extract Action Logs (from the AXCS DB Interface) and to send it to the AXMEDIS Accounting or reporting tools. When a Distributor wants to recover information on actions performed on the objects he has rights, AXMEDIS Statistic or reporting tools queries AXCS (or SuperAXCS, who recover information from the different AXCSs). Then AXCS extracts the required Action-Logs from the AXCS DB Interface and communicates them to the tools that perform actions to return results in the desired form (These are AXMEDIS Statistic or reporting tools). |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:axmedis.Supervisor" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:mySID i:type="wn1:SupervisorInputData"> <wn1:AXCID i:nil="true"/> <wn1:AXDID i:type="String">023</wn1:AXDID> <wn1:AXDOM i:type="String">myDomain </wn1:AXDOM> <wn1:AXLID i:nil="true"/> <wn1:AXOID i:nil="true"/> <wn1:AXTID i:nil="true"/> <wn1:AXUID i:nil="true"/> <wn1:AXWID i:nil="true"/> <wn1:estimatedHwFingerprint i:nil="true"/> <wn1:executionTimestamp i:nil="true"/> <wn1:instantLastFPPA i:nil="true"/> <wn1:location i:type="String">Spain</wn1:location> <wn1:logID i:type="String">797</wn1:logID> <wn1:objectVersion i:nil="true"/> <wn1:operationDetailsID i:nil="true"/> <wn1:operationID i:nil="true"/> <wn1:ownerName i:nil="true"/> <wn1:registrationTimestamp i:nil="true"/> <wn1:additionalData i:type="String">getactionlog</wn1:additionalData> </wn0:mySID> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:axmedis.Supervisor" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:ArrayList_Response i:type="wn2:ArrayList"> <wn2:Items> <wn2:Item i:type="wn1:ActionLog"> <wn1:AXCID i:type="String">687</wn1:AXCID> <wn1:AXDID i:type="String">019</wn1:AXDID> <wn1:AXDOM i:type="String">myDomain </wn1:AXDOM> </pre> |

| | |
|------------------------|---|
| | <pre> <wn1:AXLID i:type="String">101</wn1:AXLID> <wn1:AXOID i:type="String">664</wn1:AXOID> <wn1:AXTID i:type="String">mytool</wn1:AXTID> <wn1:AXUID i:type="String">940</wn1:AXUID> <wn1:AXWID i:type="String">mywork</wn1:AXWID> <wn1:estimatedHwFingerprint i:type="String">var8</wn1:estimatedHwFingerprint> <wn1:executionTimestamp i:type="String">17:00am</wn1:executionTimestamp> <wn1:instantLastFPPA i:type="String">17:00am</wn1:instantLastFPPA> <wn1:location i:type="String">Spain</wn1:location> <wn1:logID i:type="String">658</wn1:logID> <wn1:objectVersion i:type="String">val9</wn1:objectVersion> <wn1:operationDetailsID i:type="String">myoperationdetails</wn1:operationDetailsID> <wn1:operationID i:type="String">004</wn1:operationID> <wn1:ownerName i:type="String">JohnSmith</wn1:ownerName> <wn1:registrationTimestamp i:type="String">02:00am</wn1:registrationTimestamp> </wn2:Item> <wn2:Item i:type="wn1:ActionLog"> <wn1:AXCID i:type="String">68</wn1:AXCID> <wn1:AXDID i:type="String">9</wn1:AXDID> <wn1:AXDOM i:type="String">myDomain </wn1:AXDOM> <wn1:AXLID i:type="String">11</wn1:AXLID> <wn1:AXOID i:type="String">64</wn1:AXOID> <wn1:AXTID i:type="String">mytool</wn1:AXTID> <wn1:AXUID i:type="String">94</wn1:AXUID> <wn1:AXWID i:type="String">mywork</wn1:AXWID> <wn1:estimatedHwFingerprint i:type="String">var9</wn1:estimatedHwFingerprint> <wn1:executionTimestamp i:type="String">01:00am</wn1:executionTimestamp> <wn1:instantLastFPPA i:type="String">07:00am</wn1:instantLastFPPA> <wn1:location i:type="String">Spain</wn1:location> <wn1:logID i:type="String">68</wn1:logID> <wn1:objectVersion i:type="String">val8</wn1:objectVersion> <wn1:operationDetailsID i:type="String">myoperationdetails</wn1:operationDetailsID> <wn1:operationID i:type="String">003</wn1:operationID> <wn1:ownerName i:type="String">JohnSmith</wn1:ownerName> <wn1:registrationTimestamp i:type="String">02:00am</wn1:registrationTimestamp> </wn2:Item> </wn2:Items> </wn0:ArrayList_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | ax:SupervisorInputData mySID previously defined. |
| Output parameters | Sequence of ActionLog previously defined. |
| Method | getObjectId |
| Description | This function is used to communicate an Object ID when a Content Provider wants to create a new AXMEDIS Object. When a Content Provider wants to create a new AXMEDIS Object, the tool with that is creating the object requests to the AXCS an object ID. Then AXCS interact with Object ID generator to ask an ID and OID generator returns back to AXCS the generated ID. Once the AXCS has the information and communicates to the tool the generated ID. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:axmedis.Supervisor" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:mySID i:type="wn1:SupervisorInputData"> <wn1:AXCID i:type="String">764</wn1:AXCID> <wn1:AXDID i:nil="true"/> <wn1:AXDOM i:type="String">myDomain </wn1:AXDOM> <wn1:AXLID i:type="String">822</wn1:AXLID> <wn1:AXOID i:type="String">738</wn1:AXOID> <wn1:AXTID i:type="String">mytool</wn1:AXTID> <wn1:AXUID i:type="String">879</wn1:AXUID> <wn1:AXWID i:type="String">mywork</wn1:AXWID> <wn1:estimatedHwFingerprint i:nil="true"/> <wn1:executionTimestamp i:nil="true"/> <wn1:instantLastFPPA i:nil="true"/> </wn0:mySID> </e:Body> </e:Envelope> </pre> |

| | |
|-------------------------|---|
| | <pre> <wn1:location i:type="String">Italy</wn1:location> <wn1:logID i:type="String">68</wn1:logID> <wn1:objectVersion i:nil="true"/> <wn1:operationDetailsID i:type="String">myoperationdetails</wn1:operationDetailsID> <wn1:operationID i:type="String">008</wn1:operationID> <wn1:ownerName i:nil="true"/> <wn1:registrationTimestamp i:nil="true"/> <wn1:additionalData i:type="String">getobjectid</wn1:additionalData> </wn0:mySID> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:axmedis.Supervisor" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:string_Response i:type="String">712</wn0:string_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | ax:SupervisorInputData mySID previously defined. |
| Output parameters | type="xsd:string" id |
| Method | storeHistoryIsNotConsistent |
| Description | This function is used to store that history is not consistent and to block user and/or tool and/or device due to an unsuccessful verification of an AXMEDIS user. When an end User requests to perform an action on an AXMEDIS Object and the verification of the user/tool/device is unsuccessful, the information about unsuccessful verification is sent to Supervisor. The Supervisor stores that history is not consistent and controls each unsuccessful verification and then blocks user and/or tool and/or device. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:axmedis.Supervisor" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:mySID i:type="wn1:SupervisorInputData"> <wn1:AXCID i:type="String">12</wn1:AXCID> <wn1:AXDID i:type="String">02</wn1:AXDID> <wn1:AXDOM i:type="String">myDomain</wn1:AXDOM> <wn1:AXLID i:type="String">09</wn1:AXLID> <wn1:AXOID i:type="String">14</wn1:AXOID> <wn1:AXTID i:type="String">mytool</wn1:AXTID> <wn1:AXUID i:type="String">85</wn1:AXUID> <wn1:AXWID i:type="String">mywork</wn1:AXWID> <wn1:estimatedHwFingerprint i:type="String">var6</wn1:estimatedHwFingerprint> <wn1:executionTimestamp i:type="String">15:00am</wn1:executionTimestamp> <wn1:instantLastFPPA i:type="String">05:00am</wn1:instantLastFPPA> <wn1:location i:type="String">Spain</wn1:location> <wn1:logID i:type="String">97</wn1:logID> <wn1:objectVersion i:type="String">val3</wn1:objectVersion> <wn1:operationDetailsID i:type="String">myoperationdetails</wn1:operationDetailsID> <wn1:operationID i:type="String">001</wn1:operationID> <wn1:ownerName i:type="String">JohnSmith</wn1:ownerName> <wn1:registrationTimestamp i:type="String">00:00am</wn1:registrationTimestamp> <wn1:additionalData i:type="String">storynotconsistent</wn1:additionalData> </wn0:mySID> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope> </pre> |
| Input parameters | ax:SupervisorInputData mySID previously defined. |

| | |
|-------------------------|--|
| Output parameters | None |
| Method | getLastFingerprint |
| Description | This function is used to ask to Supervisor for the Last Fingerprint of an user or an object or a tool in order to certify or verify any user. |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:axmedis.Supervisor" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:mySID i:type="wn1:SupervisorInputData"> <wn1:AXCID i:type="String">168</wn1:AXCID> <wn1:AXDID i:type="String">001</wn1:AXDID> <wn1:AXDOM i:type="String">myDomain</wn1:AXDOM> <wn1:AXLID i:type="String">071</wn1:AXLID> <wn1:AXOID i:type="String">141</wn1:AXOID> <wn1:AXTID i:type="String">mytool</wn1:AXTID> <wn1:AXUID i:type="String">855</wn1:AXUID> <wn1:AXWID i:type="String">mywork</wn1:AXWID> <wn1:estimatedHwFingerprint i:type="String">var7</wn1:estimatedHwFingerprint> <wn1:executionTimestamp i:type="String">17:00am</wn1:executionTimestamp> <wn1:instantLastFPPA i:type="String">05:00am</wn1:instantLastFPPA> <wn1:location i:type="String">Spain</wn1:location> <wn1:logID i:type="String">95</wn1:logID> <wn1:objectVersion i:type="String">val7</wn1:objectVersion> <wn1:operationDetailsID i:type="String">myoperationdetails</wn1:operationDetailsID> <wn1:operationID i:type="String">007</wn1:operationID> <wn1:ownerName i:type="String">JohnSmith</wn1:ownerName> <wn1:registrationTimestamp i:type="String">20:00am</wn1:registrationTimestamp> <wn1:additionalData i:type="String">getlastfingerprint</wn1:additionalData> </wn0:mySID> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="urn:axmedis.Supervisor" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:string_Response i:type="String">lastfingerprint</wn0:string_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | ax:SupervisorInputData mySID previously defined. |
| Output parameters | type="xsd:string" lastFingerprint |
| Method | getProtectionInfo |
| Description | This function is used to extract the protection information of an AXCS Objects ID Database object. |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:AxmedisSupervisor.Supervisor" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myAXOID i:type="d:string">myAXOID</wn1:myAXOID> <wn1:myObjectVersion i:type="d:string">myObjectVersion</wn1:myObjectVersion> <wn1:myProtectionStamp i:type="d:string">myProtectionStamp</wn1:myProtectionStamp> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:AxmedisSupervisor.Supervisor" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body></pre> |

| | |
|-------------------------|--|
| | <pre> <wn1:string_Response i:nil="true"/> </e:Body> </e:Envelope> </pre> |
| Input parameters | The following fields of the Objects table in the AXCS Objects ID database: type="xsd:string" AXOID, part of primary key type="xsd:string" myObjectVersion, part of primary key type="xsd:string" myProtectionStamp, part of primary key |
| Output parameters | type="xsd:string" protectionInfo, protection information associated to the object |
| Method | updateProtectionInfo |
| Description | This function is used to update the protection information of an AXCS Objects ID Database object. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:AxmedisSupervisor.Supervisor" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myAXOID i:type="d:string">myAXOID</wn1:myAXOID> <wn1:myObjectVersion i:type="d:string">myObjectVersion</wn1:myObjectVersion> <wn1:myProtectionStamp i:type="d:string">myProtectionStamp</wn1:myProtectionStamp> <wn1:myProtectionInfo i:type="d:string">protectionInfo</wn1:myProtectionInfo> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope> </pre> |
| Input parameters | The following fields of the Objects table in the AXCS Objects ID database: type="xsd:string" AXOID, part of primary key type="xsd:string" myObjectVersion, part of primary key type="xsd:string" myProtectionStamp, part of primary key type="xsd:string" myProtectionInfo, information to be updated |
| Output parameters | None |

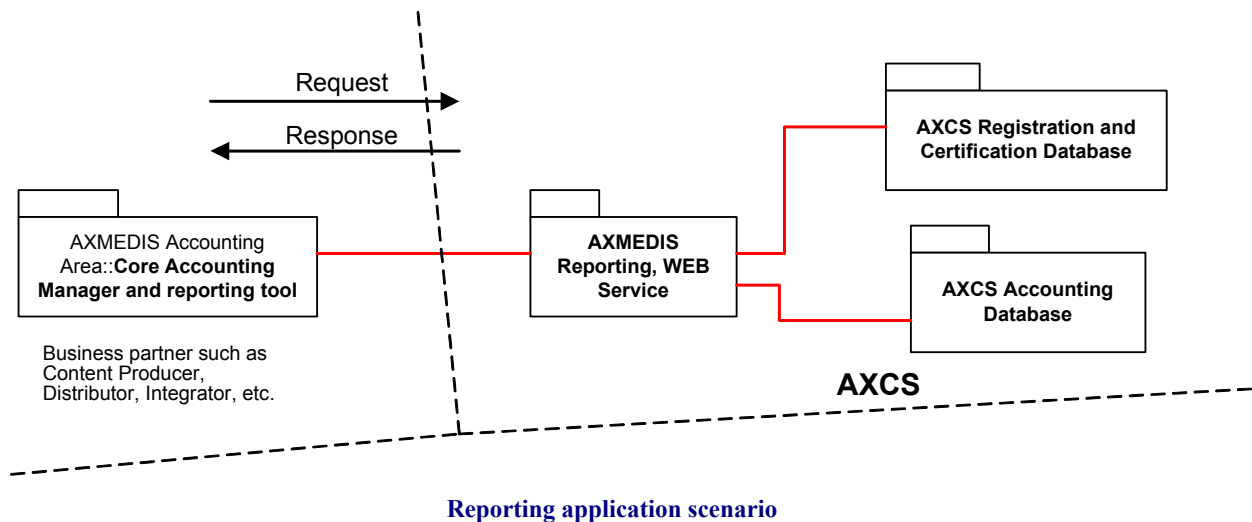
2.5 AXMEDIS Reporting (DSI, EXITECH)

| Module Profile | | |
|--------------------------------|---------------------------------|---|
| AXMEDIS Reporting | | |
| Executable or Library(Support) | Web application | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Accounting Area:CAMART | Web service based on (TBD) | Protected (SSL) |
| AXCS: Accounting Database | | |
| AXCS: Registration and | | |

| | | |
|------------------------|-----------------------------------|--|
| Certification Database | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No user interface | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

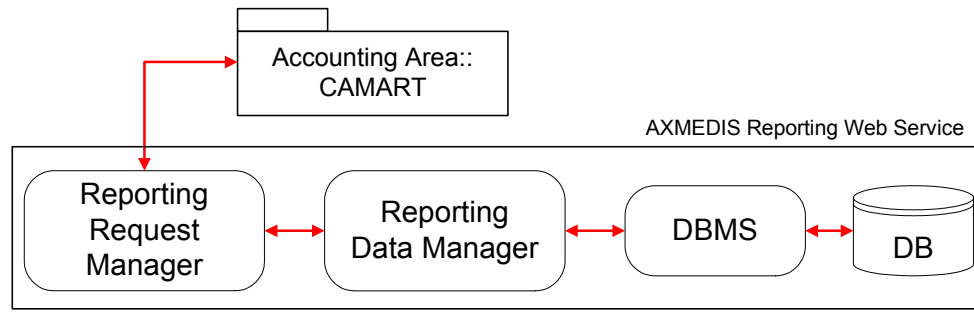
The object usage accounting activity is a fundamental asset in the AXMEDIS system. Distributors, Creators, Integrators, Collecting Societies have to know all the operations performed over their pertinent objects to receive the correctness fee from the object users. This is a vital activity if we want to guarantee the owner rights are respected. All the accounting needful information are stored in AXCS Accounting Database. The AXMEDIS Reporting unit is the high-level interface to this database that provides a sequence of services needful to retrieve information in a correct way, with no errors and with respect to the users privacy and the companies/societies confidential data. The AXMEDIS Reporting Web Services deals with CAMART (Core accounting Manager and Reporting Tool). CAMART is a sort of client application, used by requestors, that queries AXMEDIS Reporting Web Services to retrieve all the needful information. The CAMART can be consider the client part of the reporting system and the AXMEDIS Reporting Web Service can be consider the server part. Please see the pertinent section to understand relationship between CAMART and AXMEDIS Reporting Web Services.

2.5.1 AXMEDIS Reporting Logic



The AXMEDIS Reporting Web Service is a web application running on a web server implemented as a set of scripts. We can identify the following logical decomposition:

- **Reporting Request Manager:** this component receives reporting requests from CAMART, and prepares them to be processed by the Reporting Data Manager. The communication channel connecting CAMART and Reporting Request Manager is protected using a secure protocol (for instance SSL). It implements the interface with requestors and manage the whole application.
- **Reporting Data Manager:** this component receives data requests from Reporting Request Manager and query the AXCS Accounting Database on the basis of queried data. It also elaborates retrieved data before transferring it to Reporting Request Manager. The database management is performed using the related API (AXDB-API).



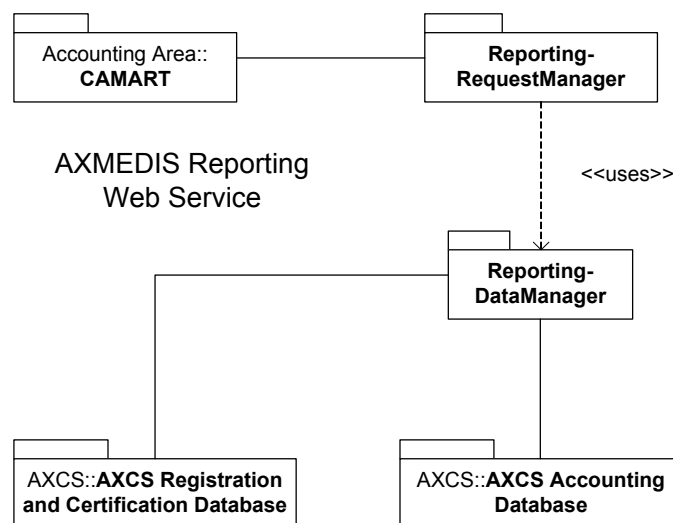
AXMEDIS Reporting Web Service architecture.

The **Reporting Request Manager** should be composed by some modules implementing the following functions:

- Accepting authentication data, needful to verify CAMART credentials and make it access the system
- Accepting reporting requests and preparing them to be transferred to Reporting Data Manager
- Reply to requestor with the most appropriate data or error messages (on the basis of Reporting Data Manager responses)

The **Reporting Data Manager** should be composed by some modules implementing the following functions:

- Receiving requests and data from the Reporting Request Manager
- Accessing the database (using the AXDB-API) to verify CAMART Users credentials information
- Accessing the database to retrieve data on the basis of CAMART Users requests (received from Reporting Request Manager) and CAMART Users profile (note that requestors can access only pertinent data)
- Reply to Reporting Request Manager according to the performed actions and obtained results



The following table describes methods thought to be used in AXMEDIS Reporting Web Service.

| Method name and parameters | Output | Description |
|---|---|---|
| scope: public | | |
| Reporting-RequestManager::AcceptRequest (string NickName, string Password, String CAMARTQuery) | String ResultStatus logDataType Response | It is the only public method of this web service. It collects CAMART credentials needful to access the system and uses the VerifyLogin() (a DataManager method) to verify requestor credentials. It also collects the CAMART query (in a field called CAMARTQuery). CAMARTQuery contains a timestamp after which log have to be returned. This method uses private methods described below to perform its tasks and to answer to requestor. The ResultStatus output parameter is set to 0 if the query is successful and is set to 1 otherwise. This method returns also the query result to the requestor using the Response field (typed logDataType). |
| scope: private | | |
| Reporting-RequestManager::RequestElaborate (String CAMARTQuery) | String Response | It elaborates CAMART requests. The CAMARTQuery field is managed to extract the needful data to perform the query on database. The query is performed by QueryExecuter() a Reporting-DataManager method. It also elaborates query results to be sent to the requesting CAMART |
| Reporting-RequestManager::EncryptComm (string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| Reporting-RequestManager::DecryptComm (string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| Reporting-DataManager::QueryExecuter() (the needful input data can be stored in the class attributes or sent as method parameters) | String QueryResult | It performs the query on the basis of data elaborated by AcceptRequest a method of Reporting-RequestManager. |
| Reporting-DataManager::VerifyLogin (string NickName, string Password) | boolean IsTrusted string PublicKey | It accesses AXCS Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process. It also retrieves the CAMART manager public key stored in the DB. Note: the password is encrypted both the one received as input from the method and the one stored in DB. The equality check is made between encrypted strings. |

Note that it has been introduced two encrypting/decrypting methods (DecryptComm, EncryptComm) to enforce the encrypting robustness. In fact we can suppose to use an encrypted protocol (like SSL), but we can enforce encryption robustness (and therefore security) encrypting our self data too using a Public/Private key paradigm. You have to remember that public keys are stored in AXCS Registration and Certification Database.

2.5.2 AXMEDIS Reporting Web Service interface formalization

In the present paragraph is explained the Reporting Web Service interface using the WSDL formalism.

| | |
|------|---|
| WSDL | <pre> <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:rpt="http://new.webservice.namespace" targetNamespace="http://new.webservice.namespace"> <types> <xs:schema targetNamespace="http://new.webservice.namespace" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="logDataType"> <sequence> <!-- event log (it's only a subset of possible fields compliant with the database)--> <element name="LogID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXOID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="ObjectVersion" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXWID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXUID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXCID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="OwnerName" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXTID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXLID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Location" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="OperationDetailsID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="OperationID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegistrationTimestamp" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="ExecutionTimestamp" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="InstantLastFPPA" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="EstimatedHwFingerprint" type="xsd:string" minOccurs="0" maxOccurs="1"/> </sequence> </complexType> </xs:schema> </types> <message name="ReportingRequest"> <part name="NickName" type="xs:string"/> <part name="Password" type="xs:string"/> <part name="CAMARTQuery" type="xs:string"/> </message> <message name="ReportingResponse"> <part name="ResultStatus" type="xs:string"/> <part name="Response" type="rpt:logDataType"/> </message> <portType name="Reporting_PortType"> <operation name="Reporting"> <input message="rpt:ReportingRequest"/> <output message="rpt:ReportingResponse"/> </operation> </portType> <binding name="Reporting" type="rpt:Reporting_PortType"> <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="Reporting"> <soap:operation soapAction="urn:#Reporting"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <soap:operation soapAction="urn:#Reporting"/> </operation> </binding> <service name="ReportingWebService"> <port name="Reporting" binding="rpt:Reporting"> </pre> |
|------|---|

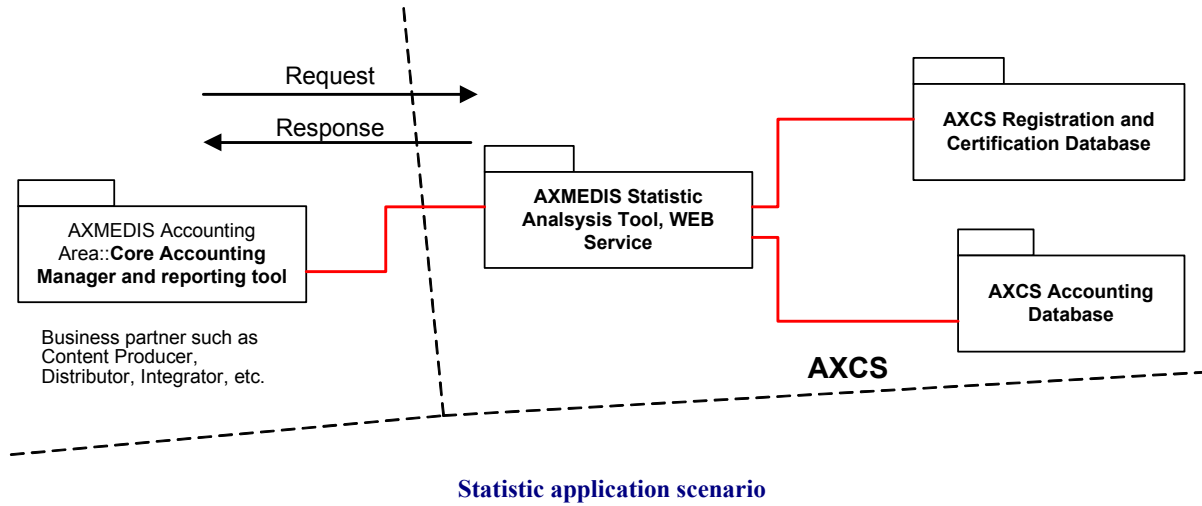
| | |
|-------------------------------|--|
| | <pre> <soap:address location="No Target Adress"/> </port> </service> </definitions> </pre> |
| Request Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Reporting xmlns:m="http://new.webservice.namespace"> <NickName>...CAMART user...</NickName> <Password>...Password...</Password> <CAMARTQuery>2005-02-12T03:05:45</CAMARTQuery> </m:Reporting> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
| Response Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Reporting xmlns:m="http://new.webservice.namespace"> <ResponseStatus>0</ResponseStatus> <Response> <LogID>035487200256541684984651</LogID> <AXOID>0A2Z4X678B0124456X89W12345B7P901</AXOID> <ObjectVersion>01.10.153</ObjectVersion> <AXWID>1D7E2G6785F7W2B56X8DC143451V5FF6</AXWID> <AXDOM>1A001C578B02WS6X89W1234555A90M24</AXDOM> <AXUID>8XW123455C690M241A001C478F02WS6X</AXUID> <AXDID>6X89W12345B7P9011A001C578B02WS6X</AXDID> <AXCID>2005-02-12T03:05:45</AXCID> <OwnerName>Fabius Monroe LTD</OwnerName> <AXTID>2005-02-12T03:05:45</AXTID> <AXLID>2005-02-12T03:05:45</AXLID> <Location>Germany</Location> <OperationDetailsID>012454408840365460540</OperationDetailsID> <OperationID>21651651051684068464</OperationID> <RegistrationTimestamp>2005-02-14T15:07:20</RegistrationTimestamp> <ExecutionTimestamp>2005-02-14T15:07:16</ExecutionTimestamp> <InstantLastFPPA>... instant last fingerprint of performed actions ... </InstantLastFPPA> <EstimatedHwFingerprint>... Estimated Hardware Fingerprint ...</EstimatedHwFingerprint> </Response> </m:Reporting> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |

2.6 AXMEDIS Statistic Analysis Tool (DSI, EXITECH)

| Module Profile | | |
|---|-----------------------------------|--|
| AXMEDIS Statistic Analysis Tool | | |
| Executable or Library(Support) | Web Application | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Accounting Area:CAMART | Web service based on (TBD) | Protected (SSL) |
| AXCS: Accounting Database | | |
| AXCS: Registration and Certification Database | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No user interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

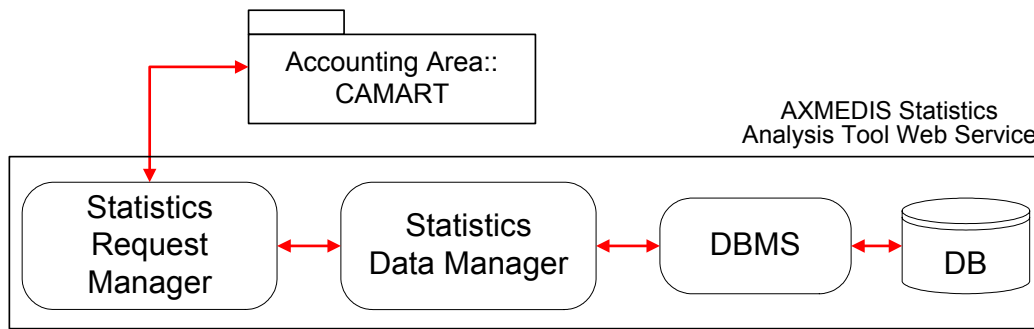
In the AXMEDIS environment, statistics can represent an important asset for the whole system. A Distributor, Creator, Integrator could be interested in knowing where, when and how an object is used: this interest could be used for commercial and marketing purpose. Knowing usage, distribution and integration statistics could be an important resource for an AXMEDIS subject to improve his business. Therefore has been introduced a statistical tool called AXMEDIS Statistics Analysis Tool Web Service which task is retrieving anonymous statistical information from the AXCS Accounting Database. This tool can be query by all AXMEDIS subjects and produce anonymous statistics concerning objects. The AXMEDIS Statistics Analysis Tool deals with CAMART (Core accounting Manager and Reporting Tool). CAMART is a sort of client application, used by requestors, that queries AXMEDIS Statistics Analysis Tool to retrieve all the needful information. The CAMART can be consider the client part of the Statistic Analysis system and the AXMEDIS Statistics Analysis Tool can be consider the server part. Please see the pertinent section to understand relationship between CAMART and AXMEDIS Statistics Analysis.

2.6.1 AXMEDIS Statistics Analysis Logic



The AXMEDIS Statistics Analysis Tool Web Service is a web application running on a web server implemented as a set of scripts. We can identify the following logical decomposition:

- **Statistics Request Manager:** this component receives requests from CAMART, and prepares them to be processed by the Statistics Data Manager. The communication channel connecting CAMART and Statistics Request Manager is protected using a secure protocol (for instance SSL). It implements the interface with requestors and manage the whole application.
- **Statistics Data Manager:** this component receives data requests from Statistics Request Manager and query the AXCS Accounting Database on the basis of queried data. It also elaborates retrieved data before transferring it to Statistics Request Manager. The database management is performed using the related API (AXDB-API).



AXMEDIS Statistics Analysis Tool Web Service architecture.

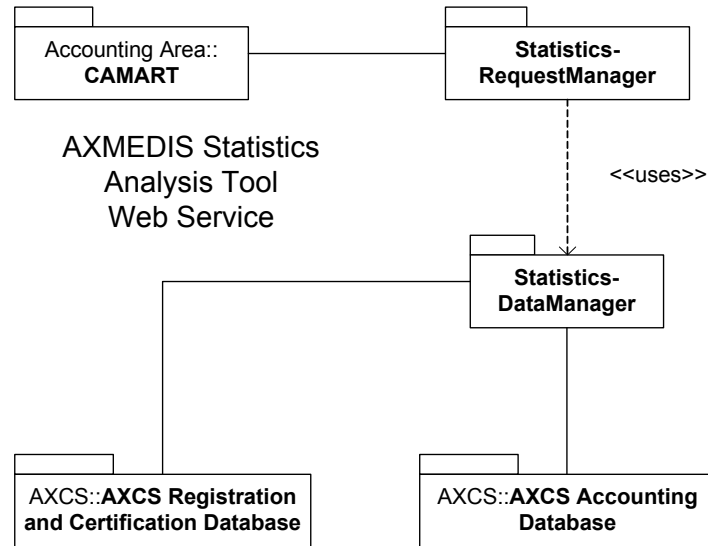
The **Statistics Request Manager** should be composed by some modules implementing the following functions:

- Accepting authentication data, needful to verify CAMART credentials and make it access the system
- Accepting requests and preparing them to be transferred to Statistics Data Manager
- Reply to requestor with the most appropriate data or error messages (on the basis of Statistics Data Manager responses)

The **Statistics Data Manager** should be composed by some modules implementing the following functions:

- Statistics requests and data from the Statistics Request Manager
- Accessing the database (using the AXDB-API) to verify CAMART Users credentials information

- Accessing the database to retrieve statistics data on the basis of CAMART Users requests (received from Statistics Request Manager)
- Reply to Statistics Request Manager according to the performed actions and obtained results



The following table describes methods thought to be used in AXMEDIS Statistics Analysis Tool Web Service.

| Method name and parameters | Output | Description |
|---|---|--|
| scope: public | | |
| Statistics-RequestManager::AcceptRequest (string NickName, string Password, String CAMARTQuery) | String ResultStatus logDataType Response | It is the only public method of this web service. It collects CAMART credentials needful to access the system and uses the VerifyLogin() (a DataManager method) to verify requestor credentials. It also collects the CAMART query (in a field called CAMARTQuery). CAMARTQuery contains a timestamp after which log have to be returned. This method uses private methods described below to perform its tasks and to answer to requestor. The ResultStatus output parameter is set to 0 if the query is successful and is set to 1 otherwise. This method returns also the query result to the requestor using the Response field (typed logDataType). |
| scope: private | | |
| Statistics-RequestManager::RequestElaborate (String CAMARTQuery) | String Response | It elaborates CAMART requests. The CAMARTQuery field is managed to extract the needful data to perform the query on database. The query is performed |

| | | |
|---|---------------------------------------|---|
| | | by QueryExecuter() a Statistics-DataManager method. It also elaborates query results to be sent to the requesting CAMART |
| Statistics-RequestManager::EncryptComm (string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| Statistics-RequestManager::DecryptComm (string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| Statistics-DataManager::QueryExecuter() (the needful input data can be stored in the class attributes or sent as method parameters) | String QueryResult | It performs the query on the basis of data elaborated by AcceptRequest a method of Statistics-RequestManager. |
| Statistics-DataManager::VerifyLogin (string NickName, string Password) | boolean IsTrusted string PublicKey | It accesses AXCS Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process. It also retrieves the CAMART manager public key stored in the DB. Note: the password is encrypted both the one received as input from the method and the one stored in DB. The equality check is made between encrypted strings. |

Note that it has been introduced two encrypting/decrypting methods (DecryptComm, EncryptComm) to enforce the encrypting robustness. In fact we can suppose to use an encrypted protocol (like SSL), but we can enforce encryption robustness (and therefore security) encrypting our self data too using a Public/Private key paradigm. You have to remember that public keys are stored in AXCS Registration and Certification Database.

2.6.2 AXMEDIS Statistics Analysis WEB Service interface formalisation

In the present paragraph is explained the Statistics Web Service interface using the WSDL formalism.

| | |
|------|--|
| WSDL | <pre> <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:stc="http://new.webservice.namespace" targetNamespace="http://new.webservice.namespace"> <types> <xs:schema targetNamespace="http://new.webservice.namespace" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="stat_logDataType"> <sequence> <!-- event log (it's only a subset of possible fields compliant with the database)--> <element name="LogID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXOID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="ObjectVersion" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXWID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDOM" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="AXDID" type="xsd:string" minOccurs="0" maxOccurs="1"/> </sequence> </complexType> </xs:schema> </types> </pre> |
|------|--|

| | |
|-------------------------------|--|
| | <pre> <element name="AXCID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="OwnerName" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="Location" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="OperationDetailsID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="OperationID" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="RegistrationTimestamp" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="ExecutionTimestamp" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="InstantLastFPPA" type="xsd:string" minOccurs="0" maxOccurs="1"/> <element name="EstimatedHwFingerprint" type="xsd:string" minOccurs="0" maxOccurs="1"/> </sequence> </complexType> </xs:schema> </types> <message name="StatisticsRequest"> <part name="NickName" type="xs:string"/> <part name="Password" type="xs:string"/> <part name="CAMARTQuery" type="xs:string"/> </message> <message name="StatisticsResponse"> <part name="ResultStatus" type="xs:string"/> <part name="Response" type="stc:stat_logDataType"/> </message> <portType name="Statistics_PortType"> <operation name="Statistics"> <input message="stc:StatisticsRequest"/> <output message="stc:StatisticsResponse"/> </operation> </portType> <binding name="Statistics" type="stc:Statistics_PortType"> <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="Statistics"> <soap:operation soapAction="urn:#Statistics"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <soap:operation soapAction="urn:#Statistics"/> </operation> </binding> <service name="StatisticsWebService"> <port name="Statistics" binding="stc:Statistics"> <soap:address location="No Target Adress"/> </port> </service> </definitions> </pre> |
| Request Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Statistics xmlns:m="http://new.webservice.namespace"> <NickName>... CAMART user ...</NickName> <Password>...password...</Password> <CAMARTQuery>String</CAMARTQuery> </m:Statistics> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
| Response Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Statistics xmlns:m="http://new.webservice.namespace"> <ResultStatus>0</ResultStatus> <Response> <LogID>035487200256541684984651</LogID> <AXOID>0A2Z4X678B0124456X89W12345B7P901</AXOID> <ObjectVersion>01.10.153</ObjectVersion> <AXWID>1D7E2G6785F7W2B56X8DC143451V5FF6</AXWID> <AXDOM>1A001C578B02WS6X89W1234555A90M24</AXDOM> </Response> </m:Statistics> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |

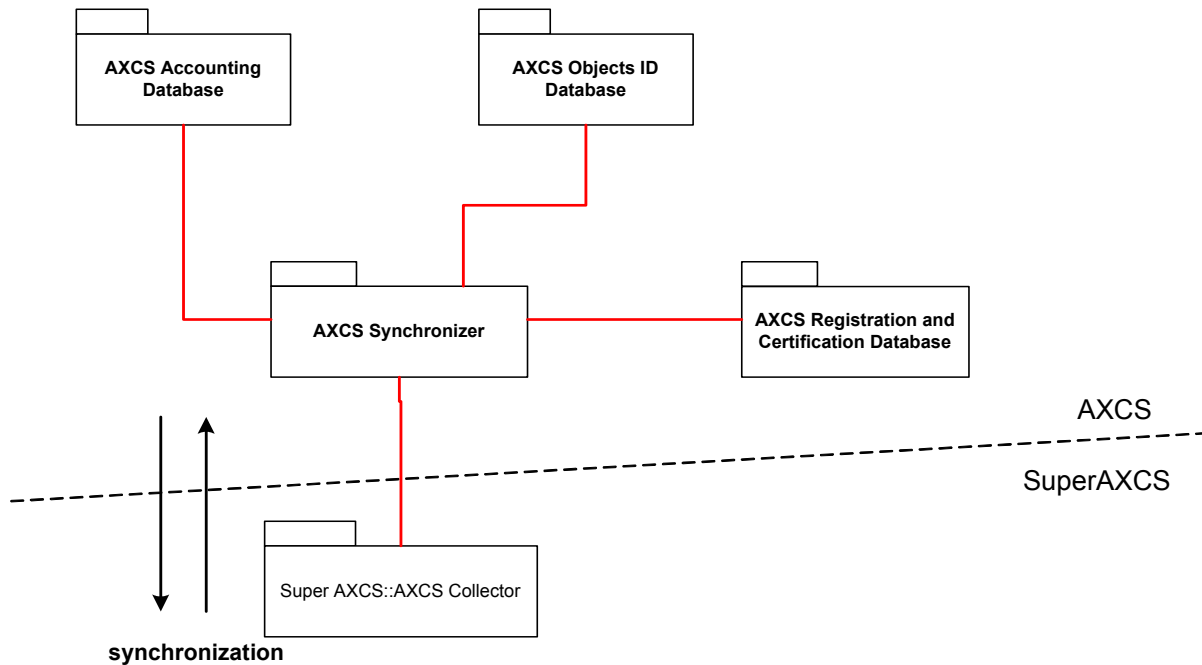
| | |
|--|--|
| | <pre> <AXDID>6X89W12345B7P9011A001C578B02WS6X</AXDID> <AXCID>2005-02-12T03:05:45</AXCID> <OwnerName>Fabius Monroe LTD</OwnerName> <Location>Germany</Location> <OperationDetailsID>012454408840365460540</OperationDetailsID> <OperationID>21651651051684068464</OperationID> <RegistrationTimestamp>2005-02-14T15:07:20</RegistrationTimestamp> <ExecutionTimestamp>2005-02-14T15:07:16</ExecutionTimestamp> <InstantLastFPPA>... instant last fingerprint of performed actions ... </InstantLastFPPA> <EstimatedHwFingerprint>... Estimated Hardware Fingerprint ...</EstimatedHwFingerprint> </Response> </m:Statistics> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
|--|--|

2.7 AXCS Synchronizer (DSI, EXITECH)

| Module Profile | | |
|--|-----------------------------------|--|
| AXCS Synchronizer | | |
| Executable or Library(Support) | Web service or Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| SuperAXCS:Collector | Web service based on (TBD) | Protected (SSL) |
| AXCS:Accounting Database | | |
| AXCS:Registration and Certification Database | | |
| AXCS:Object ID Database | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No user interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

AXCS Synchronizer is an AXCS module used to synchronize AXCSs and SuperAXCS database. Every AXCS Database entry has to be retrievable by SuperAXCS in order to be able to provide to requestors all the information contained in each AXCS database.

2.7.1 AXCS Synchronizer Logic



The AXCS Synchronizer is the counterpart of SuperAXCS Collector. The couple AXCS Synchronizer and SuperAXCS Collector has been defined in order to supply an efficient way to transfer AXCS database content (or a part of it) to SuperAXCS. SuperAXCS has to be able to provide to authorized requestors the AXCS database data they request without take in account which is the AXCS that holds it. To perform their own tasks, several architectures are possible:

1. Hierarchical Network. In this architecture SuperAXCS is located at an higher level than AXCSs. Here has to be present a client part and a server part. SuperAXCS Collector could be the client part that periodically requests to AXCSs Synchronizer all the data needed and store it in its database. This is the simplest way but probably it isn't the most efficient. A simple variant could be the "on-line" synchronization where AXCSs Synchronizer tries to transfer data at the same time they perform their tasks. Here are reported two simple cases to better explain these concepts.

| On-line synchronization | |
|-------------------------|--|
| 1 | End user uses an AXMEDIS tool to operate on AXMEDIS Protected Objects that are on different distribution channels |
| 2 | Protection Manager Support allows only authorized operations on the objects |
| 3 | Objects are accessed on different channels and each AXCS stores its Action-Logs |
| 4 | Via AXCS synchronizer general information on Objects or information that allow Super AXCS to recover Action-Logs from the different AXCSs are transferred to Super AXCS Collector |
| 4a | If connection between AXCS and Super AXCS is not active, AXCS synchronizer stores the information to be transferred in a queue called AXCS Synchronizer Queue. Information stored in that queue are transferred to SuperAXCS Collector when the connection returns active. |
| 5 | Super AXCS collects and sores the received information |

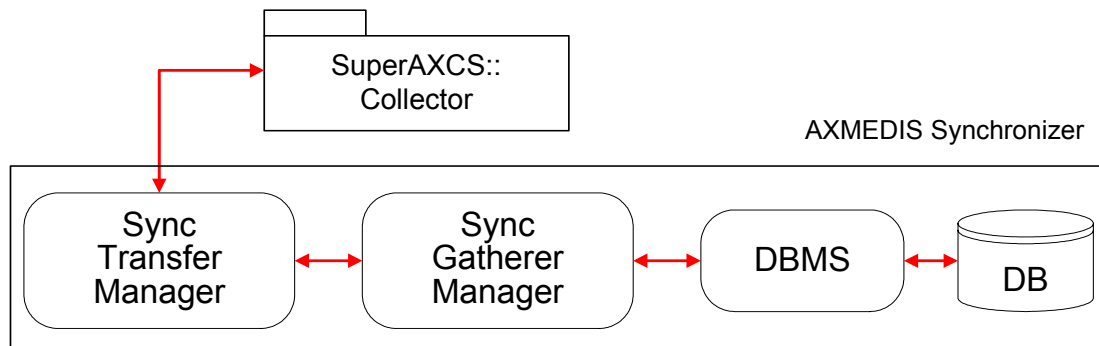
| Off-line synchronization | |
|--------------------------|--|
| 1 | Super AXCS Collector retrieve the list of all the AXCS registered in the system (performing a query to |

| | |
|---|--|
| | Active AXCS List Database) |
| 2 | Super AXCS Collector slides every entry in that list and, for each AXCS, sends a Queue Pull Request, i.e. a request for data present in the AXCS Synchronizer Queue. |
| 3 | The contacted AXCS Synchronizer respond providing the requested data (if present) and empty the AXCS Synchronizer Queue. |

2. **Peer Network.** In this architecture SuperAXCS is only a coordinator element. It stores only the list of registered AXCSs and other few information. Every AXCS manage its own data. When information stored in a specific AXCS is needed, SuperAXCS sent it to some AXCS that probably retains the needed data (chosen with some criteria). Every AXCS is a peer and queries are propagated over the network in order to retrieve the needed information. Every AXCS stores its own data and some other data coming from the adjacent AXCSs to be able to propagate the query in a tricky way. In this architecture the couple Synchronizer/Collector is not necessary.
3. **Blend Network.** In this architecture SuperAXCS is more than a coordinator element. It stores only data needful to retrieve information about which is the correct AXCS to query (the AXCS that really stores the needed data). In this architecture data transferred by Synchronizer to Collector is not so excessive. The synchronization methods could be the same as the Hierarchical Network.

This different network architectures could be implemented and tested in a second time, in order to test them and be able to make the better choice. Whatever will be the chosen architecture the synchronizer should be composed by some modules performing the following tasks:

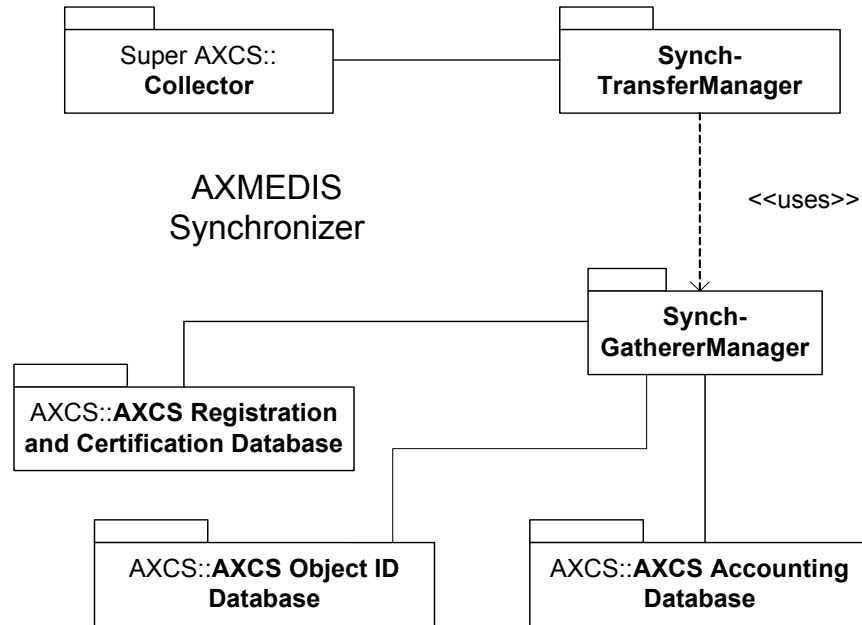
- gather the AXCS database data that has to be transferred to SuperAXCS Collector
- transfer it to SuperAXCS Collector



To perform these tasks can be considered two different modules:

SyncGathererManager: it gathers all data that has to be transfer to Collector and elaborates it before is treated by SyncTransferManager. The database management is performed using the related API (AXDB-API). To support synchronization a new structure has been introduced: the AXCS Synchronizer Queue. When the connection betrwwen Synchronizer (in AXCS) and Collector (in SuperAXCS) is not active, AXCS synchronizer stores the information to be transferred in the AXCS Synchronizer Queue. Information stored in that queue are transferred to SuperAXCS Collector when the connection returns active.

SyncTransferManager: it manages the connection and deals with SuperAXCS Collector in order to transfer the pertinent data. The communication channel connecting SuperAXCS Collector and AXCS Synchronizer is protected using a secure protocol (for instance SSL). The communication can happen with Synchronizer as client or server (and the Collector as a respective counterpart). To perform both paradigms have been considered a public method both in Synchronizer and in Collector.



The following table describes methods thought to be used in AXCS Synchronizer.

| Method name and parameters | Output | Description |
|---|--|--|
| scope: public | | |
| SyncTransferManager::Synchronize(string NickName, string Password, string TimeStampQuery) | String ResultStatus syncDataType Response | It is the only public method of this module. It collects Super AXCS credentials needful to access the system and uses the VerifyLogin() (a GathererManager method) to verify requestor credentials. This method uses private methods described below to perform its own tasks and to answer to requestor. The TimeStampQuery contains the time (in timestamp form) after which the collector needs to be updated; this field is optional and depends on the type of architecture/synchronization mode will be chosen. The ResultStatus output parameter is set to 0 if the task is successful and is set to 1 otherwise. This method returns also the data to be transferred to the requestor using the Response field (typed syncDataType). |
| scope: private | | |
| SyncTransferManager::EncryptComm (string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| SyncTransferManager::DecryptComm (string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| SyncGathererManager::DataGatherer() (the needful input data can be stored in the class attributes or received as method | String QueryResult | It gathers the needed data from database. In order to support its own task it uses the AXCS |

| | | |
|--|-------------------|--|
| parameters) | | Synchronizer Queue (see above) .Query results are stored in the QueryResult field |
| SyncGathererManager::VerifyLogin (string NickName, string Password) | boolean IsTrusted | It accesses AXCS Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process Note: the password is encrypted both the one received as input from the method and the one stored in DB. The equality check is made between encrypted strings. |

Note that it has been introduced two encrypting/decrypting methods (DecryptComm, EncryptComm) to enforce the encrypting robustness. In fact we can suppose to use an encrypted protocol (like SSL), but we can enforce encryption robustness (and therefore security) encrypting our self data too using a Public/Private key paradigm. You have to remember that public keys are stored in AXCS Registration and Certification Database.

For more information about synchronization please see the SuperAXCS Collector pertinent section (the counterpart of AXCS Synchronizer).

2.8 AXCS Database Interface (DSI, EXITECH)

| Module Profile | | |
|--|-----------------------------------|--|
| AXCS Database Interface | | |
| Executable or Library(Support) | Library | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXCS:Registration and Certification Database | | Not Protected |
| AXCS:Objects ID Database | | |
| AXCS:Accounting Database | | |
| AXCS:Certification and Verification | | |
| AXCS:AXMEDIS Supervisor | | |
| AXMEDIS OID Generator | | |
| AXCS:Manager User Interface | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No user interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

2.8.1 AXCS Database Interface Logic

This module task is to give to the rest of AXCS modules a view of the database that is abstracted among the database that is really employed. This module can be decomposed in an abstraction layer with plugin that can also have different implementations on the basis of the database adopted. The logic and implementation is similar to the AXMEDIS Database Interface discussed in the pertinent section. Therefore, in this section, will be discussed only public interfaces to database (AXDB-API) to be used by other AXCS modules such as AXMEDIS Supervisor. It can be identified the following modules performing their pertinent tasks:

- Registration-DBManager: It provides a way to access AXCS Registration and Certification Database.
- ObjectsID-DBManager: It provides a way to access AXCS Objects ID Database.
- Accounting-DBManager: It provides a way to access AXCS Accounting Database.

Here are reported the main methods of each module. Some other methods can be added so as making available more functionality to the rest of the system.

| Registration-DBManager | |
|-------------------------------|---|
| Method | dmn Insert |
| Description | It inserts domain fields value in Domains table |
| Input parameters | domainDataType DomainData - set of related domain data |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| Method | dmn Delete |
| Description | It deletes domain fields value (the whole table row) in Domains table |
| Input parameters | String AXDOM – ID of domain |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| Method | dmn Update |
| Description | It updates domain fields value in Domains table |
| Input parameters | String AXDOM – ID of domain |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| Method | dmn Retrieve |
| Description | It recovers domain fields value in Domains table |
| Input parameters | String Query – SQL query used to retrieve data from the database |
| Output parameters | domainDataType DomainData - set of related domain data |
| Method | SwFingerprint Retrieve |
| Description | It recovers software fingerprint for a given AXRTID, accessing RegTools table |
| Input parameters | String AXRTID - ID of registered tool |
| Output parameters | String SwFingerprint – fingerprint of a registered tool |
| Method | certTool Insert |
| Description | It allows the insertion of a new certified tool in the CertTools table. Tool related data has to be provided. |
| Input parameters | certToolDataType certToolData – set of related certified tool data |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| Method | certTool Retrieve |
| Description | It allows recovering of certified tool located in the CertTools table. |
| Input parameters | String AXTID – ID of certified tool |
| Output parameters | certToolDataType certToolData – set of related certified tool data |
| Method | LastFPPA Update |
| Description | It allows updating of LastFPPA field in CertTools table. |
| Input parameters | String AXTID - ID of certified tool String LastFPPA – last fingerprint of performed actions related to the certified tool with ID equal to AXTID |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |

| Method | Status_Update |
|-------------------|--|
| Description | It allows updating of Status field in CertTools table. |
| Input parameters | String AXTID - ID of certified tool String Status – new status of certified tool with ID equal to AXTID |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| | |
| Method | Query_exec |
| Description | It allows performing a generic query over database. |
| Input parameters | String Query – Generic query to be performed |
| Output parameters | String QueryResult – Data retrieved by the performed query String Result – status result of operation: 0 for successful 1 otherwise |
| | |

| ObjectsID-DBManager | |
|---------------------|---|
| Method | obj_Insert |
| Description | It inserts object field values in Object ID database |
| Input parameters | objectDataType ObjectData - set of related object data |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| | |
| Method | obj_Delete |
| Description | It deletes object fields value in Object ID database |
| Input parameters | AXOID - ID of related object |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| | |
| Method | obj_Update |
| Description | It updates object fields value in Object ID database |
| Input parameters | AXOID - ID of related object |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| | |
| Method | obj_Retrieve |
| Description | It recovers object fields value in Object ID database |
| Input parameters | String Query – SQL query used to retrieve data from the database |
| Output parameters | objectDataType ObjectData - set of related object data |
| | |
| Method | ProtectionInfo_Retrieve |
| Description | It recovers ProtectionInfo field value in Object ID database |
| Input parameters | String AXOID – Object ID String ObjectVersion – Version of the object String ProtectionStamp – Protection related timestamp |
| Output parameters | String ProtectionInfo – Information about object protection |
| | |
| Method | FingerprintInfo_Retrieve |
| Description | It recovers FingerprintInfo field value in Object ID database |
| Input parameters | String AXOID – Object ID String ObjectVersion – Version of the object String ProtectionStamp – Protection related timestamp |
| Output parameters | String FingerprintInfo – Information about object fingerprint |
| | |
| Method | Query_exec |
| Description | It allows performing a generic query over database. |
| Input parameters | String Query – Generic query to be performed |
| Output parameters | String QueryResult – Data retrieved by the performed query |

| | |
|-----------------------------|--|
| | String Result – status result of operation: 0 for successful 1 otherwise |
| | |
| Accounting-DBManager | |
| Method | al_Insert |
| Description | It inserts log field values in Action Log table |
| Input parameters | logDataType LogData - set of related log data |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| | |
| Method | al_Delete |
| Description | It deletes logs fields value in Action Log table |
| Input parameters | Integer LogID - ID of related log item |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| | |
| Method | al_Update |
| Description | It updates logs fields value in Action Log table |
| Input parameters | Integer LogID - ID of related log item |
| Output parameters | String Result – result of operation: 0 for successful 1 otherwise |
| | |
| Method | al_Retrieve |
| Description | It recovers logs fields value in Action Log table |
| Input parameters | String Query – SQL query used to retrieve data from the database |
| Output parameters | logDataType LogData - set of related log data |
| | |
| Method | Query_exec |
| Description | It allows performing a generic query over database. |
| Input parameters | String Query – Generic query to be performed |
| Output parameters | String QueryResult – Data retrieved by the performed query String Result – status result of operation: 0 for successful 1 otherwise |
| | |

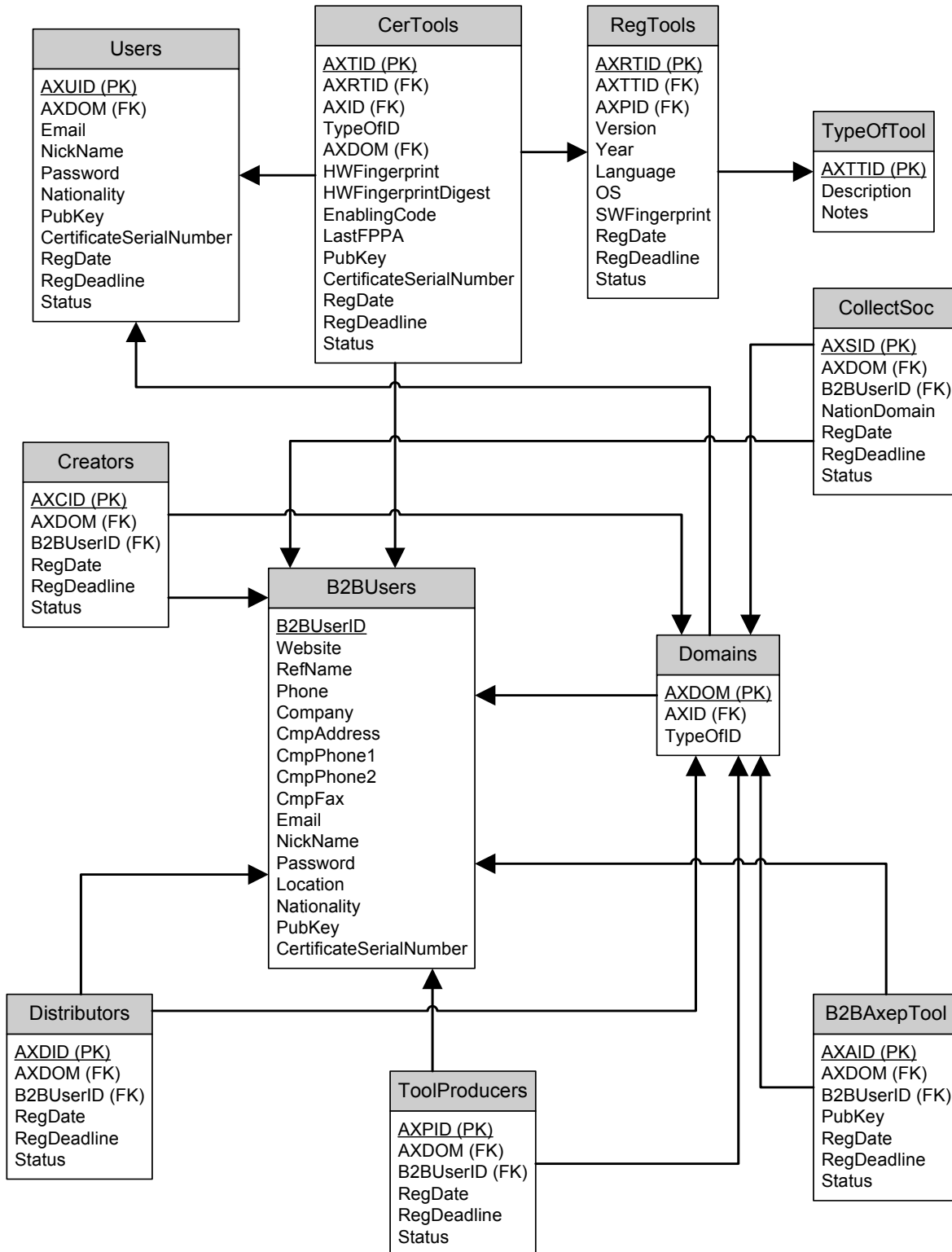
2.9 AXCS registration and certification database (DSI, EXITECH)

2.9.1 Entity-Relationship description

In order to define the AXCS Registration and Certification Database schema, first have to identify the entities and related relations. Here is reported the list of the identified entities and the related meaning.

1. **Users:** this entity contains a part of data about AXMEDIS end users, requested by AXCS to perform its work.
2. **B2BUsers:** this entity contains general data about B2B users like Creators, Distributors, Collecting Society, Tool Producers and so on.
3. **Domains:** this entity contains information about AXMEDIS Domains. A Domain can be referred to a User (if the PMS is for private use at home) or to a B2BUser (if the PMS is located in an organization like a company or a school). It is linked to the parent entity User or B2BUsers according to the value of TypeOdID which identifies if the relative AXID is a AXUID or a B2BUserID.
4. **Creators:** this entity contains specific data about Object Creators, Integrators and Producers. It is linked to the parent entity B2BUsers.
5. **Distributors:** this entity contains specific data about Object Distributors. It is linked to the parent entity B2BUsers.
6. **B2BAxepTool:** this entity contains data about B2B AxepTool. It is linked to the parent entity B2BUsers.
7. **CollectSoc:** this entity contains data about Collecting Societies. It is linked to the parent entity B2BUsers.
8. **ToolProducers:** this entity contains data about Tool Producers. It is linked to the parent entity B2BUsers.
9. **RegTools:** this entity contains data about Registered Tools. The “registration” term refers to Tool Off-line Registration scenario. A registered tool is a software product. An instance of a Registered Tool running on a terminal becomes a Certified Tool. A registered tool is identified by an ID called AXRTID.
10. **TypeOfTool:** this entity contains all the type of tool that can be used in the AXMEDIS system.
 A set of possible tool type is:
 Composition Engine
 Formatting Engine
 Editor PC/MAC
 Viewer PC/MAC
 Viewer/player: PDA
 Viewer/player: mobile
 Protection Tool Editor
 AXEPTool
 Programme & Publication Engine
 Publication Tool
 Generator from CMS
 AXCS
 Super AXCS
 AXMEDIS OID Generator
 AXMEDIS PMS Client
 AXMEDIS PMS Home
 AXMEDIS PMS Server
 PLUGIN xxxxx
 PLUGIN yyyyy
11. **CerTools:** this entity contains data about Certified Tools. The “certification” term refers to Certification of a Tool/User scenario. A certified tool is an instance of a tool running on a terminal.

A certified tool is identified by an ID called AXTID field. It is linked to the parent entity User or B2BUsers according to the value of TypeOfID which identifies if the relative AXID is a AXUID or a B2BUserID.



2.9.2 Relational database schema extended description

Here is reported the list of identified tables came from entities and relations previously stated.

Users

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------------------|----|-----|----------------------|-------------|-------------------------|
| AXUID | PK | I | C-Large Length | Not allowed | |
| AXDOM | FK | I | C-Large Length | Allowed | |
| Email | | I | C-Large Length | Not allowed | |
| NickName | | I | C-Large Length | Not allowed | |
| Password | | | C-Large Length | Not allowed | |
| Nationality | | I | C-Large Length | Not allowed | |
| PubKey | | | C-Large Length | Not allowed | |
| CertificateSerialNumber | | I | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

Column details

1. AXUID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the user

2. AXDOM (FK)

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: AXMEDIS Current Domain of the user (if any)

3. Email

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: email of the user

4. NickName

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: NickName of the user

5. Password

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: MD5 or other encryption of user password

6. Nationality

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Nationality of the user

7. PubKey

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: public key of the user

8. CertificateSerialNumber

Physical data type: LONGTEXT

AXMEDIS Project

CONFIDENTIAL

Allow NULLs: Not allowed
Notes: user certificate serial number

9. RegDate

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of user registration

10. RegDeadline

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of user registration end

11. Status

Physical data type: VARCHAR(1)
Allow NULLs: Not Allowed
Notes: Status of the user: B/U (Blocked/Unblocked)

B2BUsers

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------------------|----|-----|----------------|-------------|-------------|
| B2BUserID (PK) | PK | I | C-Large Length | Not allowed | |
| Website | | | C-Large Length | Allowed | |
| RefName | | I | C-Large Length | Allowed | |
| Phone | | | C-Large Length | Allowed | |
| Company | | I | C-Large Length | Allowed | |
| CmpAddress | | | C-Large Length | Allowed | |
| CmpPhone1 | | | C-Large Length | Allowed | |
| CmpPhone2 | | | C-Large Length | Allowed | |
| CmpFax | | | C-Large Length | Allowed | |
| Email | | I | C-Large Length | Not allowed | |
| NickName | | I | C-Large Length | Not allowed | |
| Password | | | C-Large Length | Not allowed | |
| Location | | | C-Large Length | Not allowed | |
| Nationality | | I | C-Large Length | Not allowed | |
| PubKey | | | C-Large Length | Not allowed | |
| CertificateSerialNumber | | | C-Large Length | Not allowed | |

Column details**1. B2BUserID (PK)**

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the related B2B user

2. Website

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: email of the B2B user

3. RefName

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Reference name in the B2B company

4. Phone

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Reference telephone number in the B2B company

5. Company

Physical data type: LONGTEXT

AXMEDIS Project

CONFIDENTIAL

| | |
|---|--|
| Allow NULLs: | Allowed |
| Notes: | Name of the B2B company |
| <u>6. CmpAddress</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Address of the B2B company |
| <u>7. CmpPhone1</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Phone number of the B2B company |
| <u>8. CmpPhone2</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Phone number of the B2B company |
| <u>9. CmpFax</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Address of the B2B company |
| <u>10. Email</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not Allowed |
| Notes: | Email of the B2B user |
| <u>11. NickName</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | NickName of the B2B user |
| <u>12. Password</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | MD5 or other encryption of the B2B user password |
| <u>13. Location</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Nationality of the B2B user |
| <u>14. Nationality</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Nationality of the B2B user |
| <u>15. PubKey</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | public key of the B2B user |
| <u>16. CertificateSerialNumber</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | user certificate serial number |

Domains

| | |
|--------------------------------|---|
| Number of indexes: | ? |
| Number of foreign keys: | ? |

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|----------|----|-----|-----------------------|-------------|-----------------|
| AXDOM | PK | I | C-Large Length | Not allowed | |
| AXID | FK | I | C-Large Length | Not Allowed | |
| TypeOfID | | | C-Variable Length(10) | Not allowed | B2BUserID/AXUID |

Column details**1. AXDOM (PK)**

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: ID of the AXMEDIS Domain

2. AXID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: It's the Domain Manager ID: it can be an ID of a B2BUser or an End User according to the value of TypeOfID

3. TypeOfID

Physical data type: VARCHAR(10)
 Allow NULLs: Not allowed
 Notes: It refers to AXID and can have one of the two values: B2BUserID/AXUID

Creators

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------|----|-----|----------------------|-------------|-------------------------|
| AXCID | PK | I | C-Large Length | Not allowed | |
| AXDOM | FK | I | C-Large Length | Allowed | |
| B2BUserID | FK | I | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

Column details**1. AXCID (PK)**

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: ID of the creator

2. AXDOM (FK)

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: AXMEDIS Current Domain of the creator (if any)

3. B2BUserID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: ID of the related B2B user

4. RegDate

Physical data type: DATETIME
 Allow NULLs: Not allowed
 Notes: Timestamp of creator registration

5. RegDeadline

Physical data type: DATETIME
 Allow NULLs: Not allowed
 Notes: Timestamp of creator registration end

6. Status

Physical data type: VARCHAR(1)

Allow NULLs: Not Allowed
Notes: Status of the creator: B/U (Blocked/Unblocked)

Distributors

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------|----|-----|----------------------|-------------|-------------------------|
| AXDID | PK | I | C-Large Length | Not allowed | |
| AXDOM | FK | I | C-Large Length | Allowed | |
| B2BUserID | FK | I | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

Column details

1. AXDID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the distributor

2. AXDOM (FK)

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: AXMEDIS Current Domain of the distributor (if any)

3. B2BUserID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the related B2B user

4. RegDate

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of distributor registration

5. RegDeadline

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of distributor registration end

6. Status

Physical data type: VARCHAR(1)
Allow NULLs: Not Allowed
Notes: Status of the distributor: B/U (Blocked/Unblocked)

B2BAxepTool

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------|----|-----|----------------------|-------------|-------------------------|
| AXAID | PK | I | C-Large Length | Not allowed | |
| AXDOM | FK | I | C-Large Length | Not allowed | |
| B2BUserID | FK | I | C-Large Length | Not allowed | |
| PubKey | | | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

Column details**1. AXAID (PK)**

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: ID of the AXEPTool

2. AXDOM (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not Allowed
 Notes: AXMEDIS Domain of the AXEPTool (if any)

3. B2BUserID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: ID of the related B2B user

4. PubKey

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: public key of the user

5. RegDate

Physical data type: DATETIME
 Allow NULLs: Not allowed
 Notes: Timestamp of AXEPTool registration

6. RegDeadline

Physical data type: DATETIME
 Allow NULLs: Not allowed
 Notes: Timestamp of AXEPTool registration end

7. Status

Physical data type: VARCHAR(1)
 Allow NULLs: Not Allowed
 Notes: Status of the AXEPTool: B/U (Blocked/Unblocked)

CollectSoc

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|----------------|----|-----|-----------------------|-------------|-------------------------|
| AXSID (PK) | PK | I | C-Large Length | Not allowed | |
| AXDOM (FK) | FK | I | C-Large Length | Allowed | |
| B2BUserID (FK) | FK | I | C-Large Length | Not allowed | |
| NationDomain | | I | C-Variable Length(20) | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

Column details**1. AXSID (PK)**

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: ID of the collecting society

2. AXDOM (FK)

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: AXMEDIS CurrentDomain of the collecting society (if any)

3. B2BUserID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the related B2B user

4. NationDomain

Physical data type: VARCHAR(20)
Allow NULLs: Not allowed
Notes: Nation related to the collecting society.

5. RegDate

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of collecting society registration

6. RegDeadline

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of collecting society registration end

7. Status

Physical data type: VARCHAR(1)
Allow NULLs: Not Allowed
Notes: Status of the collecting society: B/U (Blocked/Unblocked)

ToolProducers

Number of indexes: ?0
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|----------------|----|-----|----------------------|-------------|-------------------------|
| AXPID (PK) | PK | I | C-Large Length | Not allowed | |
| AXDOM (FK) | FK | I | C-Large Length | Allowed | |
| B2BUserID (FK) | FK | I | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

Column details**1. AXPID (PK)**

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the tool producer

2. AXDOM (FK)

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: AXMEDIS Domain of the tool producer (if any)

3. B2BUserID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the related B2B user

4. RegDate

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of tool producer registration

5. RegDeadline

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of tool producer registration end

6. Status

Physical data type: VARCHAR(1)

AXMEDIS Project

Allow NULLs: Not Allowed
Notes: Status of tool producer: B/U (Blocked/Unblocked)

RegTools

(This table refers to Tool Off-line Registration)

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|---------------|----|-----|----------------------|-------------|-------------------------|
| AXRTID (PK) | PK | I | C-Large Length | Not allowed | |
| AXTTID (FK) | FK | I | C-Large Length | Not allowed | |
| AXPID (FK) | FK | I | C-Large Length | Not allowed | |
| Version | | | C-Large Length | Not allowed | |
| Year | | | C-Large Length | Not allowed | |
| Language | | I | C-Large Length | Not allowed | |
| OS | | I | C-Large Length | Not allowed | |
| SWFingerprint | | | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

Column details

1. AXRTID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the registered tool (ID of the the class of registered tool given by SuperAXCS)

2. AXTTID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the type of tool

3. AXPID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the tool producer

4. Version

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: version of the registered tool

5. Year

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: year of the registered tool

6. Language

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: language of the registered tool

7. OS

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Operative System of the registered tool

8. SWFingerprint

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Software fingerprint of the registered tool

9. RegDate

AXMEDIS Project

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of tool producer registration

10. RegDeadline

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of tool producer registration end

11. Status

Physical data type: VARCHAR(1)
Allow NULLs: Not Allowed
Notes: Status of tool producer: B/U (Blocked/Unblocked)

TypeOfTool

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------|----|-----|----------------|-------------|-------------|
| AXTTID (PK) | PK | I | C-Large Length | Not allowed | |
| Description | | I | C-Large Length | Not allowed | |
| Notes | | | C-Large Length | Allowed | |

Column details**1. AXTTID (PK)**

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the type of tool

2. Description

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Description of the type of tool.

3. Notes

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: additional information about the type of tool

CerTools

(This table refers to the “Certification of a Tool” scenario)

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------------------|----|-----|-----------------------|-------------|-------------------------|
| AXTID | PK | I | C-Large Length | Not allowed | |
| AXRTID | FK | I | C-Large Length | Not allowed | |
| AXID | FK | I | C-Large Length | Not allowed | |
| TypeOfID | | | C-Variable Length(10) | Not allowed | B2BUserID/AXUID |
| AXDOM | FK | I | C-Large Length | Allowed | |
| HWFingerprint | | | C-Large Length | Not allowed | |
| HWFingerprintDigest | | | C-Large Length | Not allowed | |
| EnablingCode | | | C-Large Length | Not allowed | |
| LastFPPA | | | C-Large Length | Not allowed | |
| PubKey | | | C-Large Length | Not allowed | |
| CertificateSerialNumber | | | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Not allowed | B/U (Blocked/Unblocked) |

| Column details | |
|---|---|
| <u>1. AXTID (PK)</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | ID of the certified tool (the single instance of the installed tool) |
| <u>2. AXRTID (FK)</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | ID of the related registered tool (ID of the the class of registered tool given by SuperAXCS) |
| <u>3. AXID (FK)</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | related user ID: it can be an ID of a B2BUser or an End User according to the value of TypeOfID |
| <u>4. TypeOfID</u> | |
| Physical data type: | VARCHAR(10) |
| Allow NULLs: | Not allowed |
| Notes: | It refers to AXID and can have one of the two values: B2BUserID/AXUID |
| <u>5. AXDOM (FK)</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | AXMEDIS Domain of certified tool (if any) |
| <u>6. HWFingerprint</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | HW fingerprint of the related terminal (PC or anything else) the software is running on |
| <u>7. DigestHWFingerprint</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Digest of HW fingerprint of the related terminal (PC or anything else) the software is running on |
| <u>8. EnablingCode</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | An activation code provided by AXCS to Certified Tool during the initialization phase. |
| <u>9. LastFPPA</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | LastFPPA: Last Fingerprint of Performed Actions |
| <u>10. PubKey</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | public key of the certified tool |
| <u>11. CertificateSerialNumber</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | tool certificate serial number |
| <u>12. RegDate</u> | |
| Physical data type: | DATETIME |
| Allow NULLs: | Not allowed |

Notes:

Timestamp of certified tool registration

13. RegDeadline

Physical data type:

DATETIME

Allow NULLs:

Not allowed

Notes:

Timestamp of certified tool registration end

14. Status

Physical data type:

VARCHAR(1)

Allow NULLs:

Not Allowed

Notes:

Status of certified tool: B/U (Blocked/Unblocked)

2.10 AXCS Objects ID Database (DSI, EXITECH)

2.10.1 Entity-Relationship description

In order to define the AXCS ObjectID Database schema, first have to identify the entities and related relations. Here is reported the list of the identified entities and the related meaning.

Objects: this entity contains a part of data about objects. It is linked to Creators entity (located in the AXCS Registration and Certification Database) with an 1-N relation and to the Distributors entity (also located in the AXCS Registration and Certification Database) with an M-N relation implemented as a table called **DistributedBy**. It means that an Object can be distributed by more than one Distributor. It is also linked to itself with an N-M relation implemented as a table called **ComposedBy** to state that an Object can be a Complex Object composed by other Objects.

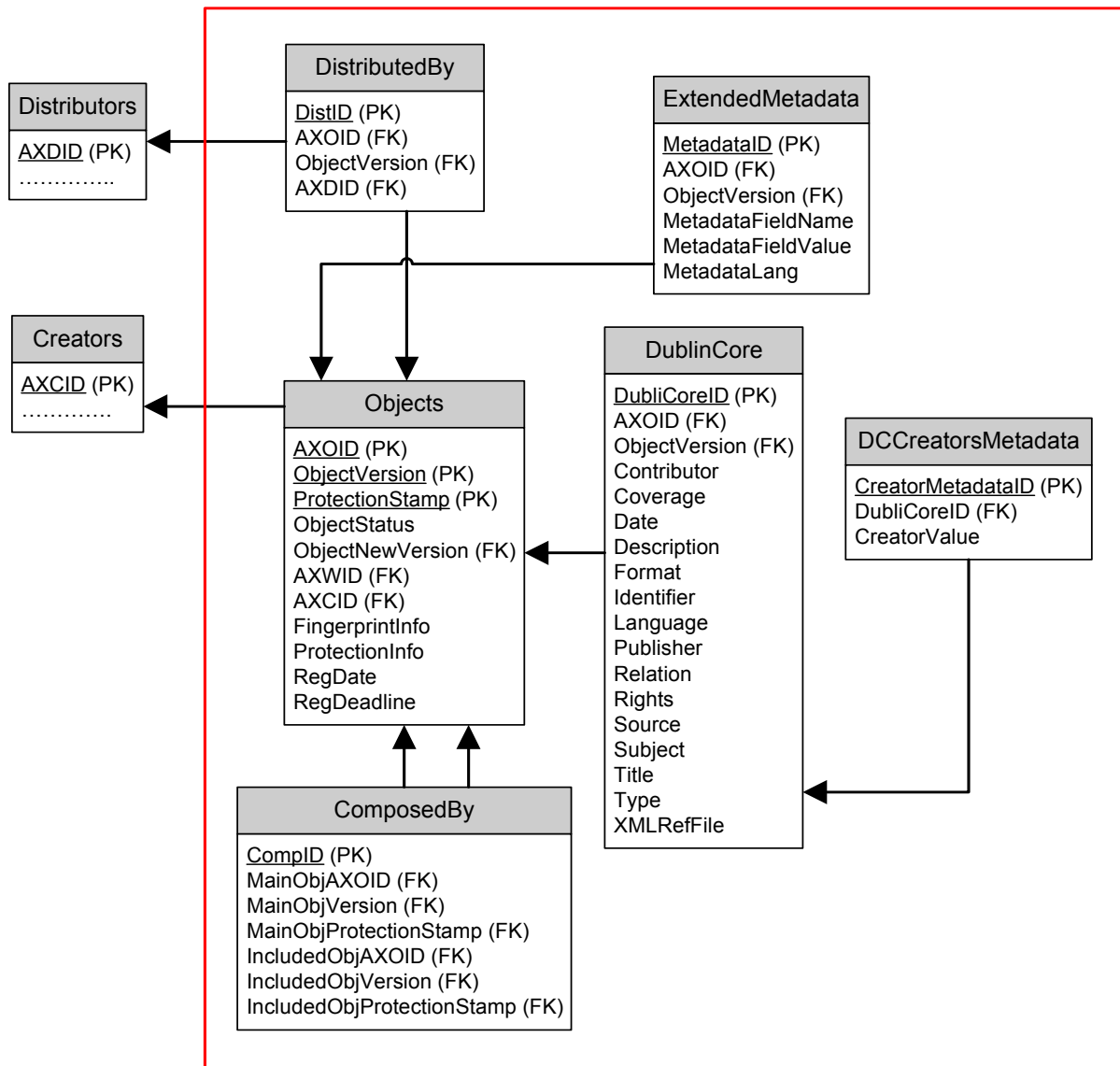
DublinCore: this entity contains Dublin Core metadata related to an Object. Every row in the pertinent relational schema table is a Dublin Core set of metadata description about an Object related to a language.

DCCreatorsMetadata: this entity contains Authors pertinent to a specified set of Dublin Core metadata related to a language. It has been introduced to take into account the more than one Author multiplicity. It is linked to the **DublinCore** entity with an 1-N relation.

ExtendedMetadata: this entity contains optional metadata about Object not included in Dublin Core. Every row in the pertinent relational schema table is a single metadata value. This is a way to have a variable number of metadata fields related to every object.

2.10.2 Relational database schema extended description

Here is reported the list of identified tables came from entities and relations previously stated.



Objects

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|------------------|----|-----|----------------------|-------------|---------------------|
| AXOID | | I | C-Large Length | Not allowed | |
| ObjectVersion | PK | I | C-Large Length | Not allowed | |
| ProtectionStamp | | I | C-Large Length | Not allowed | |
| ObjectStatus | | | C-Variable Length(1) | Not allowed | B/V (Blocked/Valid) |
| ObjectNewVersion | FK | I | C-Large Length | Allowed | |
| AXWID | FK | I | C-Large Length | Not allowed | |
| AXCID | FK | I | C-Large Length | Not allowed | |
| FingerprintInfo | | I | C-Large Length | Not allowed | |
| ProtectionInfo | | | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Allowed | |
| RegDeadline | | | T-Date & Time | Allowed | |

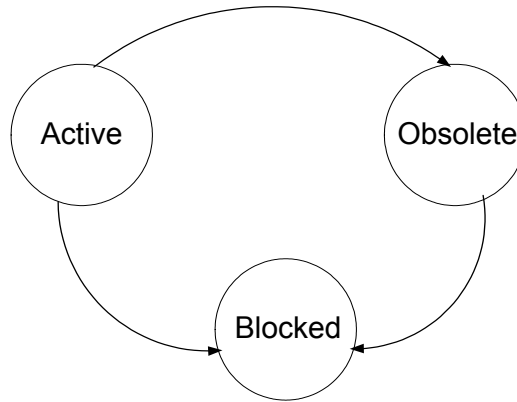
| Column details | |
|---|---|
| 1. AXOID (part of PK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | ID of the Object |
| 2. ObjectVersion (part of PK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Current version of the object |
| 3. ProtectionStamp (part of PK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Indicates the way to protect the related object. It is part of the primary key within ObjectVersion and AXOID |
| 4. ObjectNewVersion (part of FK, with AXOID) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Eventually new version of the object (if any) |
| 5. ObjectStatus | |
| Physical data type: | VARCHAR(1) |
| Allow NULLs: | Not Allowed |
| Notes: | Status of object: B/V (Blocked/Valid) |
| 6. AXWID (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | ID of the Work (the intellectual work) requested by Object Creator |
| 7. AXCID (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | ID of the Object Creator |
| 8. FingerprintInfo | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Fingerprint information related to the object |
| 9. ProtectionInfo | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Protection information related to the object |
| 10. RegDate | |
| Physical data type: | DATETIME |
| Allow NULLs: | Allowed |
| Notes: | Timestamp of the object registration (insertion in the AXMEDIS system) |
| 11. RegDeadline | |
| Physical data type: | DATETIME |
| Allow NULLs: | Allowed |
| Notes: | Timestamp of the object lifecycle end (if any) |

Here we have to specify how the ObjectStatus works and how it can be used. When an object is submitted for the distribution in the AXMEDIS system the ObjectStatus is set to “Active”. This means that the object is ready to use by all the users who got the right to use it. An object can be set to “Blocked” when the system manager or the object owner discovery some problems about it and decide to block the object usage. This block can be temporary or definitive. An object can also become obsolete if a new version of that specific object is issue. In this case the object status is set to “Obsolete” and the object is updated by the new version of object. We have to notice that the AXOID of a new version of the object is the same the old object but the version is different. When an object become “obsolete” can be blocked or not, depending on the will of object owner and user license. An user can automatically have the

AXMEDIS Project

rights to use the new version of the object or must acquire a new right, depending on the license he owns. The system has to flag to the user the availability of a new version of the object. In term of database field values, the object status can be summarized by the following table where is showed the object status depending on the database field values.

| | Active | Obsolete Usable | Obsolete Unusable | Blocked |
|-------------------------|--------|--------------------|--------------------|-----------|
| ObjectStatus | V | V | B | B |
| ObjectNewVersion | NULL | any value not null | any value not null | any value |



ObjectStatus state diagram

ComposedBy

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|----------------------------|----|-----|------------------|-------------|-------------|
| CompID | PK | I | N-Signed Integer | Not allowed | |
| MainObjAXOID | | I | C-Large Length | Not allowed | |
| MainObjVersion | FK | I | C-Large Length | Not allowed | |
| MainObjProtectionStamp | | I | C-Large Length | Not allowed | |
| IncludedObjAXOID | | I | C-Large Length | Not allowed | |
| IncludedObjVersion | FK | I | C-Large Length | Not allowed | |
| IncludedObjProtectionStamp | | I | C-Large Length | Not allowed | |

Column details

1. CompID

Physical data type: LONG
 Allow NULLs: Not allowed
 Notes: Table primary key

2. MainObjAXOID (part of FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Main Object ID (ID of the complex Object)

3. MainObjVersion (part of FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Main Object version (version of the complex Object)

4. MainObjProtectionStamp (part of FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Indicates the way to protect the related object.

5. IncludedObjAXOID (FK)

Physical data type: LONG
Allow NULLs: Not allowed
Notes: ID of the Object included in the related Main Object ID (MainAXOID)

6. IncludedObjVersion (part of FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Version of the included Object (referred to IncludedAXOID)

7. IncludedObjProtectionStamp (part of FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Indicates the way to protect the related object.

DistributedBy

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-----------------|----|-----|------------------|-------------|-------------|
| DistID | PK | I | N-Signed Integer | Not allowed | |
| AXOID | | I | C-Large Length | Not allowed | |
| ObjectVersion | FK | I | C-Large Length | Not allowed | |
| ProtectionStamp | | I | C-Large Length | Not allowed | |
| AXDID | FK | I | C-Large Length | Not allowed | |

Column details**1. DistID (PK)**

Physical data type: LONG
Allow NULLs: Not allowed
Notes: ID of the row. It can be an auto-incremental ID and is used only by DB Manager and API DB.

2. ObjectVersion (part of FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Current version of the object

3. AXOID (part of FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the Object

4. ProtectionStamp (part of FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Indicates the way to protect the related object.

5. AXDID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the Distributor

DublinCore

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------|----|-----|------------------|-------------|-------------|
| DubliCoreID | PK | I | N-Signed Integer | Not allowed | |
| AXOID | FK | I | C-Large Length | Not allowed | |

| | | | | |
|----------------------|----------|-------------------|-------------|------------------------|
| ObjectVersion | I | C-Large Length | Not allowed | |
| Contributor | | C-Large Length | Allowed | |
| Coverage | | C-Large Length | Allowed | |
| Date | | T-Date & Time | Allowed | |
| Description | | C-Large Length | Allowed | |
| Format | | C-Large Length | Allowed | |
| Identifier | | C-Large Length | Allowed | |
| Language | | C-Fixed Length(3) | Allowed | See ISO 639-2 standard |
| Publisher | | C-Large Length | Allowed | |
| Relation | | C-Large Length | Allowed | |
| Rights | | C-Large Length | Allowed | |
| Source | | C-Large Length | Allowed | |
| Subject | | C-Large Length | Allowed | |
| Title | | C-Large Length | Allowed | |
| Type | | C-Large Length | Allowed | |
| XMLRefFile | | C-Large Length | Allowed | |

1. DublinCoreID

Physical data type: LONG
Allow NULLs: Not allowed
Notes: Dublin Core metadata ID. Table primary-key

2. AXOID (part of FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the related Object

3. ObjectVersion (part of FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Current version of the object

4. Contributor

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Dublin Core metadata field

5. Coverage

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Dublin Core metadata field

6. Date

Physical data type: DATETIME
Allow NULLs: Allowed
Notes: Dublin Core metadata field

7. Description

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Dublin Core metadata field

8. Format

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Dublin Core metadata field

9. Identifier

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Dublin Core metadata field

10. Language

Physical data type: VARCHAR(3)
Allow NULLs: Allowed
Notes: Dublin Core metadata field. Its value is a language code according to ISO 639-2

standard.

11. Publisher**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Dublin Core metadata field

12. Relation**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Dublin Core metadata field

13. Rights**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Dublin Core metadata field

14. Source**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Dublin Core metadata field

15. Subject**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Dublin Core metadata field

16. Title**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Dublin Core metadata field

17. Type**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Dublin Core metadata field

18. XMLRefFile**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

Link to an XML File containing Dublin Core Metadata

DCCreatorsMetadata**Number of indexes:**

?

Number of foreign keys:

?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------------|----|-----|------------------|-------------|-------------|
| CreatorMetadataID | PK | I | N-Signed Integer | Not allowed | |
| DubliCoreID | FK | I | N-Signed Integer | Not allowed | |
| CreatorValue | | | C-Large Length | Not allowed | |

1. CreatorMetadataID**Physical data type:**

LONG

Allow NULLs:

Not Allowed

Notes:

CreatorMetadata ID. Table primary-key

2. DublinCoreID**Physical data type:**

LONG

Allow NULLs:

Not allowed

Notes:

Related Dublin Core metadata ID.

3 CreatorValue

AXMEDIS Project

CONFIDENTIAL

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Creator metadata field value

ExtendedMetadata

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|--------------------|----|-----|-------------------|-------------|------------------------|
| MetadataID | PK | I | N-Signed Integer | Not allowed | |
| AXOID | FK | I | N-Signed Integer | Not allowed | |
| ObjectVersion | | I | C-Large Length | Not allowed | |
| MetadataFieldName | | | C-Large Length | Not allowed | |
| MetadataFieldValue | | | C-Large Length | Not allowed | |
| MetadataLang | | | C-Fixed Length(3) | Not allowed | See ISO 639-2 standard |

1. MetadataID

Physical data type: LONG
 Allow NULLs: Not Allowed
 Notes: Metadata ID. Table primary-key

2. AXOID (part of FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: ID of the related Object

3. ObjectVersion (part of FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Current version of the object

4. MetadataFieldName

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Metadata field name

5. MetadataFieldValue

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Metadata field value

6. MetadataLang

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: Metadata field language code according to ISO 639-2 standard.

2.11 AXCS Accounting Database (DSI, EXITECH)

2.11.1 Action log received by AXCS (FUPF)

This section describes the information received by the AXCS when an event report is generated.

The information received by the AXCS could be:

- An ActionLog entity or a list of ActionLog entities
- A SupervisorInputData entity

The ActionLog entity has the next elements:

- LogID : Registration ID in Action-Log Registry. Identifies an ActionLog.
- AXOID: The action log references to a certain Axmedis Object, that it is identified by its ID.
- ObjectVersion: Reference to an Axmedis Object version. The action log references to a certain version of the object related.
- ProtectionStamp: Indicates the way to protect the related object.
- AXWID: Indicates the Work ID. This element is not necessary.
- AXDOM: Indicates the User Domain, if the user has a domain related.
- AXUID: The action log references to a certain User, that it is identified by its ID.
- AXDID: Indicates the pertinent Object Distributor ID.
- AXCID: Indicates the pertinent Object Creator ID.
- OwnerName: Indicates the pertinent Object Owner.
- AXTID: The action log references to a certain certified tool, that it is identified by its ID.
- AXLID: Indicates the pertinent License ID.
- Location: Indicates the nation.
- OperationDetailsIDPk : Reference to an operation details.
- OperationIDPk: Reference to an operation.
- RegistrationTimestamp: Time of the operation Registration.
- ExecutionTimestamp: Time of the operation Execution.
- InstantLastFPPA: Last Fingerprint of Performed Actions.
- EstimatedHWFingerprint: Indicates the estimated HW Fingerprint of the related terminal.

The SupervisorInputData entity has the elements of an ActionLog entity and a new element:

- AdditionalData: Additional information to the Supervisor to control and check, that any action is being done correctly.

2.11.2 Entity-Relationship description (DSI)

The Account Log database is structured to take care of Action-Logs and once it will be standardized will have to implement all the necessary parts of MPEG21 Event Reporting. In order to define the AXCS Accounting Database schema, first have to identify the entities and related relations. Here is reported the list of the identified entities and the related meaning.

ActionLog: this entity stores the Action-Log as it is with some fixed information such as the AXOID on which the operation is performed, the AXUID of the user that performed the operation, the registration timestamp and execution timestamp (that can be different because of off-line operations performed on the objects). This entity is linked to the **OperationDetails** entity described below.

Operations: this entity will list all the allowed operations for all objects with a description and some other (at the moment not detailed) fields.

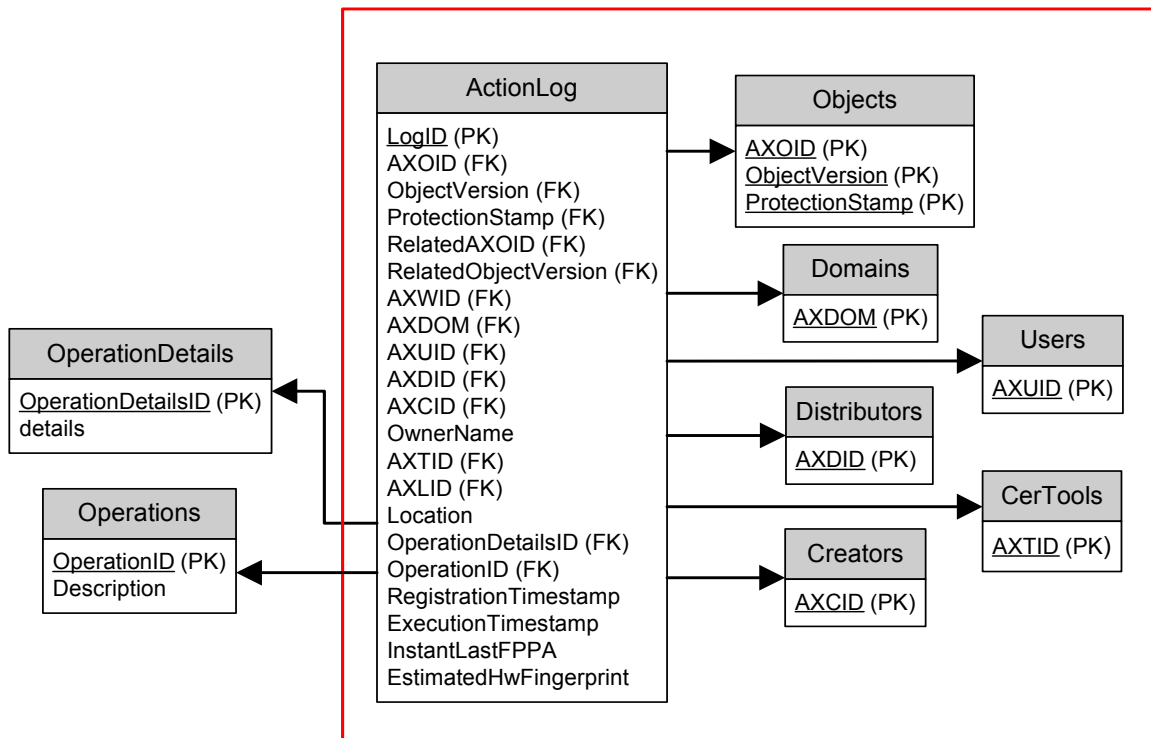
These operations are based on the RDD terms of MPEG-21 standard, which have the concept of operations associated to an Event Report (Action Log for us). We have based our proposal of structure from this. We consider only operations made over an AXMEDIS Object. These operations generate an Action Log.

A list of these operations is:

- Modify : To edit an object in order to change it, or to protect it adding rules or metadata.
- Aggregate : To obtain an AXMEDIS Object as a composition of AXMEDIS Objects
- Render : To use or view an Object
- Play : Render as Performance

- Print : Render as Fixation
- Originate : To create a new AXMEDIS Object
- Enlarge : To Add something to an AXMEDIS Object already created
- Reduce : To modify an AXMEDIS Object by taking away from it
- Diminish : To create a new AXMEDIS Object from another one. The Object created is smaller than the source.
- Adapt : To Copy. To edit an AXMEDIS Object creating a new one which has the changes.
- Embed : To put an element or AXMEDIS Object into another AXMEDIS Object
- Delete : To destroy an AXMEDIS Object
- Identify : To nominate an AXMEDIS Object uniquely

OperationDetails: this entity will contain the details of the operations such as channel, duration and other details that are to be specified in deep. It is useful for information purpose.



2.11.3 Relational database schema extended description (DSI)

Here is reported the list of identified tables came from entities and relations previously stated.

ActionLog

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | Idx | Data type | Allow NULLs | Value/Range |
|---------------|----|-----|------------------|-------------|-------------|
| <u>LogID</u> | PK | I | N-Signed Integer | Not allowed | |
| AXOID | FK | I | C-Large Length | Not allowed | |
| ObjectVersion | | I | C-Large Length | Not allowed | |

AXMEDIS Project

CONFIDENTIAL

| | | | | |
|-------------------------------|-----------|----------|------------------|-------------|
| ProtectionStamp | | I | C-Large Length | Not allowed |
| AXWID | FK | I | C-Large Length | Allowed |
| AXDOM | FK | I | C-Large Length | Allowed |
| AXUID | FK | I | C-Large Length | Not allowed |
| AXDID | FK | I | C-Large Length | Not allowed |
| AXCID | FK | I | C-Large Length | Not allowed |
| OwnerName | | | C-Large Length | Allowed |
| AXTID | FK | I | C-Large Length | Not allowed |
| AXLID | FK | I | C-Large Length | Not allowed |
| Location | | | C-Large Length | Allowed |
| OperationDetailsID | FK | | N-Signed Integer | Not allowed |
| OperationID | FK | | N-Signed Integer | Not allowed |
| RegistrationTimestamp | | | T-Date & Time | Not allowed |
| ExecutionTimestamp | | I | T-Date & Time | Not allowed |
| InstantLastFPPA | | | C-Large Length | Not allowed |
| EstimatedHwFingerprint | | | C-Large Length | Not allowed |

Column details

1. LogID

Physical data type: LONG
 Allow NULLs: Not allowed
 Notes: Current registration ID in Action-Log Registry

2. AXOID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object ID. Foreign Key of an AXMEDIS Object Table.

3. ObjectVersion (part of FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object version. Foreign Key of an AXMEDIS Object Table.

4. ProtectionStamp (part of PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Indicates the way to protect the related object.

5. AXWID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: Pertinent Work Identification

6. AXDOM (FK)

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: Pertinent User AXMEDIS Current Domain (if any)

7. AXUID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent User ID. Foreign Key of an authorized AXMEDIS User Table. The Event Report was prompted by a user with a certain AXUID (User ID).

8. AXDID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object Distributor ID

9. AXCID (FK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object Creator ID

10. OwnerName

Physical data type: LONGTEXT

AXMEDIS Project

Allow NULLs: Allowed
Notes: Pertinent Object Owner

11. AXTID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: ID of the certified tool (the single instance of the installed tool). Foreign Key of an authorized AXMEDIS Peer Table. The Event Report was prompted from a peer (a tool) with a certain ID (AXTID).

12. AXLID (FK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Pertinent Licence ID

13. Location

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: Nation related to the pertinent collecting society.

14. OperationDetailsIDPk (FK)

Physical data type: LONG
Allow NULLs: Not allowed
Notes:

15. OperationIDPk (FK)

Physical data type: LONG
Allow NULLs: Not allowed
Notes: Foreign Key of a DI Operation table.

16. RegistrationTimestamp

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: timestamp of the registration in the AXCS

17. ExecutionTimestamp

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: timestamp of operation execution

18. InstantLastFPPA

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: User related LastFPPA: Last Fingerprint of Performed Actions

19. EstimatedHWFingerprint

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Estimated HW fingerprint of the related terminal (PC or anything else) the software is running on at the specific action execution time

OperationDetails

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|--------------------|----|-----|------------------|-------------|-------------|
| OperationDetailsID | PK | I | N-Signed Integer | Not allowed | |
| details | | | C-Large Length | Not allowed | |

Column details**1. OperationDetailsID**

Physical data type: LONG
 Allow NULLs: Not allowed

2. details

Physical data type: LONGTEXT
 Allow NULLs: Not allowed

Operations

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------|----|-----|------------------|-------------|-------------|
| OperationID | PK | I | N-Signed Integer | Not allowed | |
| Description | | I | C-Large Length | Allowed | |

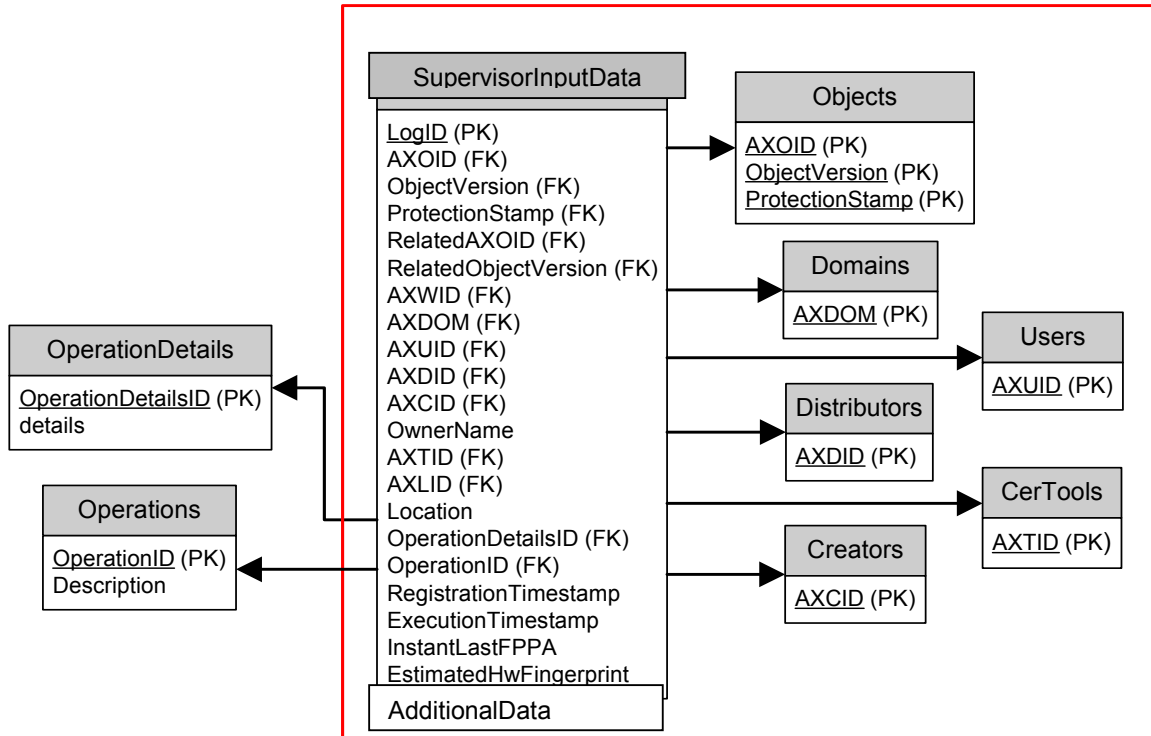
Column details**1. OperationID**

Physical data type: LONG
 Allow NULLs: Not allowed

2. Description

Physical data type: LONGTEXT
 Allow NULLs: Allowed

The following additional table has been added, to store the Supervisor Input Data. The Supervisor Input Data has the same structure that the Action Log but with an additional field, the additionaldata, to store internal actions or actions between modules. It will be used to check that some internal operations, like authorisation, key requests, etc, have been done, and check periodically that the actionlogs corresponding to these operations have been correctly added into the action log database. The new table has the following structure:



SupervisorInputData

Number of indexes: ?
 Number of foreign keys: ?

| Columns | | Idx | Data type | Allow NULLs | Value/Range |
|------------------------|----|-----|------------------|-------------|-------------|
| <u>LogID</u> | PK | I | N-Signed Integer | Not allowed | |
| AXOID | | I | C-Large Length | Not allowed | |
| ObjectVersion | FK | I | C-Large Length | Not allowed | |
| ProtectionStamp | | I | C-Large Length | Not allowed | |
| AXWID | FK | I | C-Large Length | Allowed | |
| AXDOM | FK | I | C-Large Length | Allowed | |
| AXUID | FK | I | C-Large Length | Not allowed | |
| AXDID | FK | I | C-Large Length | Not allowed | |
| AXCID | FK | I | C-Large Length | Not allowed | |
| OwnerName | | | C-Large Length | Allowed | |
| AXTID | FK | I | C-Large Length | Not allowed | |
| AXLID | FK | I | C-Large Length | Not allowed | |
| Location | | | C-Large Length | Allowed | |
| OperationDetailsID | FK | | N-Signed Integer | Not allowed | |
| OperationID | FK | | N-Signed Integer | Not allowed | |
| RegistrationTimestamp | | | T-Date & Time | Not allowed | |
| ExecutionTimestamp | | I | T-Date & Time | Not allowed | |
| InstantLastFPPA | | | C-Large Length | Not allowed | |
| EstimatedHwFingerprint | | | C-Large Length | Not allowed | |
| AdditionalData | | | C-Large Length | Not allowed | |

Column details

1. LogID

Physical data type: LONG
 Allow NULLs: Not allowed
 Notes: Current registration ID in Action-Log Registry

2. AXOID (FK)

Physical data type: LONGTEXT
 AXMEDIS Project

CONFIDENTIAL

| | |
|---|--|
| Allow NULLs: | Not allowed |
| Notes: | Pertinent Object ID. Foreign Key of an AXMEDIS Object Table. |
| 3. <u>ObjectVersion</u> (part of FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Pertinent Object version. Foreign Key of an AXMEDIS Object Table. |
| 4. <u>ProtectionStamp</u> (part of PK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Indicates the way to protect the related object. |
| 5. <u>AXWID</u> (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Pertinent Work Identification |
| 6. <u>AXDOM</u> (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Pertinent User AXMEDIS Current Domain (if any) |
| 7. <u>AXUID</u> (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Pertinent User ID. Foreign Key of an authorized AXMEDIS User Table. The Event Report was prompted by a user with a certain AXUID (User ID). |
| 8. <u>AXDID</u> (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Pertinent Object Distributor ID |
| 9. <u>AXCID</u> (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Pertinent Object Creator ID |
| 10. <u>OwnerName</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Pertinent Object Owner |
| 11. <u>AXTID</u> (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | ID of the certified tool (the single instance of the installed tool). Foreign Key of an authorized AXMEDIS Peer Table. The Event Report was prompted from a peer (a tool) with a certain ID (AXTID). |
| 12. <u>AXLID</u> (FK) | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Not allowed |
| Notes: | Pertinent Licence ID |
| 13. <u>Location</u> | |
| Physical data type: | LONGTEXT |
| Allow NULLs: | Allowed |
| Notes: | Nation related to the pertinent collecting society. |
| 14. <u>OperationDetailsIDPk</u> (FK) | |
| Physical data type: | LONG |
| Allow NULLs: | Not allowed |
| Notes: | |

15. OperationIDPk (FK)

Physical data type: LONG
Allow NULLs: Not allowed
Notes: Foreign Key of a DI Operation table.

16. RegistrationTimestamp

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: timestamp of the registration in the AXCS

17. ExecutionTimestamp

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: timestamp of operation execution

18. InstantLastFPPA

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: User related LastFPPA: Last Fingerprint of Performed Actions

19. EstimatedHWFingerprint

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Estimated HW fingerprint of the related terminal (PC or anything else) the software is running on at the specific action execution time

20. AdditionalData

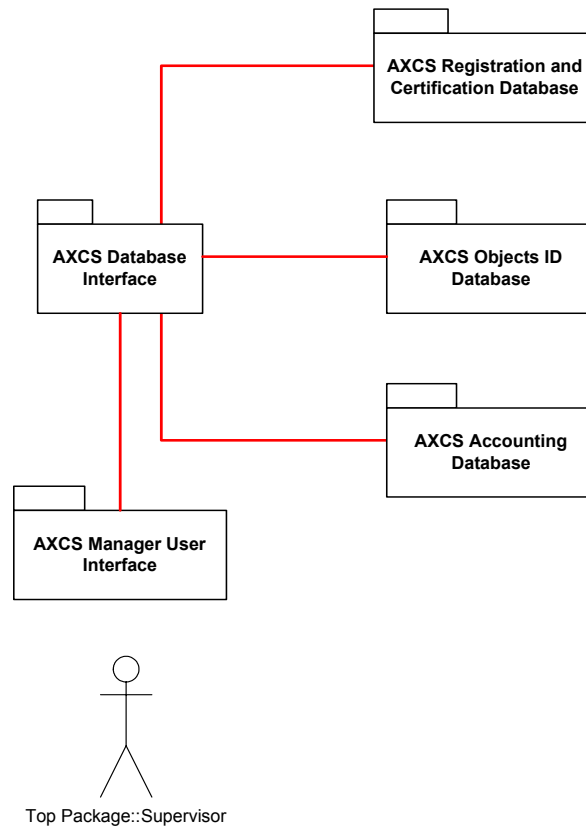
Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Additional Data added to control and check that any action is being done correctly.

2.12 AXCS Manager User Interface (DSI, EXITECH)

| Module Profile | | |
|------------------------------------|-----------------------------------|--|
| AXCS Manager User Interface | | |
| Executable or Library(Support) | Web application | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXCS:Database Interface | | Not protected (with Database Interface), protected human interface |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| Web interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

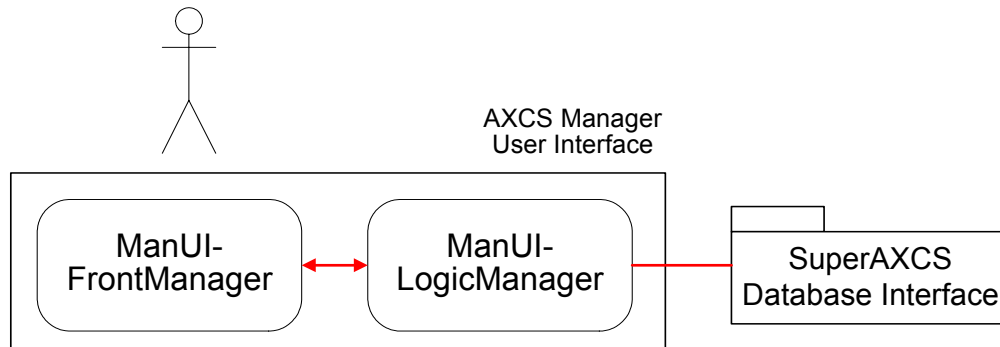
This module has been provided to perform some administrative task needed by AXCS manager. Using this module an AXCS manager can execute many jobs related to database maintenance and check.

2.12.1 AXCS Manager User Interface logic



The AXCS Manager User Interface is a module implementing a web interface intended to provide AXCS manager a simple but powerful interface to perform his administrative tasks. This module has to provide the following functions:

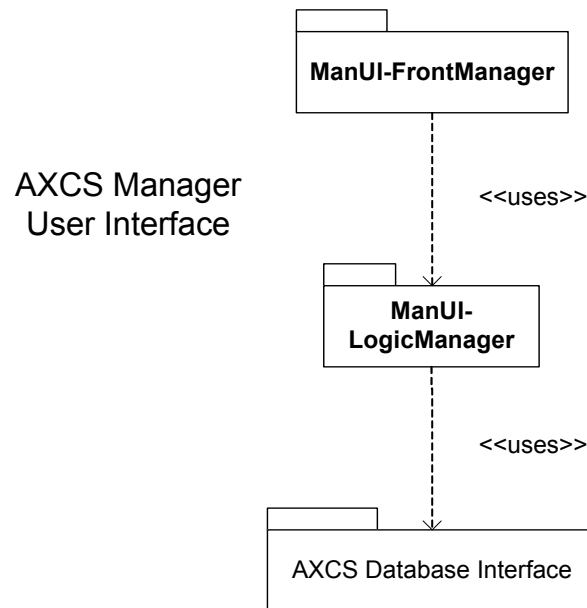
- databases maintenance functions such as insertion, editing, deletion and selection of database fields
- tool, objects, users blocking/unblocking
- log viewing in order to detect forcing system attempts



In order to perform its own task it can be found the following logical decomposition:

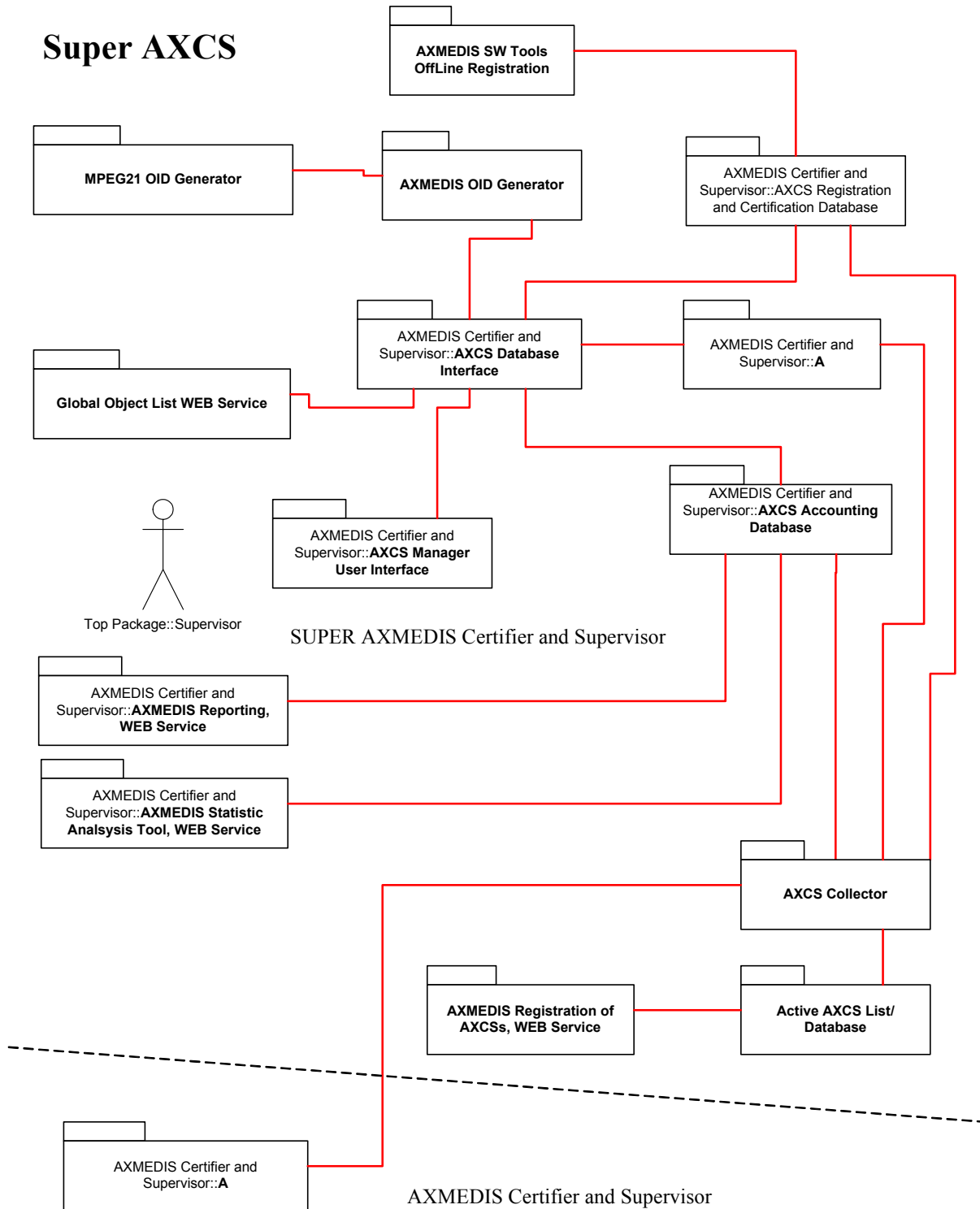
ManUI-FrontManager: it is the web pages manager. It generates all the web pages needful to data collection and data presentation.

ManUI-LogicManager: it realizes the whole application logic and uses AXCS Database Interface (and the related AXDB-API) to access database.



Web pages should be organized so as giving the possibility to user to perform the functions above. Data access views should be organized also by function and by database, in order to aggregate all tables belonging to the same database and to evidence the possible operation that could be performed on.

3 Super AXCS (DSI, EXITECH)



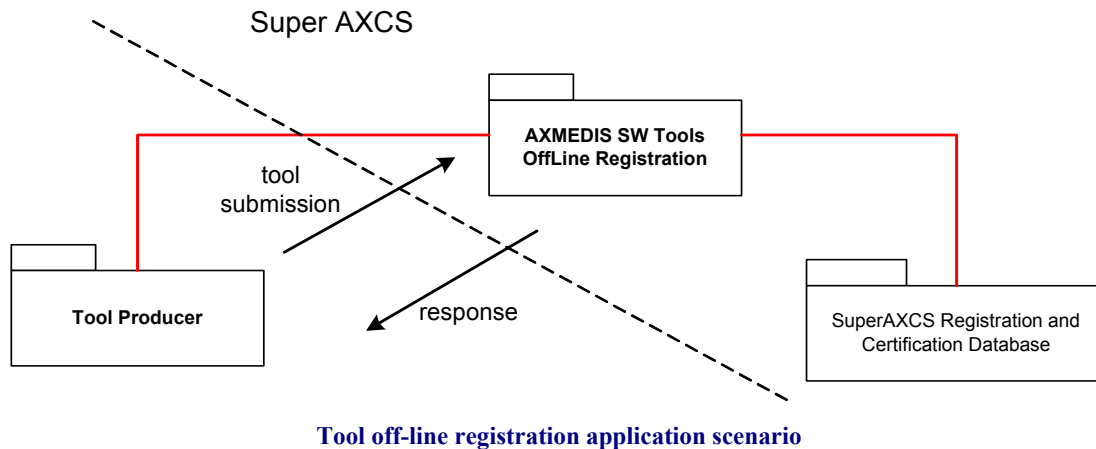
3.1 AXMEDIS Software Tool Off Line Registration (DSI, EXITECH)

| Module Profile | | |
|--|-----------------------------------|--|
| AXMEDIS Software Tool Off Line Registration | | |
| Executable or Library(Support) | Web application | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Tool Producer Area | Web service based on (TBD) | Protected (SSL) |
| SuperAXCS:Registration and Certification Database | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| Web interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

The Software Tool Off-line Registration is a web application oriented to AXMEDIS Tool Producers. Its role is to allow tool producers the submission of their software to become certified AXMEDIS Tools. Tool checking activity is not part of this module: it is made in a second time, after receiving the aspirant tool. It has to remember that a candidate tool has to be checked and tested to verify if it accomplishes AXMEDIS guidelines. The Software Tool Off-line Registration module has to supply also an interface intended to collect data about tool and to insert it in Super AXCS object database.

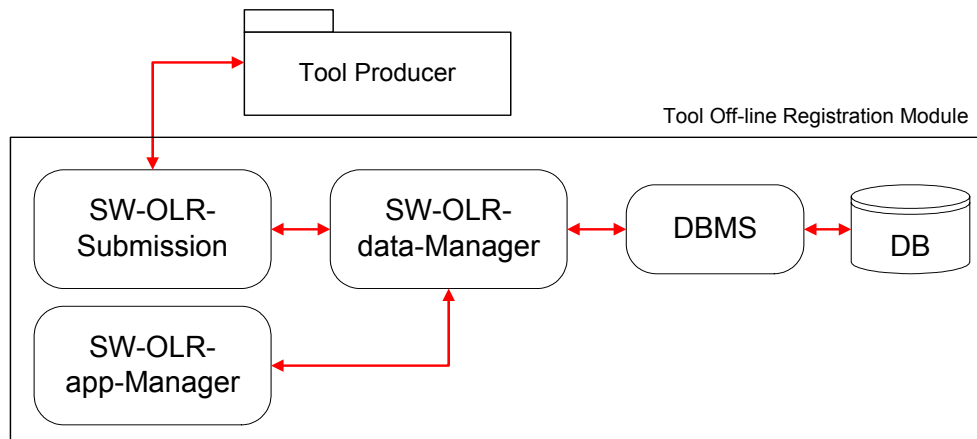
3.1.1 AXMEDIS Software Tool Off-line Registration logic

AXMEDIS tool off-line registration is a web application intended to provide services to tool producer in order to give them a way to submit their candidate tools. It gives also a way to Super AXCS manager to integrate information about candidate tools.



This module has to provide the following functions:

- Receive the software tool candidate from tool producers
- Provide a web interface to SuperAXCS Manager in order to manage the software reception, and to complete data about the received software
- Access SuperAXCS database to manage (register and retrieve data) about candidate tools.

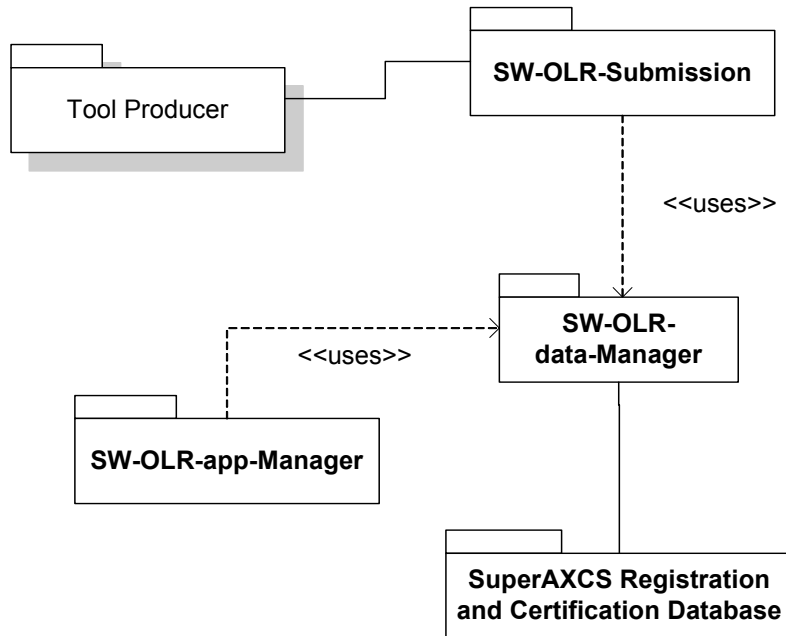


To perform its own task it can be found the following logical decomposition:

Sw-OLR-submission: it realizes the web interface needed to accept and receive the software tool candidate. The received software is temporarily stored on the web server to be retrieved for the necessarily tests. It also place a web interface used by software tool producer to give some more information about the provided candidate.

Sw-OLR-app-manager: It is only a web interface needed by the Super AXCS manager to complete data about the received software, accept or discharge it and manage the whole Software Tool Off-line Registration process.

Sw-OLR-data-manager: It is a module intended to provide the above modules some method to access database (retrieve and store data).



The following table describes methods thought to be used in AXMEDIS Tool Off-line Registration.

| Method name and parameters | Output | Description |
|--|---------------------|---|
| scope: public | | |
| SW-OLR-Submission::ReceiveTool (string NickName, string Password, toolinfoType ToolInfoData) | String ResultStatus | It is a public method. It collects requestor credentials needful to access the system and uses the VerifyLogin() (a SW-OLR-DataManager method) to verify requestor credentials. It also collects tool relate data (in a field called ToolInfoData typed toolinfoType), according to TypeofTool table in Registration and certification database. It also receives the candidate software tool and store it in a folder on the web server. This method uses private methods described below to perform its own tasks. The ResultStatus output parameter is set to 0 if the task is successful and is set to 1 otherwise. |
| SW-OLR-app-Manager::Management (string NickName, string Password) | String ResultStatus | It is a public method. In implements the application management. It collects module manager credentials needful to access the Application Manager and uses the VerifyLogin() |

| | | |
|---|---|---|
| | | <p>(a SW-OLR-DataManager method) to verify user credentials. This module provides also a web interface (a panel) allowing SuperAXCS manager to perform the following tasks:</p> <ol style="list-style-type: none"> candidate tool related data management (edit and complete) acceptation or rejection of candidate software tool sending a message to tool producer about the outcome of candidate tool analysis (using the SW-OLR-app-Manager::Response method) type of tool fingerprint calculation (using the SW-OLR-app-Manager::FingerprintCalc method) |
| scope: private | | |
| SW-OLR-app-Manager::Response (String ToolProducer-email, string message) | String ResultStatus | It sends a message to software tool candidate producer related to the result of software candidate analysis. The ResultStatus output parameter is set to 0 if the task is successful and is set to 1 otherwise. |
| SW-OLR-app-Manager::FingerprintCalc (String ToolPath) | String ResultStatus String ToolFingerprint | It calculates the software tool fingerprint and return it in the ToolFingerprint field to be used by the calling method. The ResultStatus output parameter is set to 0 if the task is successful and is set to 1 otherwise. |
| CollectorTransferManager::EncryptComm (string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| CollectorTransferManager::DecryptComm (string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| SW-OLR-DataManager::Data-Retriever(string Query) (the needful input data can be stored in the class attributes or sent as method parameters) | String ResultQuery | This method retrieve data from database according to its input values. The ResultQuery field returns query response data |
| SW-OLR-DataManager::Data-Insertier() (the needful input data can be stored in the class attributes or sent as method parameters) | String ResultQuery | This method inserts data into database according to its input values. The ResultQuery field returns a value to indicate if operation was successful |
| SW-OLR-DataManager::Data-Modifier() (the needful input data can be stored in the class attributes or sent as method parameters) | String ResultQuery | This method modifies data into database according to its input values. The ResultQuery field returns a value to indicate if operation was successful |
| SW-OLR-DataManager::VerifyLogin (string NickName, string Password) | boolean IsTrusted | It accesses Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process. |

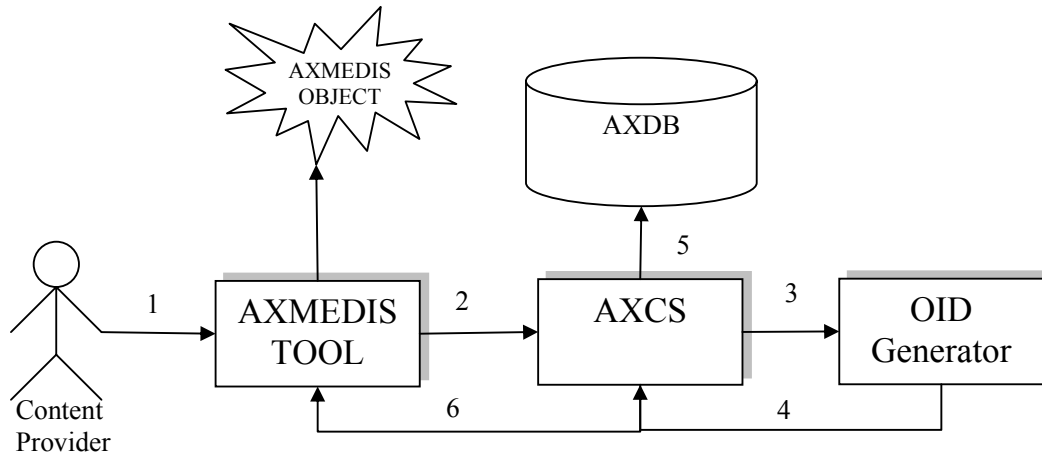
3.2 AXMEDIS OID Generator (WP5.6.2: DSI, EXITECH)

| Module Profile | | |
|----------------------------------|-----------------------------------|--|
| AXMEDIS OID Generator | | |
| Executable or Library(Support) | Web application | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXMEDIS Protection tool Area:PMS | Web service based on (TBD) | Protected (SSL) |
| SuperAXCS:Database Interface | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No user interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

When a new object ID is requested all its information is provided and used to fill the database, then the AXOID has to provide two main services to the system:

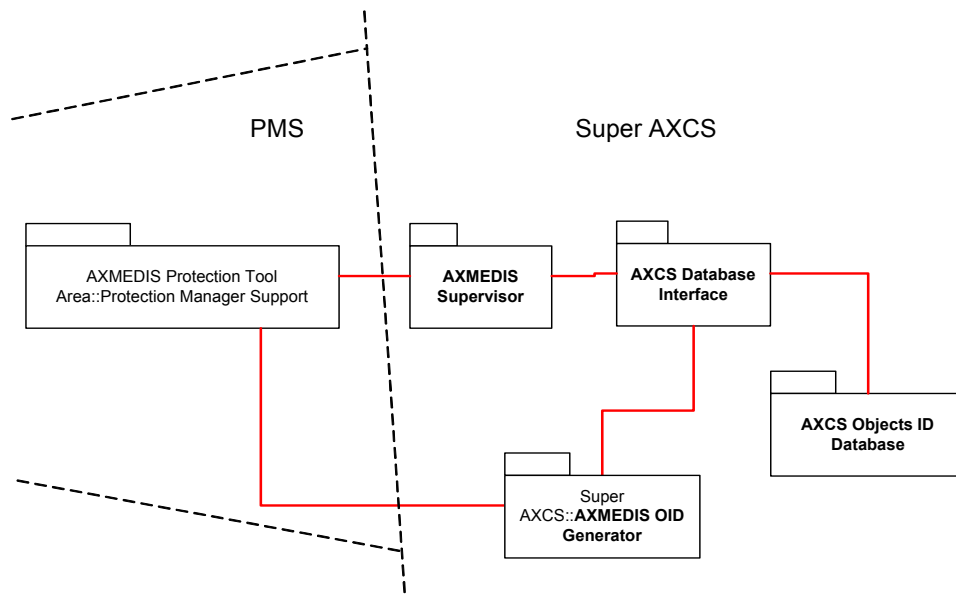
1. Univocal generation of object ID. It is important to find a way to produce not duplicated ID for the object to univocally identify every object in the system
2. Collecting all the related object information the producer has to provide to the system (as metadata, producer information and so on)

3.2.1 AXMEDIS OID Generator scenario



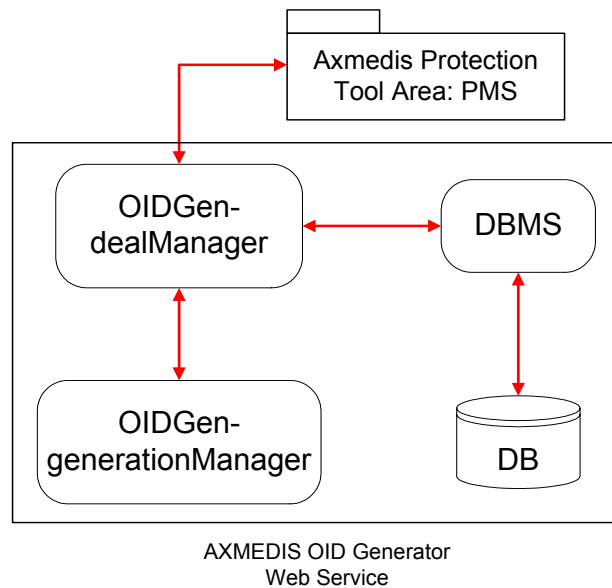
1. A Creator, Distributor or a Content Provider, in more general sense, wants to create a new AXMEDIS Object.
2. The tool with that is creating the object requests to the AXCS an object ID
3. AXCS interact with Object ID generator to ask an ID
4. OID generator returns back to AXCS the generated ID
5. AXCS stores ID and related info in the AXDB
6. Generated ID is communicated to the tool
7. A new AXMEDIS object is created with the assigned ID

3.2.2 AXMEDIS OID Generator logic



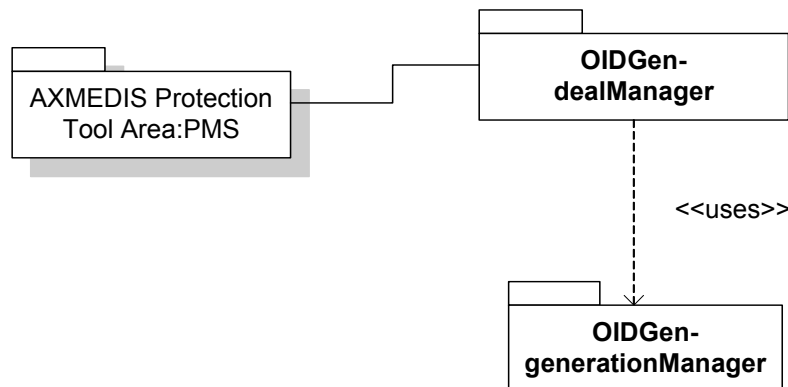
The AXMEDIS OID Generator is a module which task is the generation of Objects ID and the collection of Objects related data (as metadata and others). To access database it uses the AXDB-API (plase see the AXMEDIS Database Interface section). In particular it has to provide the following functions:

- Generate the OID using a UUID standard algorithm
- Communicate with PMS in order to fournishing it the generated OID
- Collect object related data in order to store it in Object ID database
- Store the collected data in the Object ID database



To perform its own task it can be found the following decomposition:

- **OIDGen-generationManager**: it has to supply the generation of ID. To perform its task an UUID standard algorithm is used.
- **OIDGen-dealManager**: It manages the communication with other modules to receiving and providing information. It manages also communication with AXCS database interface in order to access Object ID database.



The following table describes methods thought to be used in AXMEDIS OID Generation web service:

| Method name and parameters | Output | Description |
|---|---------------------------------------|--|
| scope: public | | |
| OIDGen-dealManager::Registration(string NickName, string Password, ObjectDataType ObjectData) | string result string UUID-AXOID | It is the only public method of this web service. It collects requestor credentials needful to access the system and uses the VerifyLogin() (an OIDGen-dealManager method) to verify requestor credentials. It also collects object data (ObjectData) provided by the requestor and uses the AXDB-API methods provided by AXCS Database Interface to insert received information into database and to provide the result to the requestor. The result output parameter is set to 0 if the registration is successful otherwise is set to 1. This method returns also the produced AXOID in UUID-AXOID field. |
| scope: private | | |
| OIDGen-dealManager::EncryptComm(string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| OIDGen-dealManager::DecryptComm(string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| OIDGen-dealManager::VerifyLogin(string NickName, string Password) | boolean IsTrusted string PublicKey | It accesses Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process. It also retrieves the requestor public key stored in the DB. Note: the password is encrypted both the one received as input from the method and the one stored in DB. The equality check is made between encrypted strings. |
| OIDGen-generationManager::UUIDGenerator(string GenInput) | String AXOID | It provides the object ID (AXOID) according to the UUID algorithm. More information can be found in the pertinent section. |

3.2.3 AXMEDIS OID Generator algorithm

In order to generate an object ID that is unique in the whole system, it has been considered to use a standard algorithm: the UUID generator algorithm.

UUID is an identifier that is unique across both space and time¹, with respect to the space of all UUIDs. A UUID can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects across a network. The generation of UUIDs does not require a registration authority for each single identifier. Instead, it requires a unique value over space for each UUID generator. This spatially unique value is specified as an IEEE 802 address, which is usually already applied to network-connected systems. This 48-bit address can be assigned based on an address block obtained through the IEEE registration authority. This UUID specification assumes the availability of an IEEE 802 address. The UUID consists of a record of 16 octets and must not contain padding between fields. The total size is 128 bits.

More information about UUID generation can be found at the following web address:

<http://www.opengroup.org/onlinepubs/9629399/apdxa.htm>

3.2.4 AXMEDIS OID Generator web service interface formalization

| | |
|------------------------------|--|
| WSDL | <pre> <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:rm="http://new.webservice.namespace" targetNamespace="http://new.webservice.namespace"> <types> <xs:schema targetNamespace="http://new.webservice.namespace" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="ObjectDataType"> <sequence> <!-- data according to database structure: info to be provided concerns objects, related metadata and pertinent relations ... --> </sequence> </complexType> </xs:schema> </types> <message name="OIDGen-Request"> <part name="NickName" type="xs:string"/> <part name="Password" type="xs:string"/> <part name="ObjectData" type="rm:ObjectDataType"/> </message> <message name="OIDGen-Response"> <part name="Result" type="xs:string"/> <part name="UUID-AXOID" type="xs:string"/> </message> <portType name="OIDGen-PortType"> <operation name="OIDGen"> <input message="rm:OIDGen-Request"/> <output message="rm:OIDGen-Response"/> </operation> </portType> <binding name="OIDGeneration" type="rm:OIDGen-PortType"> <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="OIDGen"> <soap:operation soapAction="urn:# OIDGeneration"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <soap:operation soapAction="urn:# OIDGeneration"/> </operation> </binding> <service name="OIDGeneratorWebService"> <port name="OIDGeneration" binding="rm:OIDGeneration"> <soap:address location="No Target Adress"/> </port> </service> </definitions> </pre> |
| Request Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:OIDGen xmlns:m="http://new.webservice.namespace"> <NickName>String</NickName> <Password>String</Password> <ObjectData/> <!-- data according to database structure: info to be provided concerns objects, related metadata and pertinent relations ... --> <ObjectData/> </m:OIDGen> </SOAP-ENV:Body> </pre> |

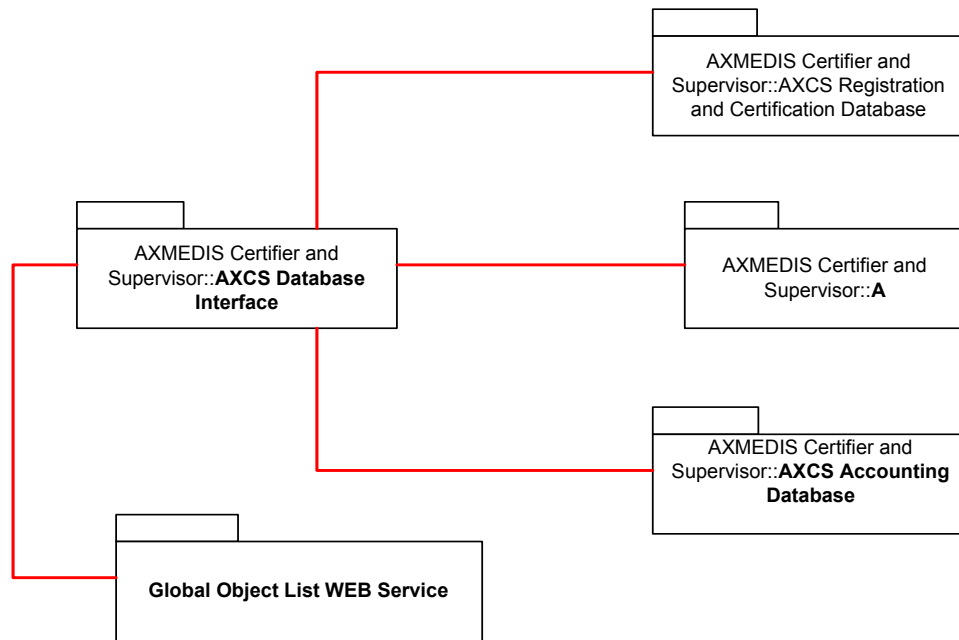
| | |
|-------------------------------|---|
| Response Sample Message | <pre> </SOAP-ENV:Envelope> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:OIDGen xmlns:m="http://new.webservice.namespace"> <Result>0</Result> <UUID-AXOID>0A2Z4X678B0124C56X89W123452CV478</UUID-AXOID> </m:OIDGen> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
|-------------------------------|---|

3.3 Global Object List WEB Service (DSI, EXITECH)

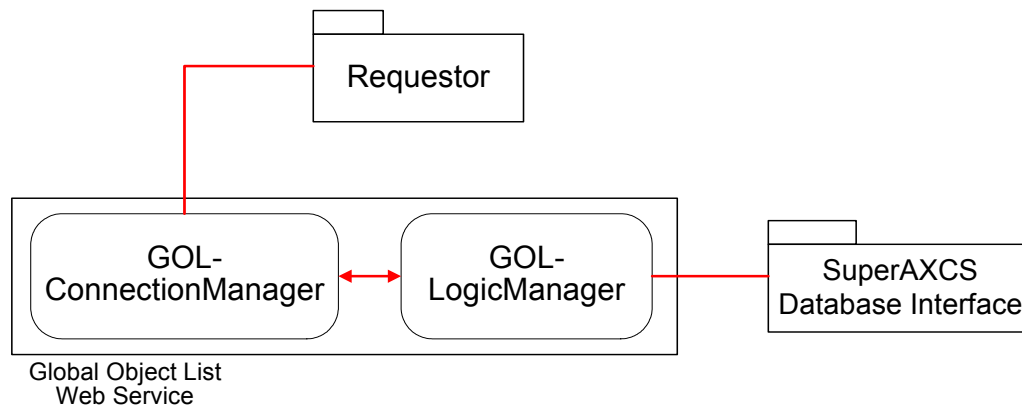
| Module Profile | | |
|--|-----------------------------------|--|
| Global Object List WEB Service | | |
| Executable or Library(Support) | Web application | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| SuperAXCS:Database interface | Web service based on (TBD) | Protected (SSL) |
| Any authorized requestor (Distributors, Integrators, Creators and so on) | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| Web interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

The Global Object List web service is intended to provide a service that has to give information about all the objects present in the system. It can be queried by Distributors, Creators, Integrators, and so on. This web service could also be used as background to web application implementing searching and browsing of objects in the system, either a global catalog or a partial catalog (as for example a single Object Distributor catalog).

3.3.1 Global Object List logic



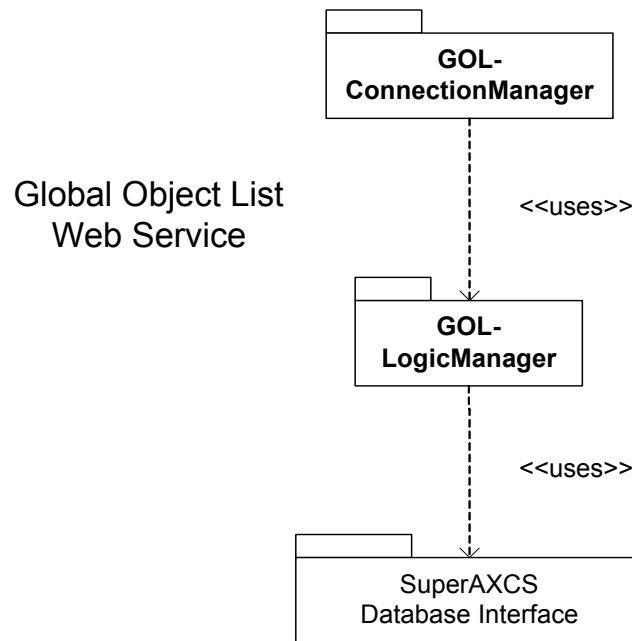
The Global Object List Web Service is a web application running on a web server implemented as a set of scripts. It has to provide to the system the possibility of retrieving information about all the objects present in the system. Given some data (typically a set of metadata) a requestor has to be able to recover all references about objects that are compliant with the given data.



It can be identified the following logical decomposition:

GOL-ConnectionManager: this component manages the requests received by web service, collects data requests and response to requestors providing information compliant with requests.

GOL-LogicManager: this component realizes the whole application logic and uses AXCS Database Interface (and the related AXDB-API) to access database.



The following table describes methods thought to be used in Global Object List Web Service:

| Method name and parameters | Output | Description |
|---|--|---|
| scope: public | | |
| GOL-ConnectionManager::AcceptRequest (string NickName, string Password, string RequestQuery) | String ResultStatus String ResponseData | It is a public method. It collects requestor credentials needful to access the system and uses the VerifyLogin() (a GOL-LogicManager method) to verify requestor credentials. It also collects queries data (in a field called RequestQuery) . This method uses private methods described below to perform its own tasks. The ResultStatus output parameter is set to 0 if the task is successful and is set to 1 otherwise. It returns query results (compliant with data request) in a field called ResponseData. String data has been choose to have more flexibility in input and output parameters. |
| scope: private | | |
| GOL-ConnectionManager::EncryptComm (string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| GOL-ConnectionManager::DecryptComm (string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| GOL-LogicManager::Data-Retriever(string Query) | String ResultQuery | This method retrieves data from database (using SuperAXCS |

| | | |
|---|-------------------|--|
| (the needful input data can be stored in the class attributes or sent as method parameters) | | Database Interface API) according to its input values. The ResultQuery field returns query response data. |
| GOL-LogicManager::VerifyLogin (string NickName, string Password) | boolean IsTrusted | It accesses Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process. |

3.3.2 Global Object List WEB Service interface formalization

| | |
|------------------------|--|
| WSDL | <pre> <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:rm="http://new.webservice.namespace" targetNamespace="http://new.webservice.namespace"> <types> <xs:schema targetNamespace="http://new.webservice.namespace" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> </xs:schema> </types> <message name="GOL-Request"> <part name="NickName" type="xs:string"/> <part name="Password" type="xs:string"/> <part name="RequestQuery" type="xs:string"/> </message> <message name="GOL-Response"> <part name="ResultStatus" type="xs:string"/> <part name="ResponseData" type="xs:string"/> </message> <portType name="GOL-PortType"> <operation name="GOL"> <input message="rm:GOL-Request"/> <output message="rm:GOL-Response"/> </operation> </portType> <binding name="GOLList" type="rm:GOL-PortType"> <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="GOL"> <soap:operation soapAction="urn:# GOL"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <soap:operation soapAction="urn:# GOL"/> </operation> </binding> <service name="GOLWebService"> <port name="GOL-port" binding="rm:GOLList"> <soap:address location="No Target Adress"/> </port> </service> </definitions> </pre> |
| Request Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:GOL xmlns:m="http://new.webservice.namespace"> <NickName>... user name ...</NickName> <Password>... relatd password ...</Password> <RequestQuery>DCCCreatorsMetadata.CreatorValue="Mozart" and DublinCore.Language=IT </pre> |

| | |
|-------------------------------|---|
| | <pre> </RequestQuery> </m:GOL> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
| Response Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:GOL xmlns:m="http://new.webservice.namespace"> <ResultStatus>0</ ResultStatus > <ResponseData>... response string ...</ ResponseData > </m:GOL> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |

3.4 Active AXCS List/Database (DSI, EXITECH)

3.4.1 Entity-Relationship description

In order to define the Active AXCS List Database schema, first have to identify the entities and related relations. Here is reported the list of the identified entities and the related meaning.

AXCSs: this entity contains data about AXMEDIS AXCSs registered to SuperAXCS.

AXCSs

Number of indexes: ?
Number of foreign keys: ?

| Columns | | idx | Data type | Allow NULLs | Value/Range |
|-------------|----|-----|----------------------|-------------|-------------------------|
| AXCSID | PK | I | N-Signed Integer | Not allowed | |
| AXDOM | FK | I | N-Signed Integer | Allowed | |
| Email | | | C-Large Length | Not allowed | |
| NickName | | | C-Large Length | Not allowed | |
| Password | | | C-Large Length | Not allowed | |
| Description | | | C-Large Length | Not allowed | |
| Location | | | C-Large Length | Not allowed | |
| Nationality | | I | C-Large Length | Not allowed | |
| PubKey | | | C-Large Length | Not allowed | |
| RegDate | | | T-Date & Time | Not allowed | |
| RegDeadline | | | T-Date & Time | Not allowed | |
| Status | | | C-Variable Length(1) | Allowed | B/U (Blocked/Unblocked) |

Column details

1. AXCSID (PK)

Physical data type: LONG
Allow NULLs: Not allowed
Notes: ID of the AXCS

2. AXDOM (FK)

Physical data type: LONG
Allow NULLs: Allowed
Notes: AXMEDIS Current Domain of the AXCS (if any)

3. Email

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: email of the related AXCS manager

4. NickName

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: NickName of the AXCS manager

5. Password

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: MD5 or other encryption of AXCS manager password

6. Description

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Description of the AXCS

7. Locations

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Location of the AXCS

8. Nationality

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Nationality of the AXCS

9. PubKey

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: public key of the AXCS

10. RegDate

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of AXCS registration

11. RegDeadline

Physical data type: DATETIME
Allow NULLs: Not allowed
Notes: Timestamp of AXCS registration end

12. Status

Physical data type: VARCHAR(1)
Allow NULLs: Not Allowed
Notes: Status of the AXCS: B/U (Blocked/Unblocked)

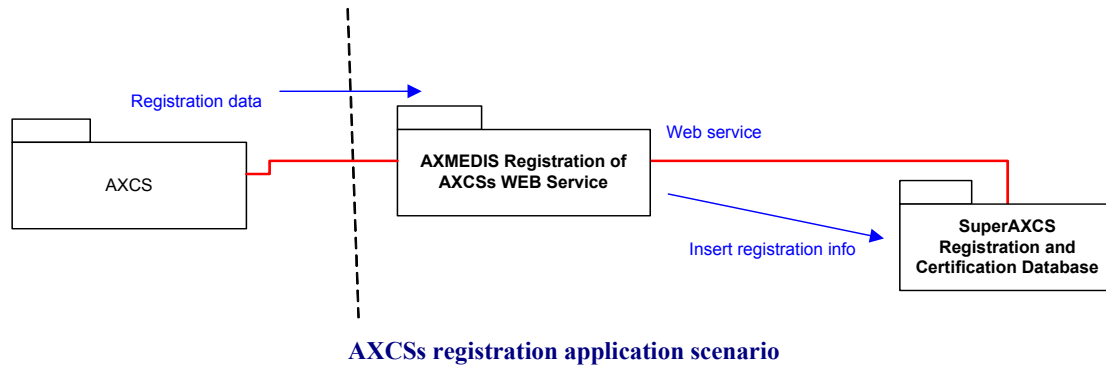
3.5 AXMEDIS Registration of AXCSs (DSI, EXITECH)

| Module Profile | | |
|--------------------------------|-----------------------------------|--|
| AXMEDIS Registration of AXCSs | | |
| Executable or Library(Support) | Web services | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXCS | Web service based on (TBD) | Protected (SSL) |
| SuperAXCS Database Interface | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

All AXMEDIS AXCS must be registered by SuerAXCS. The AXMEDIS Registration Web Services is the AXCS module that receives data about AXCS and store it in the AXCS Registration and Certification Database. Once the AXCS is registered in the system and the related data is stored in the SuperAXCS Registration and Certification Database, other SuperAXCS modules can access this database to retrieve some information needful to perform their work.

3.5.1 Application Scenario for AXCSs registration

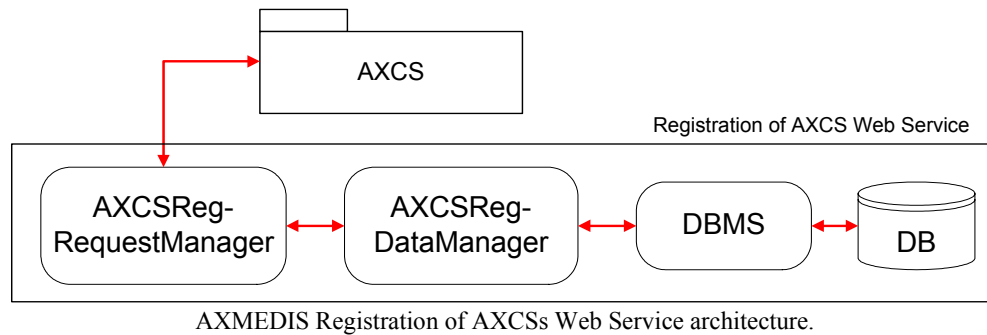
The scenario shows the relationship between the AXMEDIS Registration of AXCSs module and other modules inside SuperAXCS.



3.5.2 AXMEDIS Registration of AXCSs logic

The AXMEDIS Registration of AXCS Web Service is a web application running on a web server, implemented as a set of web scripts. We can identify the following logical decomposition:

- **AXCSReg-RequestManager**: this component receives registration requests from AXCS, and prepares them to be processed by the AXCSReg-DataManager. The communication channel connecting AXCS and AXCSReg-RequestManager is protected using a secure protocol (for instance SSL). It implements the interface with requestors and manages the whole application.
- **AXCSReg-DataManager**: this component receives data from AXCSReg-RequestManager and inserts it in the SuperAXCS Registration and Certification Database. If necessary elaborates and fits data before insert it in the database. The database management is performed using the related API (AXDB-API).

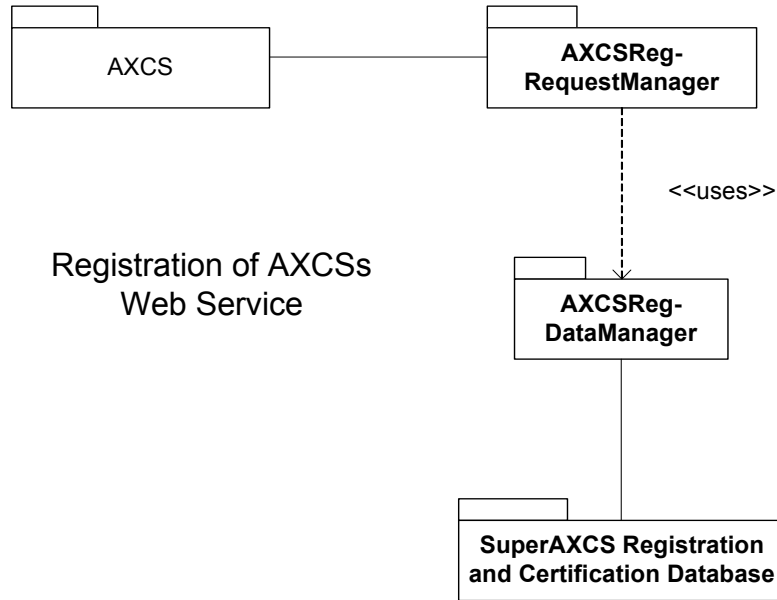


The **AXCSReg-RequestManager** should be composed by some modules implementing the following functions:

- Accepting authentication data, needful to verify requestor credentials and make it access the system
- Accepting AXCS registration data and preparing it to be transferred to AXCSReg-DataManager
- Reply to requestor with the most appropriate message (on the basis of AXCSReg-DataManager responses)

The **AXCSReg-DataManager** should be composed by some modules implementing the following functions:

- Receiving verification requests and data from the AXCSReg-RequestManager
- Accessing the database (using the AXDB-API) to verify requestors credential information
- Accessing the database to store received AXCSs registration data
- Reply to AXCSReg-RequestManager according to the performed actions



The following table describes methods thought to be used in AXMEDIS Registration Web Service.

| Method name and parameters | Output | Description |
|--|--|--|
| scope: public | | |
| AXCSReg-RequestManager::Registration(string NickName, string Password, AXCSDataType regdata) | string result string definitive-AXCSID | It is the only public method of this web service. It collects requestor credentials needful to access the system and uses the VerifyLogin() (an AXCSReg-DataManager method) to verify requestor credentials. It collects also registration data (regdata) provided by the requestor and uses the other methods (described below) to insert them in database and to provide the result to the requestor. The result output parameter is set to 0 if the registration is successful otherwise is set to 1. This method returns also the definitive AXCSID in definitive-AXCSID |
| scope: private | | |
| AXCSReg-RequestManager::DataPrepare() | All user registration fields. See below for details. | It prepares the AXCS registration data received by the web service to be stored in DB. Is also uses IDGenerator() to get the definitive AXCSID to be inserted in the database and to be sent to requestor |
| AXCSReg-AXCSReg-RequestManager::IDGenerator(string TempUID) | String DefinitiveAXCSID | It provides the definitive AXCSID. |
| AXCSReg-RequestManager::EncryptComm(string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| AXCSReg-RequestManager::DecryptComm(string | String ClearData | Decrypt EncodedData with provided asymmetric key |

| | | |
|---|---------------------------------------|---|
| String EncodedData, string asimKey) | | (asymKey) |
| AXCSReg-DataManager::VerifyLogin(string NickName, string Password) | boolean IsTrusted string PublicKey | It accesses Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process. It also retrieves the requestor public key stored in the DB. Note: the password is encrypted both the one received as input from the method and the one stored in DB. The equality check is made between encrypted strings. |
| AXCSReg-DataManager::StoreData() (the needful input data can be stored in the class attributes or sent as method parameters) | | Store received data in the Registration and Certification Database |

Note that it has been introduced two encrypting/decrypting methods (DecryptComm, EncryptComm) to enforce the encrypting robustness. In fact we can suppose to use an encrypted protocol (like SSL), but we can enforce encryption robustness (and therefore security) encrypting our self data too using a Public/Private key paradigm. You have to remember that public keys are stored in Registration and Certification Database.

3.5.3 AXMEDIS Registration of AXCSs WEB Service interface formalization

In the present paragraph is explained the Registration of AXCS Web Service interface using the WSDL formalism.

| | |
|------|---|
| WSDL | <pre> <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:rm="http://new.webservice.namespace" targetNamespace="http://new.webservice.namespace"> <types> <xs:schema targetNamespace="http://new.webservice.namespace" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="usrDatatype"> <sequence> <!-- data according to database structure ... --> </sequence> </complexType> </xs:schema> </types> <message name="RegistrationRequest"> <part name="NickName" type="xs:string"/> <part name="Password" type="xs:string"/> <part name="Regdata" type="rm:usrDatatype"/> </message> <message name="RegistrationResponse"> <part name="Result" type="xs:string"/> <part name="definitive-AXCSUID" element="" type="xs:string"/> </message> <portType name="Registration_PortType"> <operation name="Registration"> <input message="rm:RegistrationRequest"/> <output message="rm:RegistrationResponse"/> </operation> </portType> <binding name="Registration" type="rm:Registration_PortType"> </pre> |
|------|---|

| | |
|-------------------------------|--|
| | <pre> <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="Registration"> <soap:operation soapAction="urn:#Registration"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <soap:operation soapAction="urn:#Registration"/> </operation> </binding> <service name="RegistrationWebService"> <port name="Registration" binding="rm:Registration"> <soap:address location="No Target Address"/> </port> </service> </definitions> </pre> |
| Request Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Registration xmlns:m="http://new.webservice.namespace"> <NickName>...distributor user...</NickName> <Password>...password...</Password> <Regdata> <!-- data according to database structure ... --> </Regdata> </m:Registration> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
| Response Sample Message | <pre> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <SOAP-ENV:Body> <m:Registration xmlns:m="http://new.webservice.namespace"> <Result>0</Result> <definitive-AXCSID>0A2Z4X678B0124C56X89W123452CV478</definitive-AXCSID> </m:Registration> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |

3.6 AXCS Collector (DSI, EXITECH)

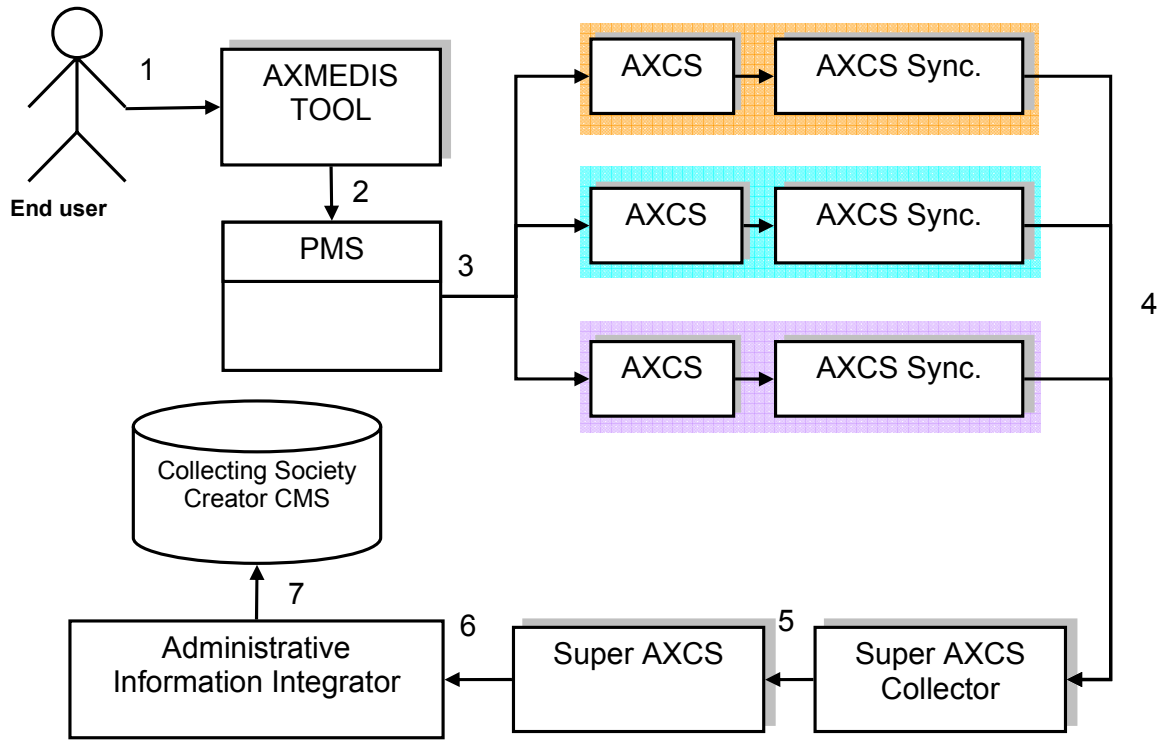
| Module Profile | | |
|--|-----------------------------------|--|
| AXCS Collector | | |
| Executable or Library(Support) | Web service or executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | Chellini | |
| Responsible Partner | DSI | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXCS:Synchronizer | TBD | Protected (SSL) |
| SuperAXCS:AXCSs Active List Database | | |
| AXCS:Accounting Database | | |
| AXCS:Registration and Certification Database | | |
| AXCS:Object ID Database | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| No user interface | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

SuperAXCS and AXCS database synchronization is a task performed by a couple of components: the AXCS Synchronizer and the SuperAXCS Collector. The interaction and collaboration between these two components is very important. To better understand the SuperAXCS Collector role and logic, please take a look to the AXCS Synchronizer section.

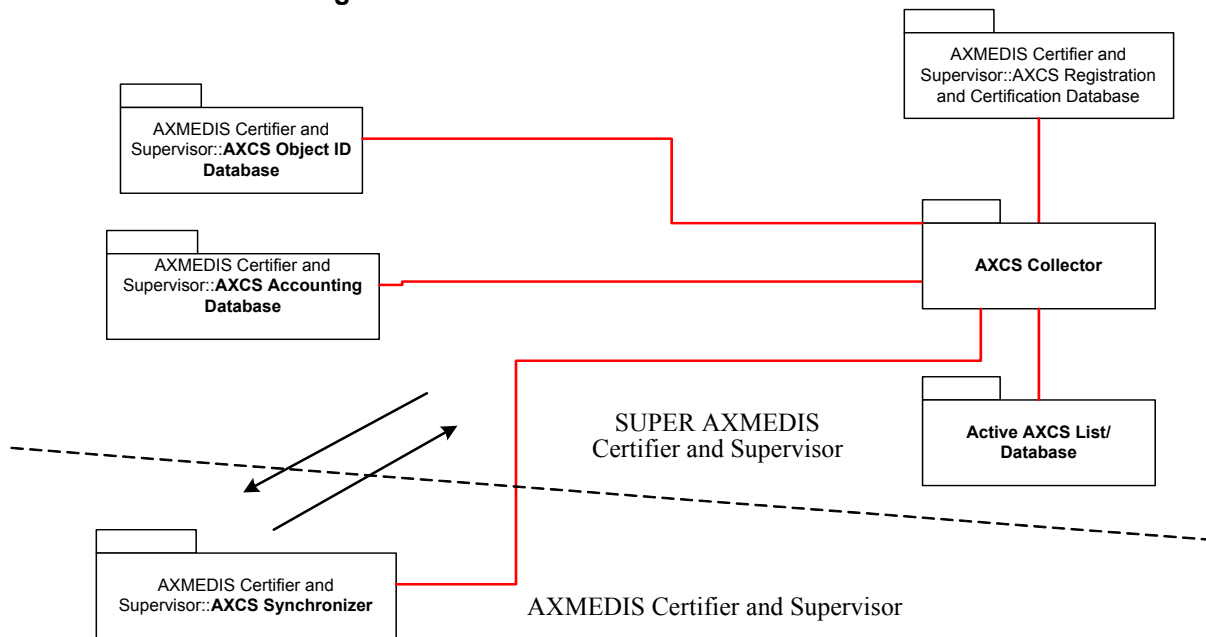
3.6.1 AXCS Collector scenario

1. End user uses an AXMEDIS tool to operate on an AXMEDIS Protected Objects that are on different distribution channels
2. Protection Manager Support allow only authorized operations on the object
3. Objects are accessed on different channels and each AXCS stores its Action-Logs
4. Via the AXCS sync general information on objects or information that allow SuperAXCS to recover Action-Logs from the different AXCSs are transferred to the SuperAXCS Collector

5. SuperAXCS collects information
6. Administrative reports are created
7. Administrative Information Integrator transfer Action-Logs on CMS



3.6.2 AXCS Collector logic



Super AXCS Collector is the counterpart of AXCS Synchronizer. AXCS Synchronizer is an AXCS module used to synchronize AXCSs and SuperAXCS database. Every AXCS Database entry has to be retrievable by

SuperAXCS in order to be able to provide to requestors all the information contained in each AXCS database.

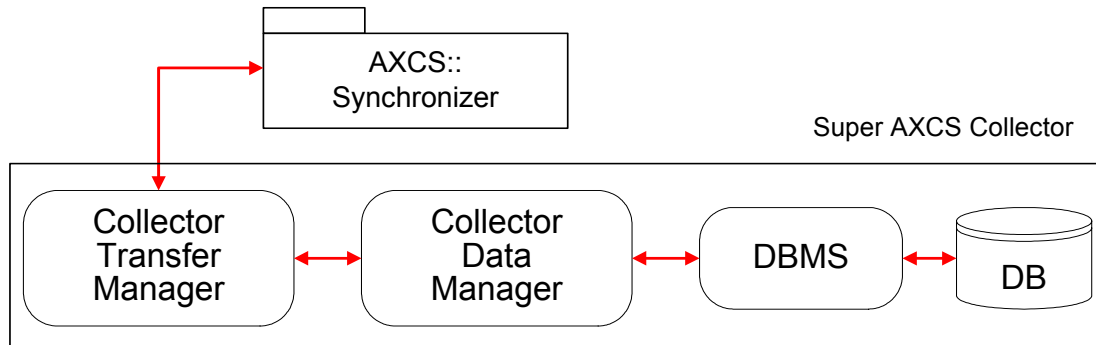
To perform their own tasks, several architectures are possible:

1. Hierarchical Network
2. Peer Network
3. Blend Network

These possible architectures have been discussed in the AXCS Collector section.

The SuperAXCS Collector has to perform the following tasks:

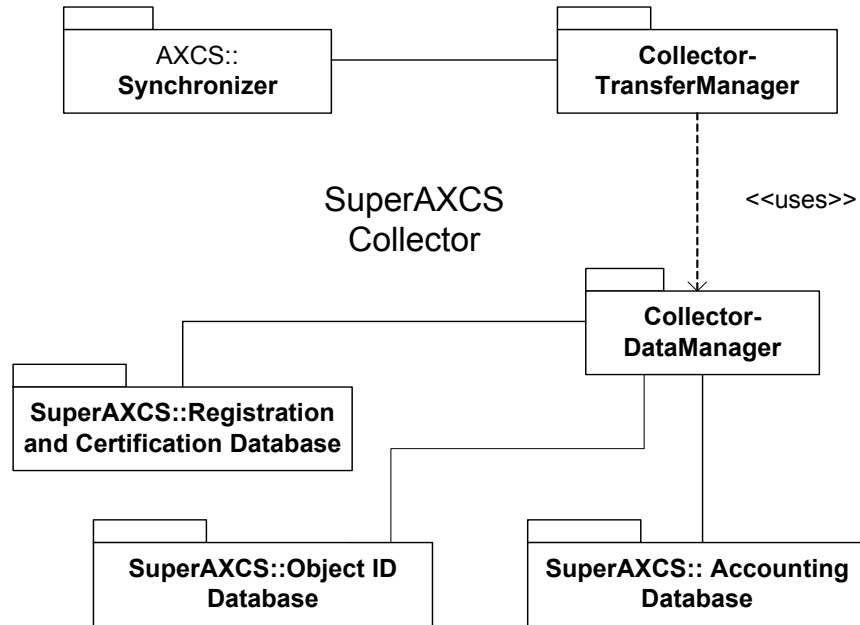
- receive data from AXCSs Synchronizer (requesting it or registering their posts)
- insert the received (or requested data) into database



In order to perform its own task SuperAXCS Collector can be decomposed in the following modules:

CollectorTransferManager: it manages the connection and deals with AXCS Synchronizer in order to receive the pertinent data. The communication channel connecting SuperAXCS Collector and AXCS Synchronizer is protected using a secure protocol (for instance SSL). The communication can happen with Synchronizer as client or server (and the Collector as a respective counterpart). To perform both paradigms have been considered a public method both in Synchronizer and in Collector.

CollectorDataManager: it collects all data received from AXCS Synchronizer and inserts it in database. It could be thought as a composition of AXCS components that perform single tasks in order to maximize the ideas and code reuse. For instance we can think to reuse the AXMEDIS Registration WEB Service to collect data about AXMEDIS Users, AXCS OID Generator to collect data about AXMEDIS Objects, and so on.



The following table describes methods thought to be used in Super AXCS Collector.

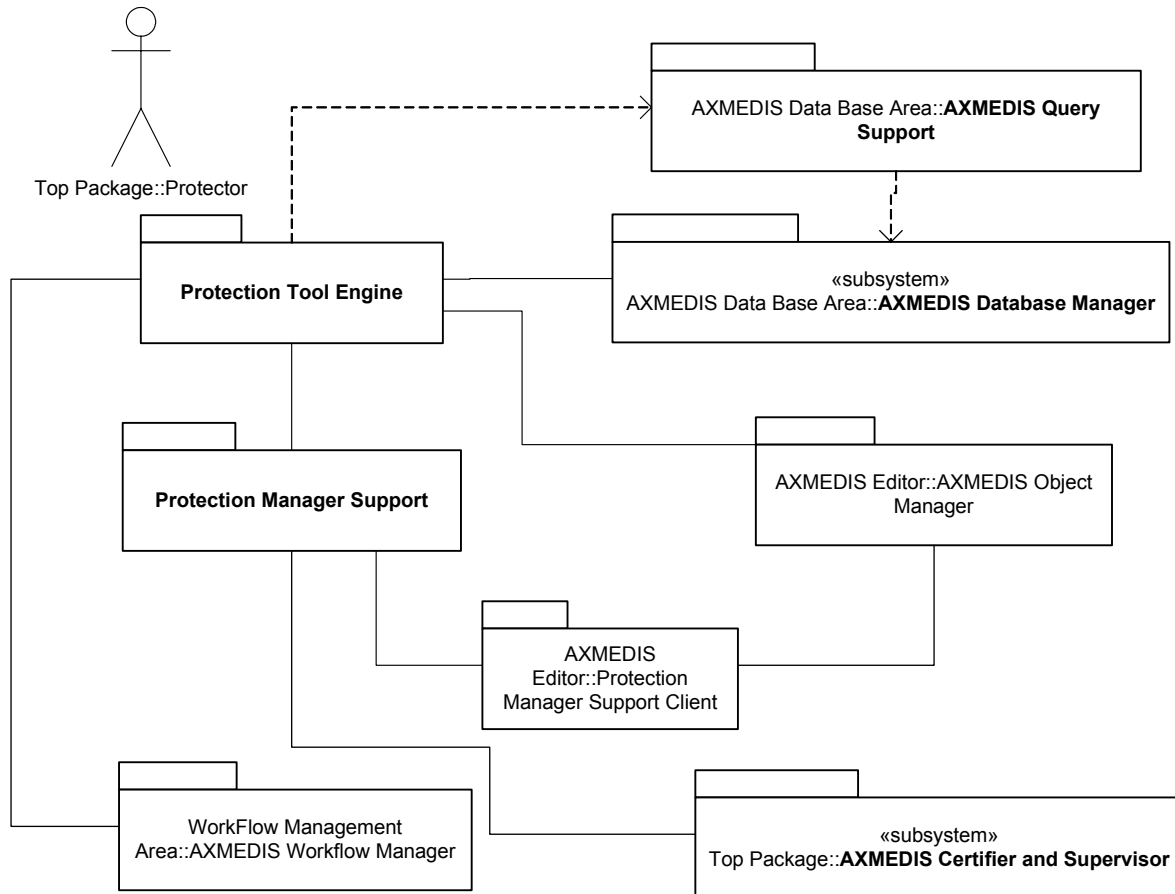
| Method name and parameters | Output | Description |
|---|----------------------|---|
| scope: public | | |
| CollectorTransferManager::Synchronize(string NickName, string Password, syncDataType TransferData) | String ResultStatus | It is the only public method of this module. It collects AXCSs credentials needful to access the system and uses the VerifyLogin() (a CollectorDataManager method) to verify requestor credentials. This method uses private methods described below to perform its own tasks and to answer to requestor. The ResultStatus output parameter is set to 0 if the task is successful and is set to 1 otherwise. This method receives (as input parameter) also data to be transferred from the requestor in the TransferData field (typed syncDataType). |
| scope: private | | |
| CollectorTransferManager::EncryptComm (string ClearData, string asimKey) | String EncodedData | Encrypt ClearData with provided asymmetric key (asymKey) |
| CollectorTransferManager::DecryptComm (string String EncodedData, string asimKey) | String ClearData | Decrypt EncodedData with provided asymmetric key (asymKey) |
| CollectorDataManager::DataCollect(syncDataType TransferData) (the needful input data can otherwise be stored in the class attributes or received as method parameters) | String ResultStausts | It collects the received data (using the TranferData field) and inserts it in database. The ResultStatus output parameter is set to 0 if the task is successful and is set to 1 otherwise. |
| CollectorDataManager::VerifyLogin (string NickName, string Password) | boolean IsTrusted | It accesses SuperAXCS Registration and Certification Database to verify if the couple NickName and Password is the same provided through the registration process Note: the password is encrypted both the one received as input from the method and the one stored in DB. The equality check is made between encrypted strings. |

4 Protection Tool area (WP4.5: FUPF, EXITECH, WP5.6.5: FHGIGD)

The following figure shows the general structure of the AXMEDIS Protection Tool Area. The main components of this area are the Protection Tool engine and the Protection Manager support. They make use of (or are used by) several other modules inside the AXMEDIS project.

In the next sections, these tools are described in detail.

AXMEDIS Protection Tool Area

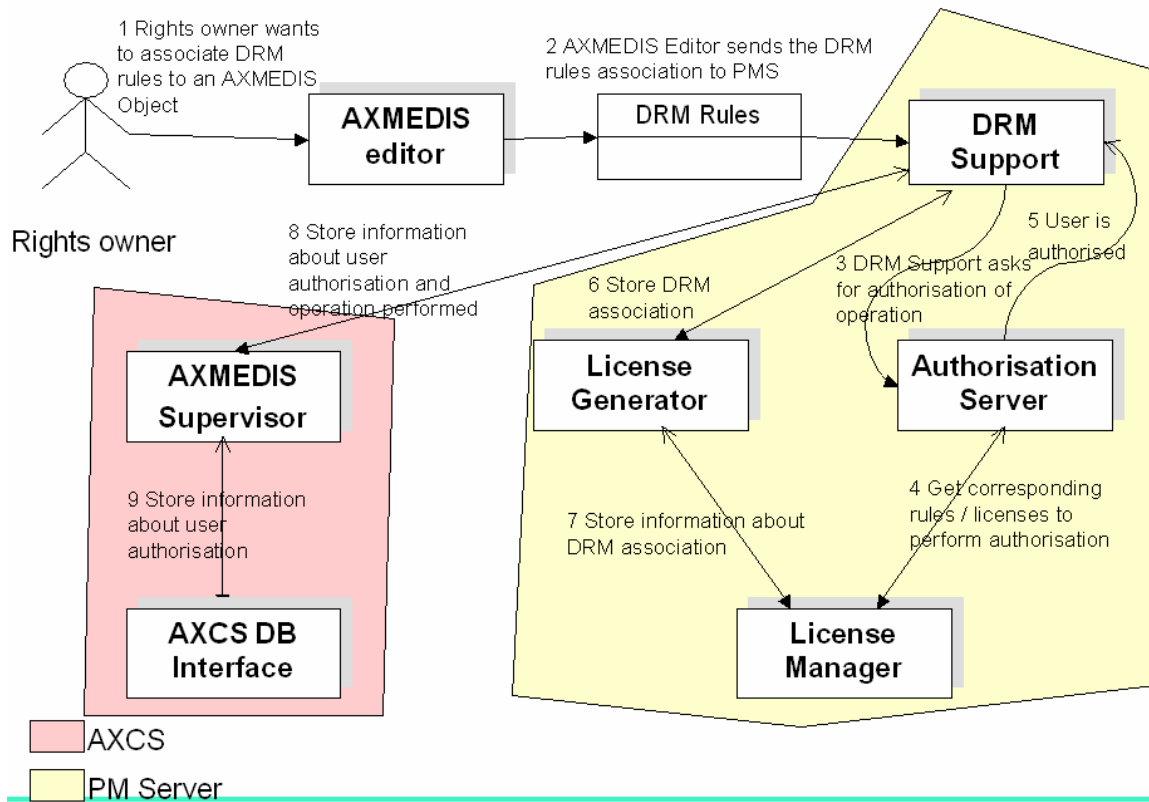


4.1 Protection Manager Support, General

4.1.1 Scenarios related with Protection Manager Support

4.1.1.1 DRM Association for License Production, license references AXMEDIS object

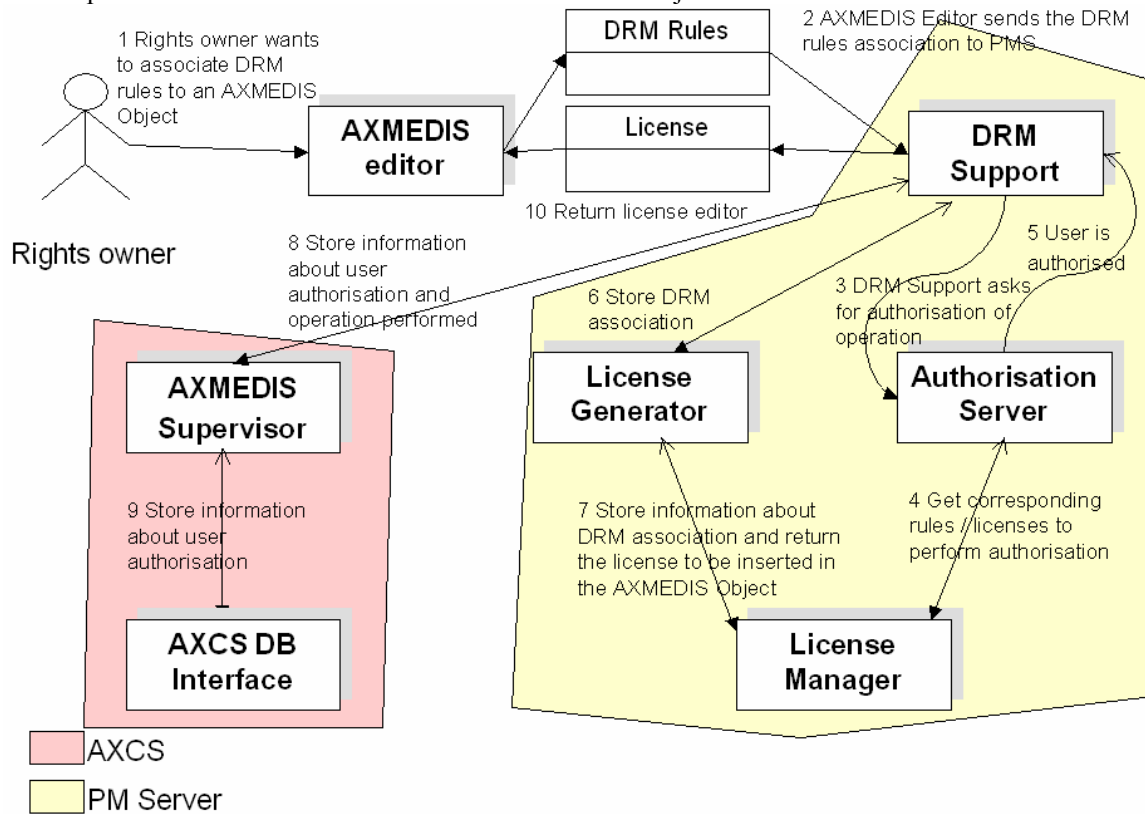
The scenario shows how DRM rules can be associated to AXMEDIS objects from AXMEDIS editor for license production. In this case license references AXMEDIS object.



DRM Association for license production, license references AXMEDIS object

4.1.1.2 DRM Association for License Production, license is stored inside the object

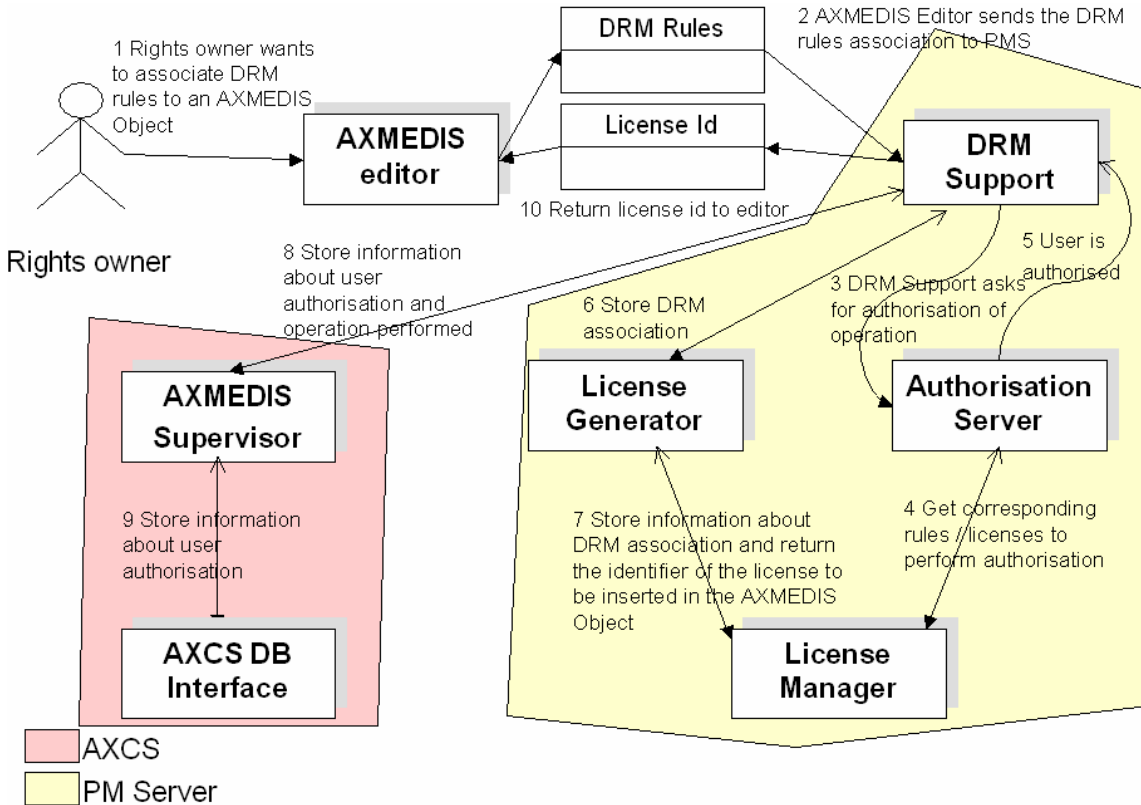
The scenario shows how DRM rules can be associated to AXMEDIS objects from AXMEDIS editor for license production. In this case license is stored inside the object.



DRM Association for license production, license is stored inside the object.

4.1.1.3 DRM Association for License Production, object stores license identifier

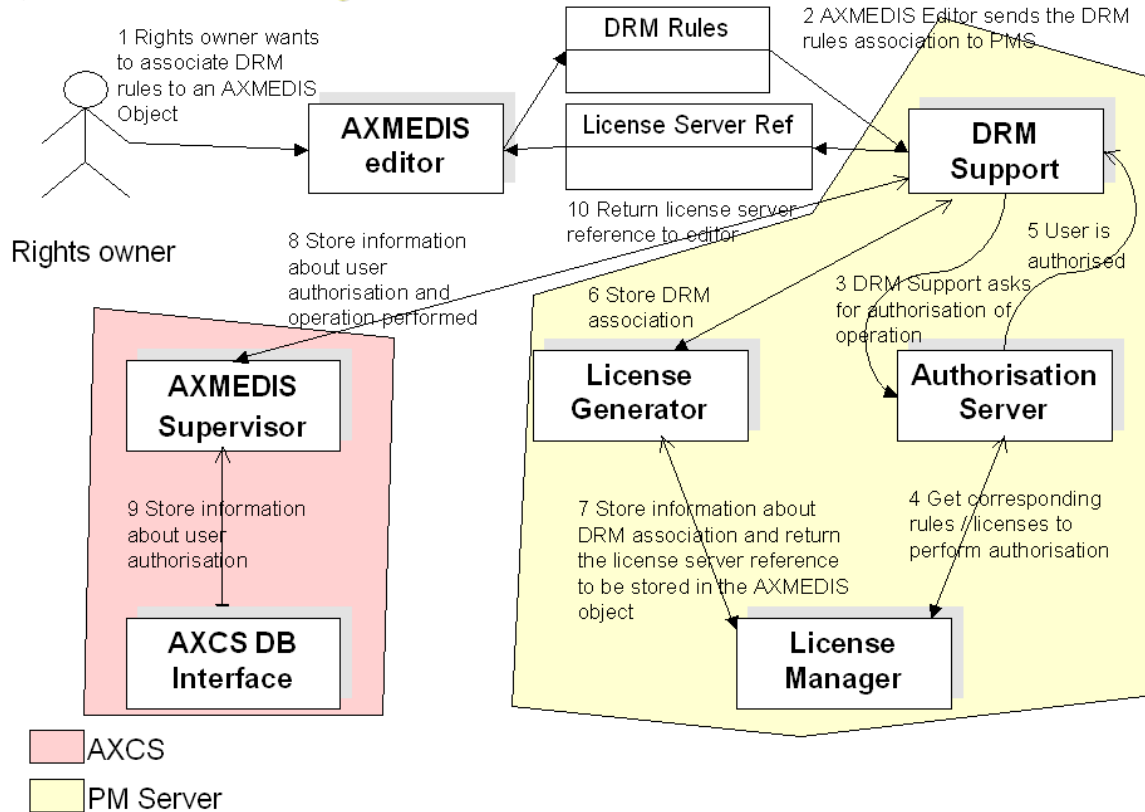
The scenario shows how DRM rules can be associated to AXMEDIS objects from AXMEDIS editor for license production. In this case AXMEDIS object stores license identifier.



DRM Association for license production, object stores license identifier

4.1.1.4 DRM Association for License Production, object stores reference to license server.

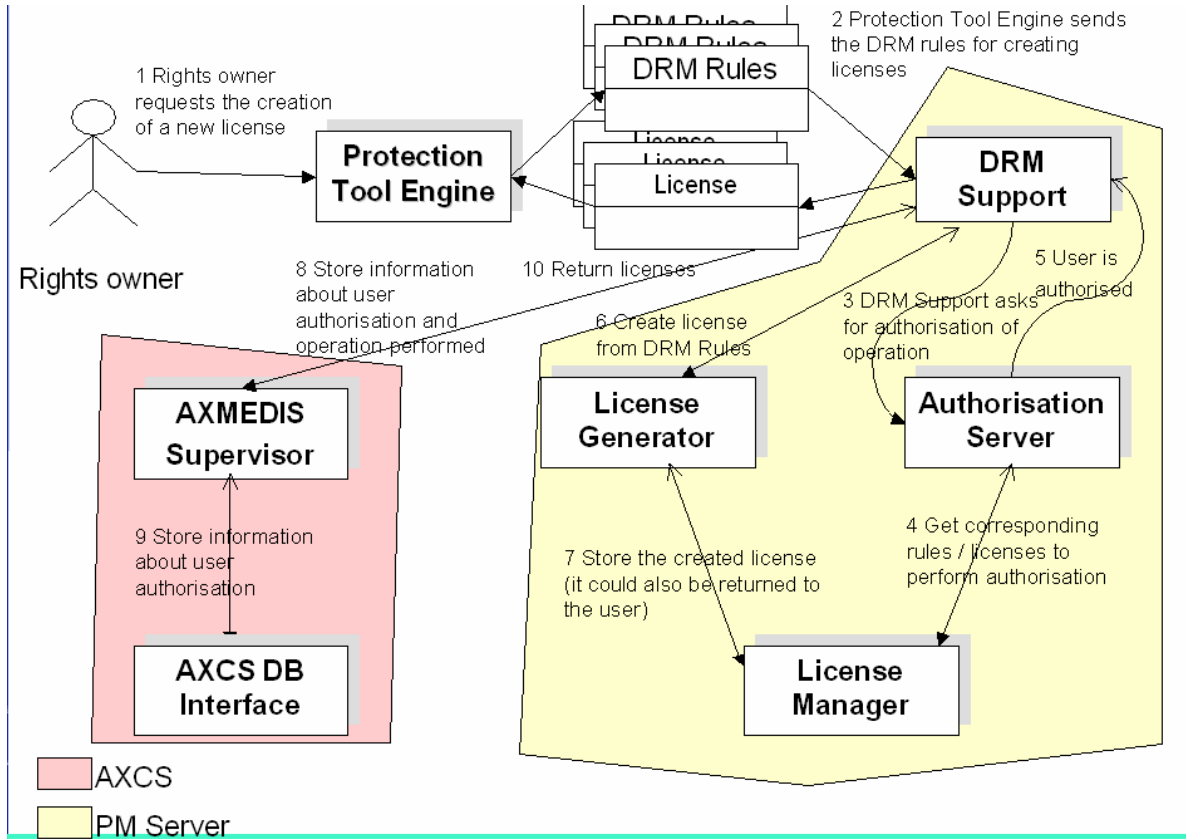
The scenario shows how DRM rules can be associated to AXMEDIS objects from AXMEDIS editor for license production. In this case AXMEDIS object stores reference to license server.



DRM Association for license production, object stores reference to license server.

4.1.1.5 Multiple license creation

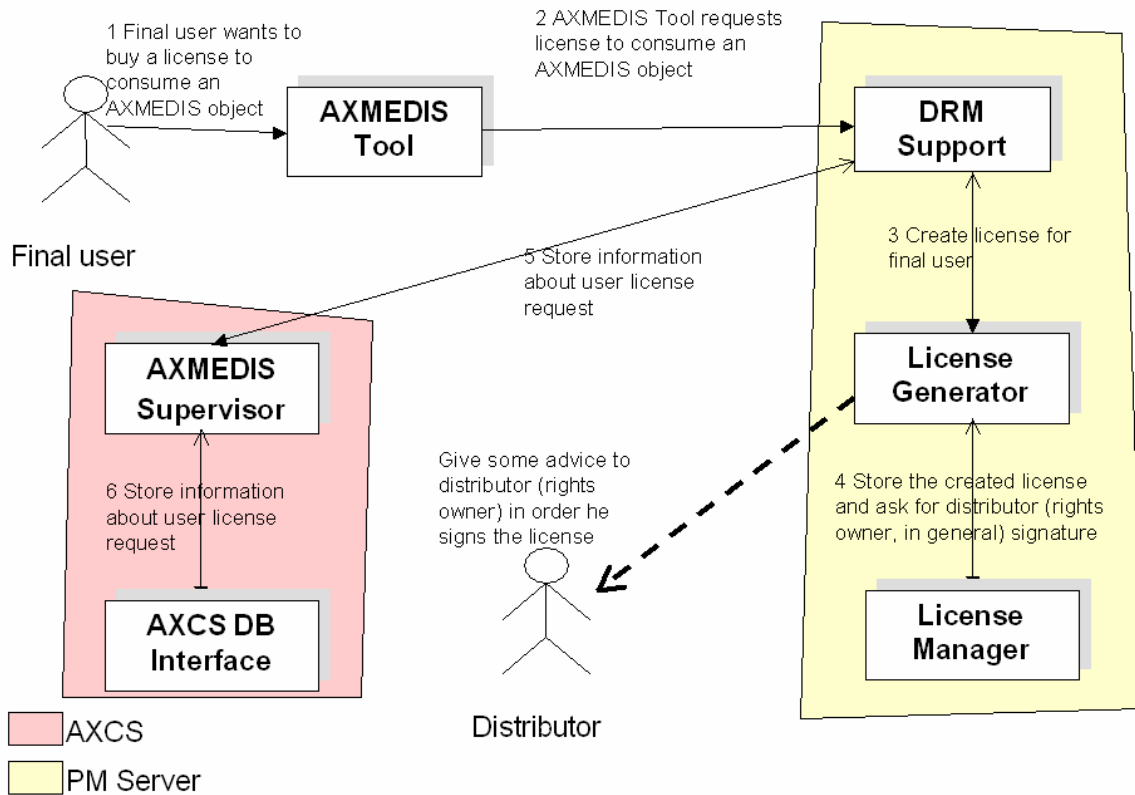
The scenario shows the creation of multiple licenses from protection tool engine.



Multiple license creation

4.1.1.6 License Acquisition (on-line use)

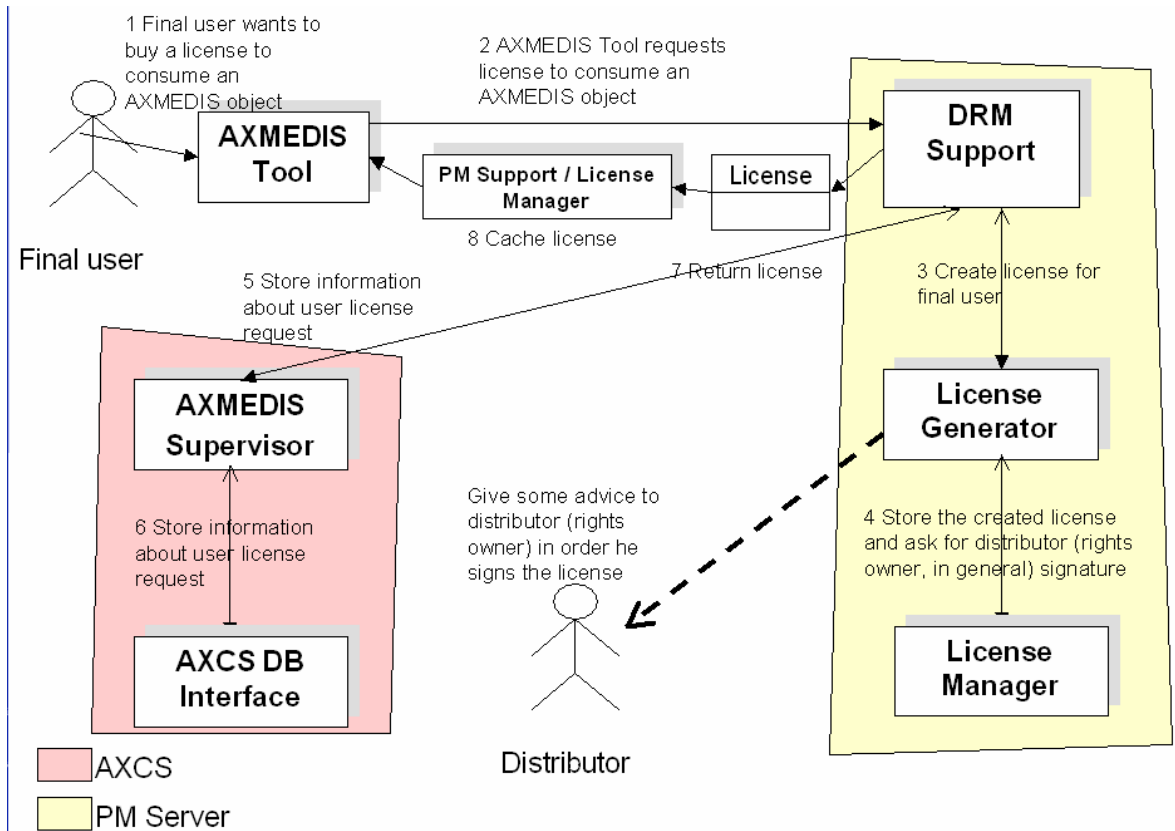
The scenario shows how a license can be acquired for on-line use.



License acquisition (on-line use)

4.1.1.7 License Acquisition (off-line use)

The scenario shows how a license can be acquired for off-line use.



License acquisition (off-line use)

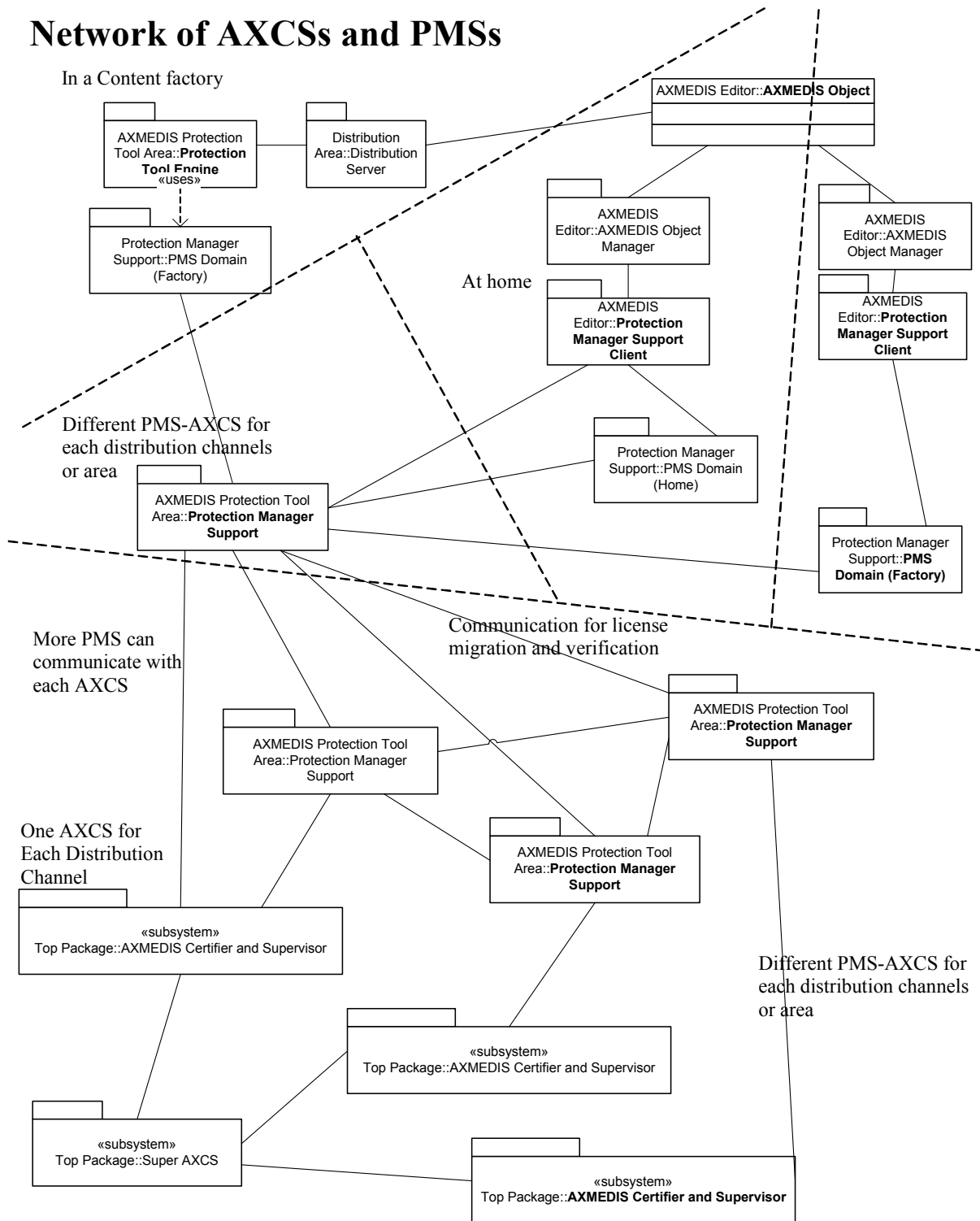
4.2 A Network of AXCS and PMS

The following figure shows the relationship among AXMEDIS Certifier and Supervisor and Protection Manager Support.

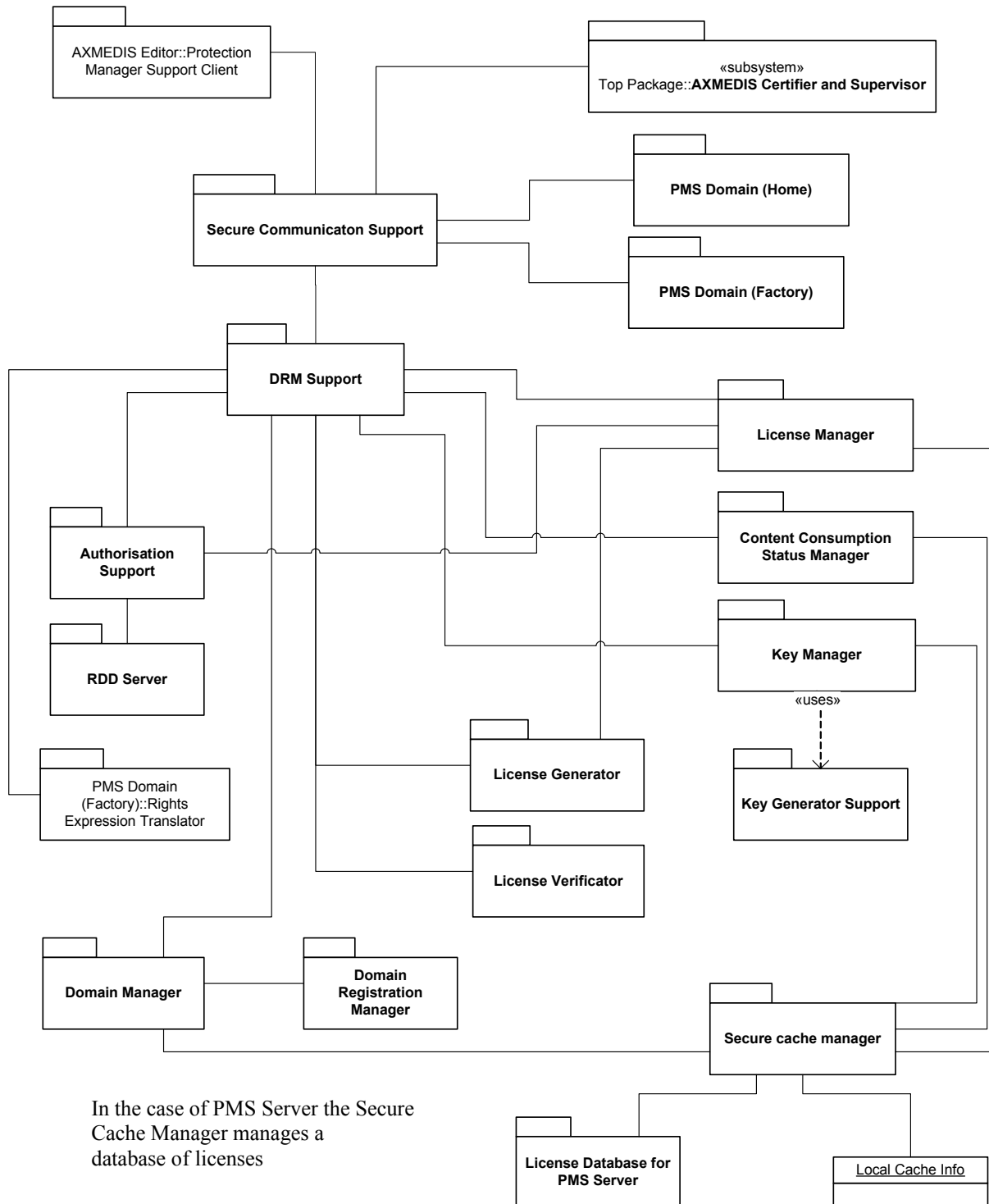
While Protection Manager Support Server can communicate with AXMEDIS Certifier and Supervisor, the rest of Protection Manager Support (Domain-Factory, Domain-Home and Client), can only connect with the user side or the Protection Manager Support Server of reference for them.

Different PMS Server can communicate among themselves. They could interchange protection information, protection models or even, action logs associated to their users.

Network of AXCSs and PMSs



Protection Manager Support (Server, Home, Client)



Next figure shows the general structure of PMS (Server, Home, Factory and Client). Not all the modules will be implemented for each PMS version, but selected ones. In the rest of this section, each PMS and their corresponding sub-modules will be explained in detail.

4.3 Protection Manager Support Server (FUPF)

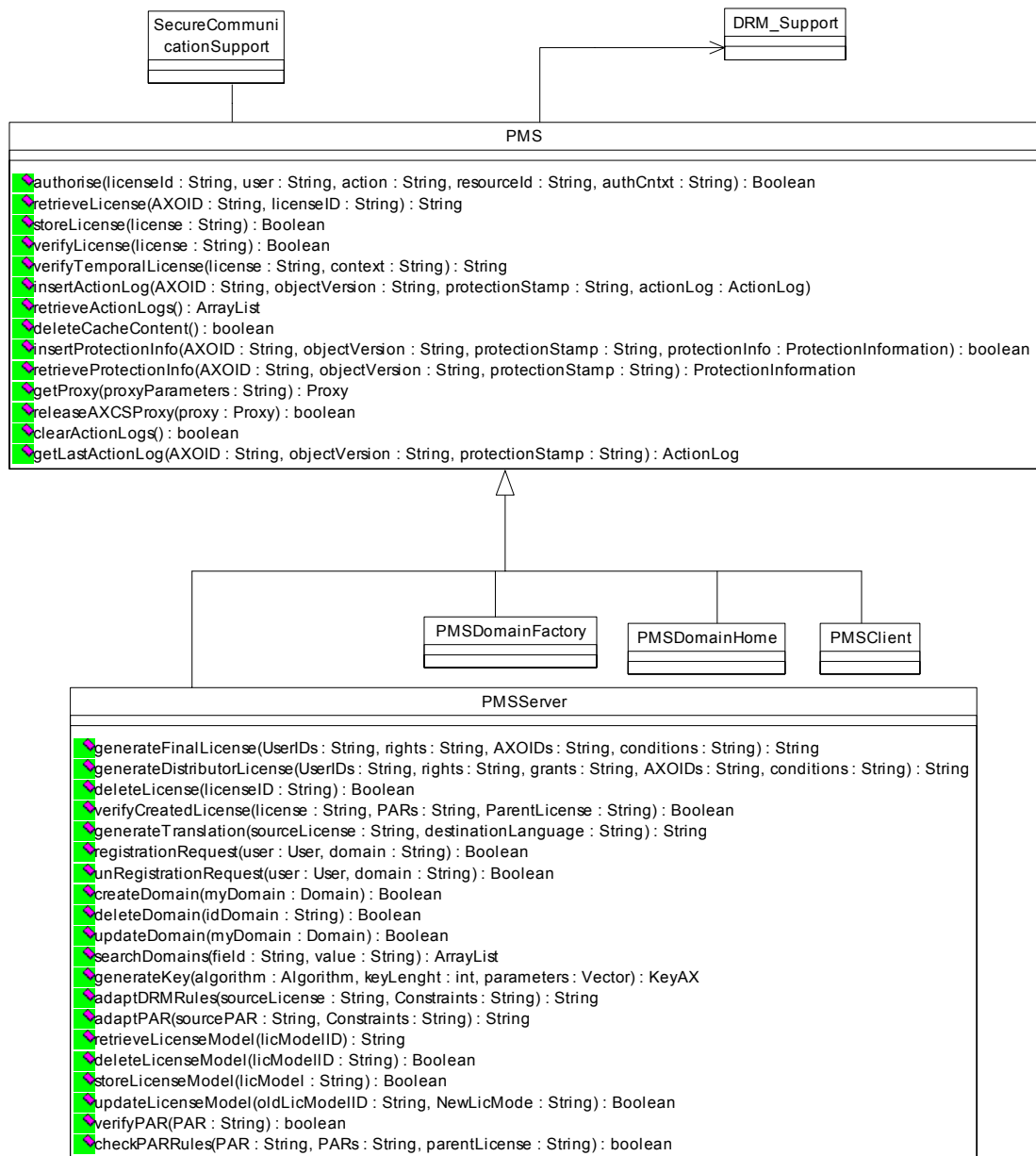
Protection Manager Support Server provides the protection needed for a set of PMS Domain Factory, Domain Home and Clients. It has connection with AXMEDIS Certifier and Supervisor, in order to check that users only perform the actions they are allowed to. In this section, the general functionality of this module is explained. In next sections, the modules forming part of PMS Server are explained in detail.

4.3.1 Module profile definition

| Module Profile | | |
|-----------------------------------|-----------------------------------|--|
| Protection Manager Support Server | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java/C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Secure Communication Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.3.2 Architecture of PMS

Next figure shows the architecture of PMS server module.



PMS Server class diagram

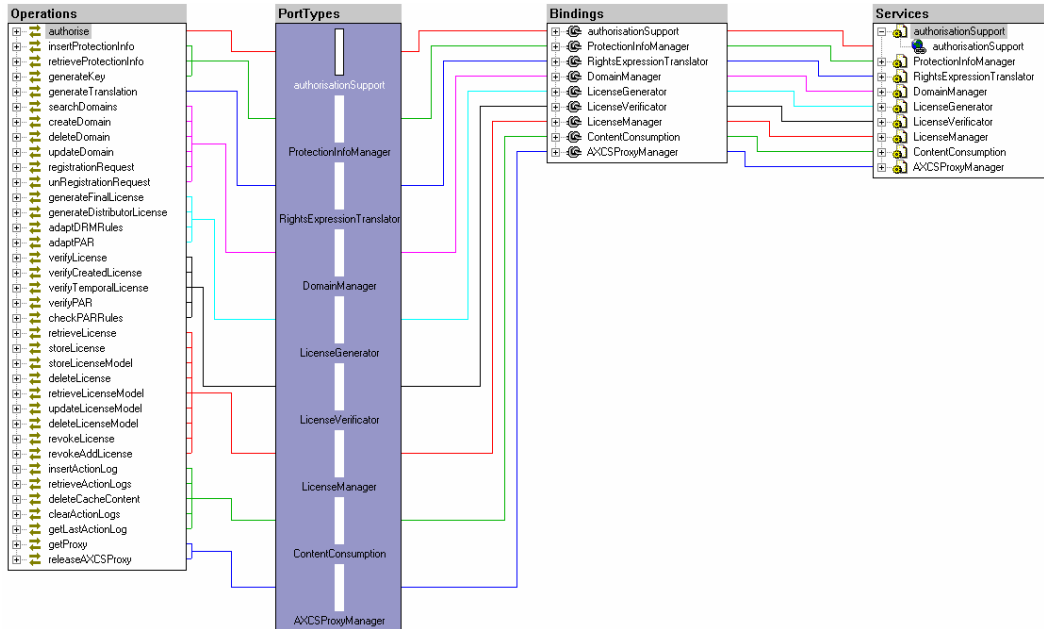
| PMS Server | |
|----------------------------|--|
| Methods | Description |
| generateFinalLicense | Generates a license for an end-user, according to the userID, right, AXOID and conditions. |
| generateDistributorLicense | Generates a license for a distributor, according to the userID, right, grants that this user will be able to distribute and conditions. |
| authorise | This function authorises AXMEDIS users to perform actions over AXMEDIS objects. It proves that a user has the appropriate license that grants him to exercise a right over a resource if the conditions are met based on the |

| | |
|------------------------|--|
| | execution context of the client. |
| retrieveLicense | This function retrieves the licenses stored in the license database. It retrieves the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| deleteLicense | This function deletes the licenses stored in the license database. It deletes the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| storeLicense | This function stores a license in the license database. |
| verifyLicense | Verifies a license syntactically against the schemas defined within the license. |
| verifyCreatedLicense | Verifies if the license can be generated according to the PARs and the parent licenses (e.g. Distributor or Creator licenses). |
| verifyTemporalLicense | Verifies that the license generated by the user fulfils the initial desirables requirements of the user. For example, the user can verify that with this license he could exercise the desired action over the AXObject. |
| generateTranslation | This function is used to return a destinationLicense. The system makes a translation from a sourceLicense to a destinationLicense. The destinationLicense, if the translation is possible, will be written in a languageDestination. |
| registrationRequest | This function is used to a registration of an user in a certain domain |
| unRegistrationRequest | This function is used to an unregistration of an user in a certain domain |
| createDomain | This function is used to create a new domain. |
| deleteDomain | This function is used to delete a domain with an idDomain identificator |
| updateDomain | This function is used to update a domain. |
| searchDomains | This function is used to return a list of domains with a certain value in a certain field. |
| insertActionLog | Stores the given action log associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveActionLogs | This method retrieves the action logs stored in the local cache info. |
| deleteCacheContent | This method retrieves the action logs stored in the local cache info. |
| insertProtectionInfo | Stores the given protection information associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveProtectionInfo | This method retrieves the requested protection information. The information needed to retrieve protection information is the AXMEDIS object identifier, the object version and the protection stamp. |
| generateKey | This method permits the creation of a key for protecting an AXMEDIS object. |
| adaptDRMRules | Adapts the given license using the given constraints |
| adaptPAR | Adapts the given PAR using the given constraints |
| verifyPAR | Verifies a PAR syntactically against the schemas defined within the PAR. |
| checkPARRules | Verifies if the PAR can be generated according to the parent licenses and PARs. |
| retrieveLicenseModel | This function returns a licenseModel stored in the License DB |
| deleteLicenseModel | This function deletes a licenseModel from the License DB |
| storeLicenseModel | This function inserts a license model in the License DB |
| updateLicenseModel | This function updates the licenseModel given a licenseModelID and a new description of licenseModel |
| revokeLicense | This function changes the status of a license to revoked in the License DB |
| revokeAddLicense | This function changes the status of a license to revoked and adds the license that substitutes it in the License DB |
| clearActionLogs | Deletes action logs from the cache, after positive authorisation of the user in the connected environment |
| getLastActionLog | Returns the last action log |
| getProxy | Returns a Proxy to establish a secure communication |

releaseAXCSProxy

Finish the secure communicaton

4.3.3 WSDL Interface



| PMS Server | |
|------------|--|
| WSDL | <pre> <wsdl:definitions xmlns:tns="urn:authorisationSupport" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:authorisationSupport" name="authorisationSupport"> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="action" type="xsd:string" nillable="true"/> <xsd:element name="resourceID" type="xsd:string" nillable="true"/> <xsd:element name="authContext" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> <xsd:element name="license" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" </pre> |

```

xmlns:tns="http://systinet.com/xsd/SchemaTypes/"
  <xsd:element name="license" type="xsd:string" nillable="true"/>
  <xsd:element name="boolean_Response" type="xsd:boolean"/>
  <xsd:element name="PARs" type="xsd:string" nillable="true"/>
  <xsd:element name="parentLicense" type="xsd:string" nillable="true"/>
  <xsd:element name="context" type="xsd:string" nillable="true"/>
  <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
  <xsd:element name="userIDs" type="xsd:string" nillable="true"/>
  <xsd:element name="rights" type="xsd:string" nillable="true"/>
  <xsd:element name="AXOIDs" type="xsd:string" nillable="true"/>
  <xsd:element name="conditions" type="xsd:string" nillable="true"/>
  <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
  <xsd:element name="grants" type="xsd:string" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
xmlns:map="http://systinet.com/mapping/">
  <xsd:complexType name="ActionLog">
    <!-- Defined in AXMEDIS Supervisor -->
    <xsd:sequence/>
  </xsd:complexType>
</xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
  <xsd:complexType name="List">
    <xsd:sequence>
      <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ArrayList">
    <xsd:complexContent mixed="false">
      <xsd:extension base="tns:List"/>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"
xmlns:xns5="http://systinet.com/containers/literal/ms.net">
  <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
  <xsd:import namespace="http://systinet.com/wsdl/default"/>
  <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
  <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
  <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
  <xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
  <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
  <xsd:complexType name="List">
    <xsd:sequence>
      <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Vector">
    <xsd:complexContent mixed="false">
      <xsd:extension base="tns:List"/>
    </xsd:complexContent>
  </xsd:complexType>

```

| | |
|--|---|
| | <pre> </xsd:complexType> <xsd:complexType name="ArrayOfAnyType"> <xsd:sequence> <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsd/default/" xmlns:xns5="http://systinet.com/containers/literal/ms.net"> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:import namespace="http://systinet.com/wsd/default"/> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="objectVersion" type="xsd:string" nillable="true"/> <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/> <xsd:element name="protectionInfo" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="ProtectionInformation_Response" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="algorithm" type="xns4:Algorithm" nillable="true"/> <xsd:element name="keyLength" type="xsd:int"/> <xsd:element name="parameters" type="xns5:Vector" nillable="true"/> <xsd:element name="KeyAX_Response" type="xns4:KeyAX" nillable="true"/> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/wsd/default/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsd/default/" xmlns:map="http://systinet.com/mapping"/> <xsd:complexType name="ProtectionInformation"> <xsd:sequence> <xsd:element name="protectionInfo" type="xsd:string" nillable="true"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="Algorithm"> <xsd:sequence> <xsd:element name="algorithm" type="xsd:int"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="KeyAX"> <xsd:sequence> <xsd:element name="key" type="xsd:base64Binary" nillable="true"/> <xsd:element name="keyByte" type="xsd:base64Binary" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/> <xsd:element name="destinationLanguage" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayList"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> </xsd:schema> </pre> |
|--|---|

```

<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/containers/literal/ms.net"
xmlns:xns5="urn:DomainPackage.DomainManager">
  <xsd:import namespace="urn:DomainPackage.DomainManager"/>
  <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
  <xsd:element name="field" type="xsd:string" nillable="true"/>
  <xsd:element name="value" type="xsd:string" nillable="true"/>
  <xsd:element name="ArrayList_Response" type="xns4:ArrayList" nillable="true"/>
  <xsd:element name="myDomain" type="xns5:Domain" nillable="true"/>
  <xsd:element name="boolean_Response" type="xsd:boolean"/>
  <xsd:element name="idDomain" type="xsd:string" nillable="true"/>
  <xsd:element name="myUser" type="xns5:User" nillable="true"/>
  <xsd:element name="myDomain_1" type="xsd:string" nillable="true"/>
  <xsd:element name="userID" type="xsd:string" nillable="true"/>
  <xsd:element name="myDomain_2" type="xsd:string" nillable="true"/>
</xsd:schema>
<xsd:schema targetNamespace="urn:DomainPackage.DomainManager"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:DomainPackage.DomainManager" xmlns:map="http://systinet.com/mapping/">
  <xsd:complexType name="Domain">
    <xsd:sequence>
      <xsd:element name="AXDOM" type="xsd:string" nillable="true"/>
      <xsd:element name="AXID" type="xsd:string" nillable="true"/>
      <xsd:element name="typeOfID" type="xsd:string" nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="User">
    <xsd:sequence/>
  </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="AXCSPProxy">
      <xsd:sequence>
        <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/>
        <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/>
        <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ActionLog">
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="CertificationResult">
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="VerificationResult">
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/">
    <xsd:import namespace="http://systinet.com/wsdl/default/">
    <xsd:element name="AXCSPProxy_Response" type="xns4:AXCSPProxy" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="userIDs" type="xsd:string" nillable="true"/>
    <xsd:element name="rights" type="xsd:string" nillable="true"/>
    <xsd:element name="AXOIDs" type="xsd:string" nillable="true"/>
    <xsd:element name="conditions" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
    <xsd:element name="grants" type="xsd:string" nillable="true"/>
  </xsd:schema>

```

```

        <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/>
        <xsd:element name="constraints" type="xsd:string" nillable="true"/>
        <xsd:element name="PAR" type="xsd:string" nillable="true"/>
    </xsd:schema>
</wsdl:types>
<wsdl:types>
    <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
        elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
        <xsd:element name="license" type="xsd:string" nillable="true"/>
        <xsd:element name="boolean_Response" type="xsd:boolean"/>
        <xsd:element name="PARs" type="xsd:string" nillable="true"/>
        <xsd:element name="parentLicense" type="xsd:string" nillable="true"/>
        <xsd:element name="context" type="xsd:string" nillable="true"/>
        <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
        <xsd:element name="PAR" type="xsd:string" nillable="true"/>
    </xsd:schema>
</wsdl:types>
<wsdl:types>
    <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
        elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
        <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
        <xsd:element name="licenseID" type="xsd:string" nillable="true"/>
        <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
        <xsd:element name="license" type="xsd:string" nillable="true"/>
        <xsd:element name="boolean_Response" type="xsd:boolean"/>
        <xsd:element name="licenseModel" type="xsd:string" nillable="true"/>
        <xsd:element name="licModelID" type="xsd:string" nillable="true"/>
        <xsd:element name="licModelNew" type="xsd:string" nillable="true"/>
        <xsd:element name="newLicense" type="xsd:string" nillable="true"/>
    </xsd:schema>
</wsdl:types>
<wsdl:types>
    <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
        xmlns:map="http://systinet.com/mapping/">
        <xsd:complexType name="ActionLog">
            <xsd:annotation>
                <xsd:appinfo>
                    <map:java-type name="ActionLog"/>
                </xsd:appinfo>
            </xsd:annotation>
            <xsd:sequence/>
        </xsd:complexType>
    </xsd:schema>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:tns="http://systinet.com/containers/literal/ms.net"
        targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
        <xsd:complexType name="List">
            <xsd:sequence>
                <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
            </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="ArrayList">
            <xsd:complexContent mixed="false">
                <xsd:extension base="tns:List"/>
            </xsd:complexContent>
        </xsd:complexType>
        <xsd:complexType name="LinkedList">
            <xsd:complexContent mixed="false">
                <xsd:extension base="tns:List"/>
            </xsd:complexContent>
        </xsd:complexType>
        <xsd:complexType name="Vector">
            <xsd:complexContent mixed="false">
                <xsd:extension base="tns:List"/>
            </xsd:complexContent>
        </xsd:complexType>
        <xsd:complexType name="Set">

```


| | |
|--|--|
| | <pre> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="HashSet"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:Set"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="SortedSet"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:Set"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="TreeSet"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:SortedSet"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="Map"> <xsd:sequence> <xsd:element name="Keys" type="tns:ArrayOfAnyType"/> <xsd:element name="Values" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="Hashtable"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:Map"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="HashMap"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:Map"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="SortedMap"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:Map"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="TreeMap"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:SortedMap"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="Iterator"> <xsd:sequence> <xsd:element name="Data" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="Enumeration"> <xsd:sequence> <xsd:element name="Data" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayOfAnyType"> <xsd:sequence> <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wSDL/default/" xmlns:xns5="http://systinet.com/containers/literal/ms.net"> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:import namespace="http://systinet.com/wSDL/default"/> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="objectVersion" type="xsd:string" nillable="true"/> <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/> </pre> |
|--|--|


```

<xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
<xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
<xsd:element name="boolean_Response" type="xsd:boolean"/>
<xsd:element name="ActionLog_Response" type="xns4:ActionLog" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="AXCSPProxy">
        <xsd:annotation>
            <xsd:appinfo>
                <map:java-type name="AXCSPProxy"/>
            </xsd:appinfo>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/>
            <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/>
            <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ActionLog">
        <xsd:annotation>
            <xsd:appinfo>
                <map:java-type name="ActionLog"/>
            </xsd:appinfo>
        </xsd:annotation>
        <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="CertificationResult">
        <xsd:annotation>
            <xsd:appinfo>
                <map:java-type name="CertificationResult"/>
            </xsd:appinfo>
        </xsd:annotation>
        <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="VerificationResult">
        <xsd:annotation>
            <xsd:appinfo>
                <map:java-type name="VerificationResult"/>
            </xsd:appinfo>
        </xsd:annotation>
        <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/">
    <xsd:import namespace="http://systinet.com/wsdl/default/">
    <xsd:element name="proxyParameters" type="xsd:string" nillable="true"/>
    <xsd:element name="AXCSPProxy_Response" type="xns4:AXCSPProxy" nillable="true"/>
    <xsd:element name="proxy" type="xns4:AXCSPProxy" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="authorisationSupport_authorise_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="authorisationSupport_authorise_1_Request">
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="action" element="ns0:action"/>
  <wsdl:part name="resourceID" element="ns0:resourceID"/>
  <wsdl:part name="authContext" element="ns0:authContext"/>
</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>

```

```

</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseValidator_verifyTemporalLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="context" element="ns0:context"/>
</wsdl:message>
<wsdl:message name="LicenseValidator_verifyCreatedLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseValidator_verifyCreatedLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>
<wsdl:message name="LicenseValidator_verifyTemporalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseValidator_verifyLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseValidator_verifyLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>
  <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
  <wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>
  <wsdl:part name="grants" element="ns0:grants"/>
  <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
  <wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/>
<wsdl:message name="ContentConsumption_retrieveActionLogs_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_insertActionLog_Response"/>
<wsdl:message name="ContentConsumption_insertActionLog_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
  <wsdl:part name="actionLog" element="ns0:actionLog"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"/>
<wsdl:message name="ContentConsumption_deleteCacheContent_Response"/>

```

```

<wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
  <wsdl:part name="protectionInfo" element="ns0:protectionInfo"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:ProtectionInformation_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_1_Request">
  <wsdl:part name="algorithm" element="ns0:algorithm"/>
  <wsdl:part name="keyLength" element="ns0:keyLength"/>
  <wsdl:part name="parameters" element="ns0:parameters"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_Response">
  <wsdl:part name="response" element="ns0:KeyAX_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_1_Request">
  <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/>
  <wsdl:part name="destinationLanguage" element="ns0:destinationLanguage"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_1_Request">
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_2"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_1_Request">
  <wsdl:part name="idDomain" element="ns0:idDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_1_Request">
  <wsdl:part name="myUser" element="ns0:myUser"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_1"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_1_Request">

```

| | |
|--|--|
| | <pre> <wsdl:part name="field" element="ns0:field"/> <wsdl:part name="value" element="ns0:value"/> </wsdl:message> <wsdl:message name="AXCSProxy_getProxy_1_Request"/> <wsdl:message name="AXCSProxy_getProxy_Response"> <wsdl:part name="response" element="ns0:AXCSProxy_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptPAR_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptDRMRules_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptPAR_1_Request"> <wsdl:part name="PAR" element="ns0:PAR"/> <wsdl:part name="constraints" element="ns0:constraints"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="grants" element="ns0:grants"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptDRMRules_1_Request"> <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/> <wsdl:part name="constraints" element="ns0:constraints"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseVericator_verifyCreatedLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> <wsdl:part name="PARs" element="ns0:PARs"/> <wsdl:part name="parentLicense" element="ns0:parentLicense"/> </wsdl:message> <wsdl:message name="LicenseVericator_checkPARRules_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVericator_checkPARRules_1_Request"> <wsdl:part name="PAR" element="ns0:PAR"/> <wsdl:part name="PARs" element="ns0:PARs"/> <wsdl:part name="parentLicense" element="ns0:parentLicense"/> </wsdl:message> <wsdl:message name="LicenseVericator_verifyCreatedLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVericator_verifyPAR_1_Request"> <wsdl:part name="PAR" element="ns0:PAR"/> </wsdl:message> <wsdl:message name="LicenseVericator_verifyTemporalLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseVericator_verifyLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </wsdl:message> <wsdl:message name="LicenseVericator_verifyPAR_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVericator_verifyTemporalLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </pre> |
|--|--|

| | |
|--|--|
| | <pre> <wsdl:part name="context" element="ns0:context"/> </wsdl:message> <wsdl:message name="LicenseVerifier_verifyLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeLicense_1_Request"> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicenseModel_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_updateLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> <wsdl:part name="licModelNew" element="ns0:licModelNew"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeAddLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeAddLicense_1_Request"> <wsdl:part name="licenseID" element="ns0:licenseID"/> <wsdl:part name="newLicense" element="ns0:newLicense"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicenseModel_1_Request"> <wsdl:part name="licenseModel" element="ns0:licenseModel"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> </wsdl:message> <wsdl:message name="LicenseManager_updateLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </wsdl:message> <wsdl:message name="ContentConsumption_clearActionLogs_1_Request"/> <wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/> <wsdl:message name="ContentConsumption_getLastActionLog_Response"> <wsdl:part name="response" element="ns0:ActionLog_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_getLastActionLog_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> </pre> |
|--|--|

| | |
|--|---|
| | <pre> <wsdl:part name="objectVersion" element="ns0:objectVersion"/> <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/> </wsdl:message> <wsdl:message name="ContentConsumption_deleteCacheContent_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"/> <wsdl:message name="ContentConsumption_retrieveActionLogs_Response"> <wsdl:part name="response" element="ns0:ArrayList_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_insertActionLog_Response"/> <wsdl:message name="ContentConsumption_insertActionLog_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="objectVersion" element="ns0:objectVersion"/> <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/> <wsdl:part name="actionLog" element="ns0:actionLog"/> </wsdl:message> <wsdl:message name="ContentConsumption_clearActionLogs_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_releaseAXCSProxy_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_releaseAXCSProxy_1_Request"> <wsdl:part name="proxy" element="ns0:proxy"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_getProxy_Response"> <wsdl:part name="response" element="ns0:AXCSProxy_Response"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_getProxy_1_Request"> <wsdl:part name="proxyParameters" element="ns0:proxyParameters"/> </wsdl:message> <wsdl:portType name="authorisationSupport"> <wsdl:operation name="authorise" parameterOrder="licenseID userID action resourceID authContext"> <wsdl:input message="tns:authorisationSupport_authorise_1_Request"/> <wsdl:output message="tns:authorisationSupport_authorise_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="ProtectionInfoManager"> <wsdl:operation name="insertProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp protectionInfo"> <wsdl:input message="tns:ProtectionInfoManager_insertProtectionInfo_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_insertProtectionInfo_Response"/> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp"> <wsdl:input message="tns:ProtectionInfoManager_retrieveProtectionInfo_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_retrieveProtectionInfo_Response"/> </wsdl:operation> <wsdl:operation name="generateKey" parameterOrder="algorithm keyLength parameters"> <wsdl:input message="tns:ProtectionInfoManager_generateKey_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_generateKey_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="RightsExpressionTranslator"> <wsdl:operation name="generateTranslation" parameterOrder="sourceLicense destinationLanguage"> <wsdl:input message="tns:RightsExpressionTranslator_generateTranslation_1_Request"/> <wsdl:output message="tns:RightsExpressionTranslator_generateTranslation_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="DomainManager"> <wsdl:operation name="searchDomains" parameterOrder="field value"> <wsdl:input message="tns:DomainManager_searchDomains_1_Request"/> <wsdl:output message="tns:DomainManager_searchDomains_Response"/> </wsdl:operation> <wsdl:operation name="createDomain" parameterOrder="myDomain"> <wsdl:input message="tns:DomainManager_createDomain_1_Request"/> <wsdl:output message="tns:DomainManager_createDomain_Response"/> </wsdl:operation> <wsdl:operation name="deleteDomain" parameterOrder="idDomain"> </pre> |
|--|---|

```

        <wsdl:input message="tns:DomainManager_deleteDomain_1_Request"/>
        <wsdl:output message="tns:DomainManager_deleteDomain_Response"/>
    </wsdl:operation>
    <wsdl:operation name="updateDomain" parameterOrder="myDomain">
        <wsdl:input message="tns:DomainManager_updateDomain_1_Request"/>
        <wsdl:output message="tns:DomainManager_updateDomain_Response"/>
    </wsdl:operation>
    <wsdl:operation name="registrationRequest" parameterOrder="myUser myDomain">
        <wsdl:input message="tns:DomainManager_registrationRequest_1_Request"/>
        <wsdl:output message="tns:DomainManager_registrationRequest_Response"/>
    </wsdl:operation>
    <wsdl:operation name="unRegistrationRequest" parameterOrder="userID myDomain">
        <wsdl:input message="tns:DomainManager_unRegistrationRequest_1_Request"/>
        <wsdl:output message="tns:DomainManager_unRegistrationRequest_Response"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseGenerator">
    <wsdl:operation name="generateFinalLicense" parameterOrder="userIDs rights AXOIDs conditions">
        <wsdl:input message="tns:LicenseGenerator_generateFinalLicense_1_Request"/>
        <wsdl:output message="tns:LicenseGenerator_generateFinalLicense_Response"/>
    </wsdl:operation>
    <wsdl:operation name="generateDistributorLicense" parameterOrder="userIDs rights grants AXOIDs
conditions">
        <wsdl:input message="tns:LicenseGenerator_generateDistributorLicense_1_Request"/>
        <wsdl:output message="tns:LicenseGenerator_generateDistributorLicense_Response"/>
    </wsdl:operation>
    <wsdl:operation name="adaptDRMRules" parameterOrder="sourceLicense constraints">
        <wsdl:input message="tns:LicenseGenerator_adaptDRMRules_1_Request"/>
        <wsdl:output message="tns:LicenseGenerator_adaptDRMRules_Response"/>
    </wsdl:operation>
    <wsdl:operation name="adaptPAR" parameterOrder="PAR constraints">
        <wsdl:input message="tns:LicenseGenerator_adaptPAR_1_Request"/>
        <wsdl:output message="tns:LicenseGenerator_adaptPAR_Response"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseVerificator">
    <wsdl:operation name="verifyLicense" parameterOrder="license">
        <wsdl:input message="tns:LicenseVerificator_verifyLicense_1_Request"/>
        <wsdl:output message="tns:LicenseVerificator_verifyLicense_Response"/>
    </wsdl:operation>
    <wsdl:operation name="verifyCreatedLicense" parameterOrder="license PARs parentLicense">
        <wsdl:input message="tns:LicenseVerificator_verifyCreatedLicense_1_Request"/>
        <wsdl:output message="tns:LicenseVerificator_verifyCreatedLicense_Response"/>
    </wsdl:operation>
    <wsdl:operation name="verifyTemporalLicense" parameterOrder="license context">
        <wsdl:input message="tns:LicenseVerificator_verifyTemporalLicense_1_Request"/>
        <wsdl:output message="tns:LicenseVerificator_verifyTemporalLicense_Response"/>
    </wsdl:operation>
    <wsdl:operation name="verifyPAR" parameterOrder="PAR">
        <wsdl:input message="tns:LicenseVerificator_verifyPAR_1_Request"/>
        <wsdl:output message="tns:LicenseVerificator_verifyPAR_Response"/>
    </wsdl:operation>
    <wsdl:operation name="checkPARRules" parameterOrder="PAR PARs parentLicense">
        <wsdl:input message="tns:LicenseVerificator_checkPARRules_1_Request"/>
        <wsdl:output message="tns:LicenseVerificator_checkPARRules_Response"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseManager">
    <wsdl:operation name="retrieveLicense" parameterOrder="AXOID licenseID">
        <wsdl:input message="tns:LicenseManager_retrieveLicense_1_Request"/>
        <wsdl:output message="tns:LicenseManager_retrieveLicense_Response"/>
    </wsdl:operation>
    <wsdl:operation name="storeLicense" parameterOrder="license">
        <wsdl:input message="tns:LicenseManager_storeLicense_1_Request"/>
        <wsdl:output message="tns:LicenseManager_storeLicense_Response"/>
    </wsdl:operation>
    <wsdl:operation name="storeLicenseModel" parameterOrder="licenseModel">
        <wsdl:input message="tns:LicenseManager_storeLicenseModel_1_Request"/>
        <wsdl:output message="tns:LicenseManager_storeLicenseModel_Response"/>
    </wsdl:operation>

```

```
<wsdl:operation name="deleteLicense" parameterOrder="AXOID licenseID">
    <wsdl:input message="tns:LicenseManager_deleteLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_deleteLicense_Response"/>
</wsdl:operation>
<wsdl:operation name="retrieveLicenseModel" parameterOrder="licModelID">
    <wsdl:input message="tns:LicenseManager_retrieveLicenseModel_1_Request"/>
    <wsdl:output message="tns:LicenseManager_retrieveLicenseModel_Response"/>
</wsdl:operation>
<wsdl:operation name="updateLicenseModel" parameterOrder="licModelID licModelNew">
    <wsdl:input message="tns:LicenseManager_updateLicenseModel_1_Request"/>
    <wsdl:output message="tns:LicenseManager_updateLicenseModel_Response"/>
</wsdl:operation>
<wsdl:operation name="deleteLicenseModel" parameterOrder="licModelID">
    <wsdl:input message="tns:LicenseManager_deleteLicenseModel_1_Request"/>
    <wsdl:output message="tns:LicenseManager_deleteLicenseModel_Response"/>
</wsdl:operation>
<wsdl:operation name="revokeLicense" parameterOrder="licenseID">
    <wsdl:input message="tns:LicenseManager_revokeLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_revokeLicense_Response"/>
</wsdl:operation>
<wsdl:operation name="revokeAddLicense" parameterOrder="licenseID newLicense">
    <wsdl:input message="tns:LicenseManager_revokeAddLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_revokeAddLicense_Response"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="ContentConsumption">
    <wsdl:operation name="insertActionLog" parameterOrder="AXOID objectVersion protectionStamp actionLog">
        <wsdl:input message="tns:ContentConsumption_insertActionLog_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_insertActionLog_Response"/>
    </wsdl:operation>
    <wsdl:operation name="retrieveActionLogs">
        <wsdl:input message="tns:ContentConsumption_retrieveActionLogs_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_retrieveActionLogs_Response"/>
    </wsdl:operation>
    <wsdl:operation name="deleteCacheContent">
        <wsdl:input message="tns:ContentConsumption_deleteCacheContent_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_deleteCacheContent_Response"/>
    </wsdl:operation>
    <wsdl:operation name="clearActionLogs">
        <wsdl:input message="tns:ContentConsumption_clearActionLogs_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_clearActionLogs_Response"/>
    </wsdl:operation>
    <wsdl:operation name="getLastActionLog" parameterOrder="AXOID objectVersion protectionStamp">
        <wsdl:input message="tns:ContentConsumption_getLastActionLog_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_getLastActionLog_Response"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="AXCSProxyManager">
    <wsdl:operation name="getProxy" parameterOrder="proxyParameters">
        <wsdl:input message="tns:AXCSProxyManager_getProxy_1_Request"/>
        <wsdl:output message="tns:AXCSProxyManager_getProxy_Response"/>
    </wsdl:operation>
    <wsdl:operation name="releaseAXCSProxy" parameterOrder="proxy">
        <wsdl:input message="tns:AXCSProxyManager_releaseAXCSProxy_1_Request"/>
        <wsdl:output message="tns:AXCSProxyManager_releaseAXCSProxy_Response"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="authorisationSupport" type="tns:authorisationSupport">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="authorise">
        <map:java-operation name="authorise"
signature="KEXqYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nOyla"/>
        <soap:operation
soapAction="urn:authorisationSupportauthorisationSupport#authorise#KEXqYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nO0qxYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
        <wsdl:input>
            <soap:hdtv use="literal"/>

```


| | |
|--|--|
| | <pre> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="ProtectionInfoManager" type="tns:ProtectionInfoManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="insertProtectionInfo"> <map:java-operation name="insertProtectionInfo" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWNoaW9uSW5mb3JtYXRpb247KVo="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#insertProtectionInfo#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWNoaW9uSW5mb3JtYXRpb247KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo"> <map:java-operation name="retrieveProtectionInfo" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWNoaW9uSW5mb3JtYXRpb247KVo=" style="document"/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#retrieveProtectionInfo#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWNoaW9uSW5mb3JtYXRpb247KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateKey"> <map:java-operation name="generateKey" signature="KExBbGdvcmI0aG07SUxqYXZhL3V0aWwVmdG9yOylMS2V5QVg7"/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#generateKey#KExBbGdvcmI0aG07SUxqYXZhL3V0aWwVmdG9yOylMS2V5QVg7" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="RightsExpressionTranslator" type="tns:RightsExpressionTranslator"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="generateTranslation"> <map:java-operation name="generateTranslation" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs="/> <soap:operation soapAction="urn:translator.RightsExpressionTranslatorRightsExpressionTranslator#generateTranslation#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> </pre> |
|--|--|

| | |
|--|---|
| | <pre> <wsdl:binding name="DomainManager" type="tns:DomainManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="searchDomains"> <map:java-operation name="searchDomains" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGldDs="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#searchDomains#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="createDomain"> <map:java-operation name="createDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWpjbWg==" /> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#createDomain#KExEb21haW5QYWNRyWdlL0RvbWpjbWg==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteDomain"> <map:java-operation name="deleteDomain" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#deleteDomain#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="updateDomain"> <map:java-operation name="updateDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWpjbWg==" /> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#updateDomain#KExEb21haW5QYWNRyWdlL0RvbWpjbWg==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="registrationRequest"> <map:java-operation name="registrationRequest" signature="KExEb21haW5QYWNRyWdlL1VzZXI7TGphdmEvbGFuZy9TdHJpbmc7KV0=" /> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#registrationRequest#KExEb21haW5QYWNRyWdlL1VzZXI7TGphdmEvbGFuZy9TdHJpbmc7KV0=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="unRegistrationRequest"> <map:java-operation name="unRegistrationRequest" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> </pre> |
|--|---|

| | |
|--|---|
| | <pre> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#unRegistrationRequest#KExqYXZhL2xhb mcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseGenerator" type="tns:LicenseGenerator"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="generateFinalLicense"> <map:java-operation name="generateFinalLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=""/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#generateFinalLicense#KExqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2 YS9sYW5nL1N0cmluZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateDistributorLicense"> <map:java-operation name="generateDistributorLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=""/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#generateDistributorLicense#KExqYXZhL2xhbmcvU3Rya W5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqY XZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="adaptDRMRules"> <map:java-operation name="adaptDRMRules" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmlu Zzs=""/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptDRMRules#KExqYXZhL2xhbmcvU3RyaW5nO0xq YXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="adaptPAR"> <map:java-operation name="adaptPAR" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmlu Zzs=""/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptPAR#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL 2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </pre> |
|--|---|

| | |
|--|--|
| | <pre> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseVerifier" type="tns:LicenseVerifier"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="verifyLicense"> <map:java-operation name="verifyLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVerifierLicenseVerifier#verifyLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyCreatedLicense"> <map:java-operation name="verifyCreatedLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVerifierLicenseVerifier#verifyCreatedLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyTemporalLicense"> <map:java-operation name="verifyTemporalLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs="/> <soap:operation soapAction="urn:LicenseVerifierLicenseVerifier#verifyTemporalLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyPAR"> <map:java-operation name="verifyPAR" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVerifierLicenseVerifier#verifyPAR#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="checkPARRules"> <map:java-operation name="checkPARRules" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVerifierLicenseVerifier#checkPARRules#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </pre> |
|--|--|

| | |
|--|---|
| | <pre> </wsdl:binding> <wsdl:binding name="LicenseManager" type="tns:LicenseManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="retrieveLicense"> <map:java-operation name="retrieveLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs="/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#retrieveLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeLicense"> <map:java-operation name="storeLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#storeLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeLicenseModel"> <map:java-operation name="storeLicenseModel" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#storeLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteLicense"> <map:java-operation name="deleteLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#deleteLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveLicenseModel"> <map:java-operation name="retrieveLicenseModel" signature="KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs="/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#retrieveLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="updateLicenseModel"> <map:java-operation name="updateLicenseModel" </pre> |
|--|---|

```
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/>
    <soap:operation
      soapAction="urn:LicenseManagerLicenseManager#updateLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="deleteLicenseModel">
    <map:java-operation name="deleteLicenseModel"
      signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/>
    <soap:operation
      soapAction="urn:LicenseManagerLicenseManager#deleteLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="revokeLicense">
    <map:java-operation name="revokeLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/>
    <soap:operation
      soapAction="urn:LicenseManagerLicenseManager#revokeLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="revokeAddLicense">
    <map:java-operation name="revokeAddLicense"
      signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/>
    <soap:operation
      soapAction="urn:LicenseManagerLicenseManager#revokeAddLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ContentConsumption" type="tns:ContentConsumption">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="insertActionLog">
    <map:java-operation name="insertActionLog"
      signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xBy3Rpb25Mb2c7KVY="/>
    <soap:operation
      soapAction="urn:ContentConsumptionContentConsumption#insertActionLog#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xBy3Rpb25Mb2c7KVY=" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="retrieveActionLogs">
    <map:java-operation name="retrieveActionLogs"
      signature="KCIMamF2YS91dGlsLOfvcmf5TGldDs="/>
```


| | |
|--|---|
| | <pre> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#retrieveActionLogs#KCIMamF2YS91dGlsL0Fycm F5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteCacheContent"> <map:java-operation name="deleteCacheContent" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#deleteCacheContent#KClA" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="clearActionLogs"> <map:java-operation name="clearActionLogs" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#clearActionLogs#KClA" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="getLastActionLog"> <map:java-operation name="getLastActionLog" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n OylMQWN0aW9uTG9nOw==" /> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#getLastActionLog#KExqYXZhL2xhbmcvU3RyaW 5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMQWN0aW9uTG9nOw==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="AXCSPProxyManager" type="tns:AXCSPProxyManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="getProxy"> <map:java-operation name="getProxy" signature="KExqYXZhL2xhbmcvU3RyaW5nOylMQVhDU1Byb3h5Ow==" /> <soap:operation soapAction="urn:AXCSPProxyManagerAXCSPProxyManager#getProxy#KExqYXZhL2xhbmcvU3RyaW5nOylMQ VhDU1Byb3h5Ow==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="releaseAXCSPProxy"> <map:java-operation name="releaseAXCSPProxy" signature="KExBWENTUHJveHk7KV0=" /> <soap:operation soapAction="urn:AXCSPProxyManagerAXCSPProxyManager#releaseAXCSPProxy#KExBWENTUHJveHk7KV0=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> </wsdl:operation> </pre> |
|--|---|

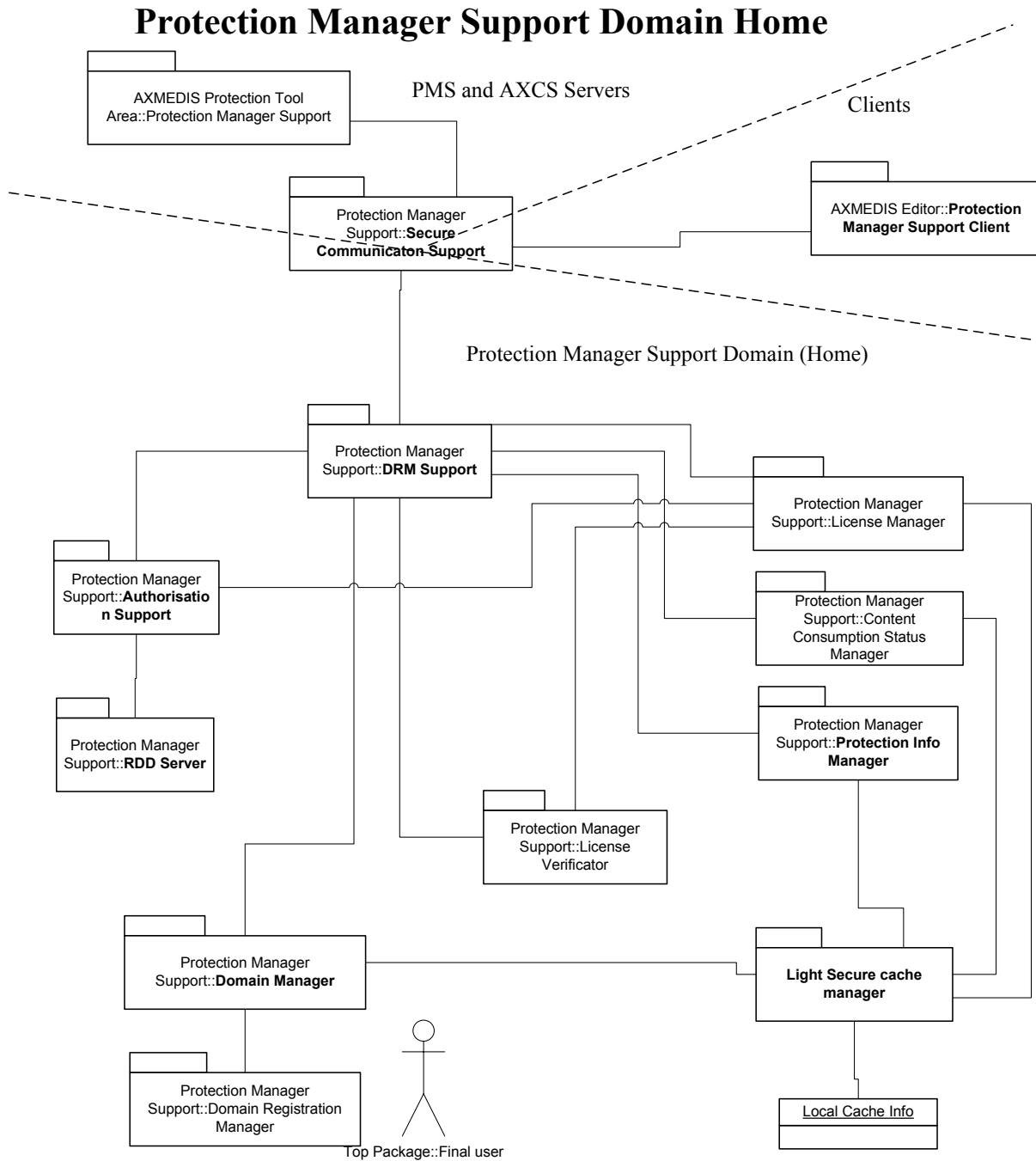
| | |
|--|---|
| | <pre> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="authorisationSupport"> <wsdl:port name="authorisationSupport" binding="tns:authorisationSupport"> <soap:address location="http://cjlal1:6060/authorisationSupport/"> </wsdl:port> </wsdl:service> <wsdl:service name="ProtectionInfoManager"> <wsdl:port name="ProtectionInfoManager" binding="tns:ProtectionInfoManager"> <soap:address location="http://efpc037:6060/ProtectionInfoManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="RightsExpressionTranslator"> <wsdl:port name="RightsExpressionTranslator" binding="tns:RightsExpressionTranslator"> <soap:address location="http://EFPC016:6060/RightsExpressionTranslator/"> </wsdl:port> </wsdl:service> <wsdl:service name="DomainManager"> <wsdl:port name="DomainManager" binding="tns:DomainManager"> <soap:address location="http://EFPC016:6060/DomainManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseGenerator"> <wsdl:port name="LicenseGenerator" binding="tns:LicenseGenerator"> <soap:address location="http://efpc037:6060/LicenseGenerator/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseVerifier"> <wsdl:port name="LicenseVerifier" binding="tns:LicenseVerifier"> <soap:address location="http://efpc037:6060/LicenseVerifier/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseManager"> <wsdl:port name="LicenseManager" binding="tns:LicenseManager"> <soap:address location="http://efpc037:6060/LicenseManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="ContentConsumption"> <wsdl:port name="ContentConsumption" binding="tns:ContentConsumption"> <soap:address location="http://efpc037:6060/ContentConsumption/"> </wsdl:port> </wsdl:service> <wsdl:service name="AXCSProxyManager"> <wsdl:port name="AXCSProxyManager" binding="tns:AXCSProxyManager"> <soap:address location="http://cjlal1:6060/AXCSProxyManager/"> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|--|---|

Operations are described in the corresponding modules throughout this section.

Each method of PMS corresponds to a method offered in a submodule. The name of the module corresponds exactly with the name of the “port type” in the WSDL definition above. For instance, if one method is offered by the port “Authorisation Support”, the description of the corresponding method can be found in the module Authorisation Support, and so on.

4.4 Protection Manager Support Domain Home (FUPF)

PMS Domain Home, provides protection functionalities into a Home Domain, for instance, at user's home, a kiosk with wireless access or a school, where users are consuming AXMEDIS objects. It does not allow license creation functionalities nor advanced licensing storage.

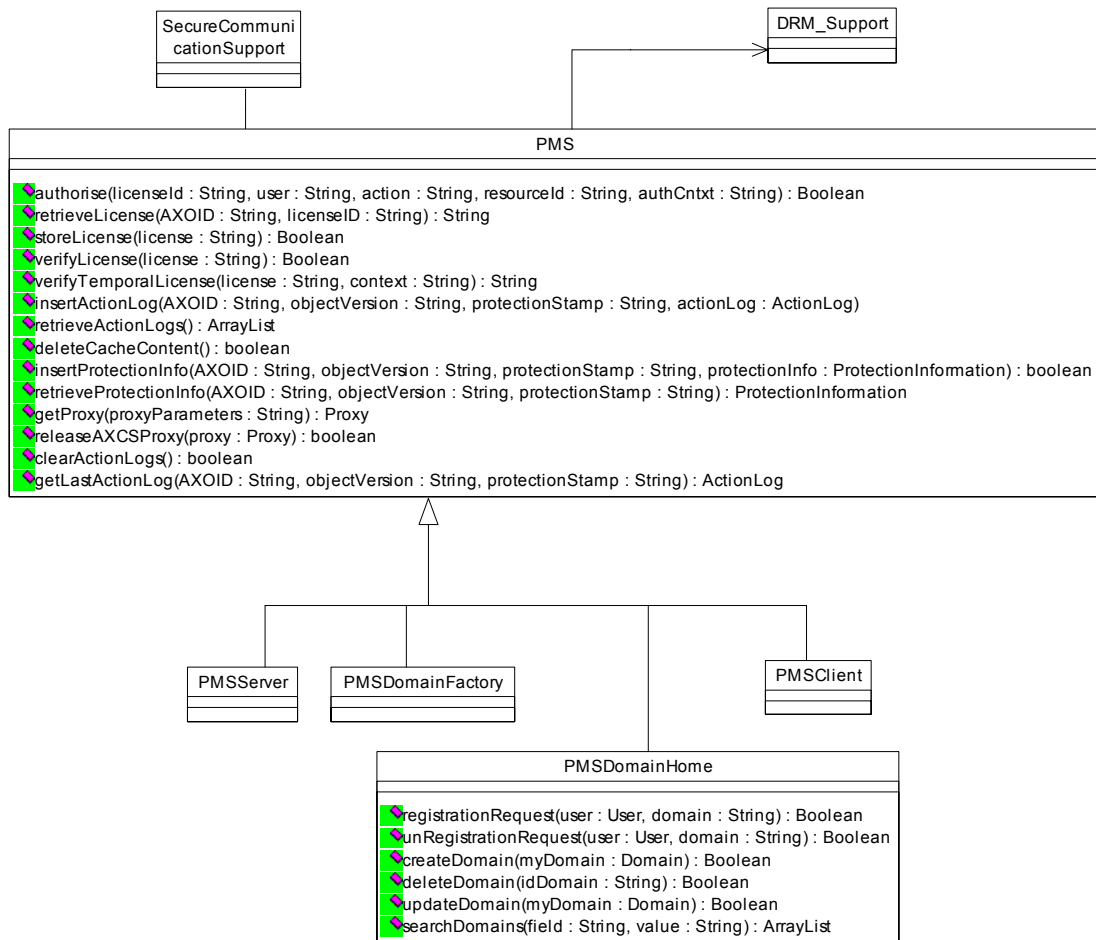


4.4.1 Module profile definition

| Module Profile | | |
|---|-----------------------------------|--|
| Protection Manager Support Domain Home | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java/C++ | |
| Responsible Name | | |
| Responsible Partner | | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | Any supported by Java | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Secure Communication Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.4.2 Architecture of PMS

Next figure shows the architecture of PMS domain home module.

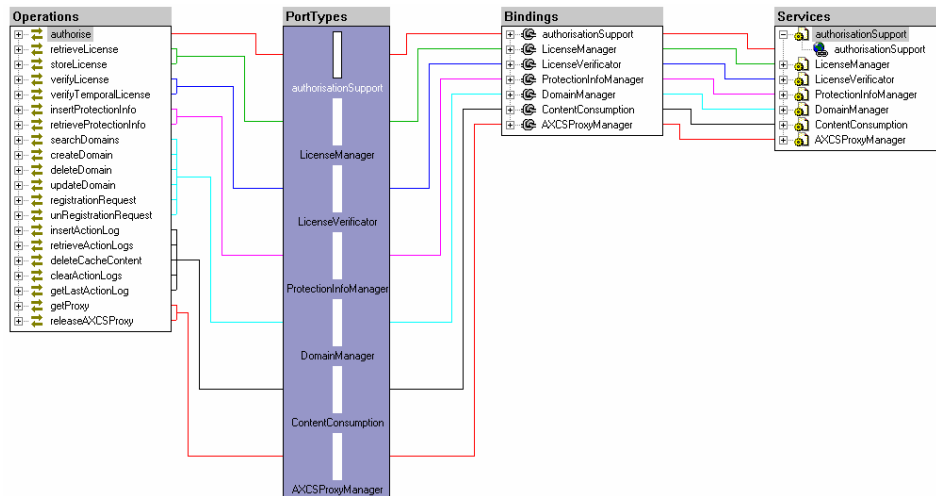


Class diagram of PMS Domain Home

| PMS Domain Home | |
|-----------------------|---|
| Methods | Description |
| authorise | This function authorises AXMEDIS users to perform actions over AXMEDIS objects. It proves that a user has the appropriate license that grants him to exercise a right over a resource if the conditions are met based on the execution context of the client. |
| retrieveLicense | This function retrieves the licenses stored in the license database. It retrieves the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| storeLicense | This function stores a license in the license database. |
| verifyLicense | Verifies a license syntactically against the schemas defined within the license. |
| verifyTemporalLicense | Verifies that the license generated by the user fulfils the initial desirables requirements of the user. For example, the user can verify that with this license he could exercise the desired action over the AXObject. |
| registrationRequest | This function is used to a registration of an user in a certain domain |
| unRegistrationRequest | This function is used to an unregistration of an user in a certain domain |
| createDomain | This function is used to create a new domain. |
| deleteDomain | This function is used to delete a domain with an idDomain identificator |

| | |
|------------------------|--|
| updateDomain | This function is used to update a domain. |
| searchDomains | This function is used to return a list of domains with a certain value in a certain field. |
| insertActionLog | Stores the given action log associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveActionLogs | This method retrieves the action logs stored in the local cache info. |
| deleteCacheContent | This method retrieves the action logs stored in the local cache info. |
| insertProtectionInfo | Stores the given protection information associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveProtectionInfo | This method retrieves the requested protection information. The information needed to retrieve protection information is the AXMEDIS object identifier, the object version and the protection stamp. |
| clearActionLogs | Deletes action logs from the cache, after positive authorisation of the user in the connected environment |
| getLastActionLog | Returns the last action log |
| getProxy | Returns a Proxy to establish a secure communication |
| releaseAXCSProxy | Finish the secure communication |

4.4.3 WSDL Interface



| PMS Domain Home | |
|-----------------|---|
| WSDL | <pre> <wsl:definitions xmlns:tns="urn:authorisationSupport" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:authorisationSupport" name="authorisationSupport"> <wsl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="action" type="xsd:string" nillable="true"/> <xsd:element name="resourceID" type="xsd:string" nillable="true"/> <xsd:element name="authContext" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsl:types> </pre> |

```

<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
    <xsd:element name="licenseID" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
    <xsd:element name="license" type="xsd:string" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="license" type="xsd:string" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="PARs" type="xsd:string" nillable="true"/>
    <xsd:element name="parentLicense" type="xsd:string" nillable="true"/>
    <xsd:element name="context" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="userIDs" type="xsd:string" nillable="true"/>
    <xsd:element name="rights" type="xsd:string" nillable="true"/>
    <xsd:element name="AXOIDs" type="xsd:string" nillable="true"/>
    <xsd:element name="conditions" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
    <xsd:element name="grants" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
    xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="ActionLog">
      <!-- Defined in AXMEDIS Supervisor -->
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/containers/literal/ms.net"
    targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
      <xsd:sequence>
        <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
      <xsd:complexContent mixed="false">
        <xsd:extension base="tns:List"/>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"
    xmlns:xns5="http://systinet.com/containers/literal/ms.net">
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:import namespace="http://systinet.com/wsdl/default"/>
    <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
    <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
    <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
    <xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
    <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
  </xsd:schema>

```

```

</wsdl:types>
<wsdl:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/containers/literal/ms.net"
    targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
      <xsd:sequence>
        <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Vector">
      <xsd:complexContent mixed="false">
        <xsd:extension base="tns:List"/>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="ArrayOfAnyType">
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"
    xmlns:xns5="http://systinet.com/containers/literal/ms.net">
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:import namespace="http://systinet.com/wsdl/default"/>
    <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
    <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
    <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
    <xsd:element name="protectionInfo" type="xns4:ProtectionInformation" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="ProtectionInformation_Response" type="xns4:ProtectionInformation"
      nillable="true"/>
    <xsd:element name="algorithm" type="xns4:Algorithm" nillable="true"/>
    <xsd:element name="keyLength" type="xsd:int"/>
    <xsd:element name="parameters" type="xns5:Vector" nillable="true"/>
    <xsd:element name="KeyAX_Response" type="xns4:KeyAX" nillable="true"/>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
    xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="ProtectionInformation">
      <xsd:sequence>
        <xsd:element name="protectionInfo" type="xsd:string" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Algorithm">
      <xsd:sequence>
        <xsd:element name="algorithm" type="xsd:int"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="KeyAX">
      <xsd:sequence>
        <xsd:element name="key" type="xsd:base64Binary" nillable="true"/>
        <xsd:element name="keyByte" type="xsd:base64Binary" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/>
    <xsd:element name="destinationLanguage" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
  <xsd:complexType name="List">
    <xsd:sequence>
      <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ArrayList">
    <xsd:complexContent mixed="false">
      <xsd:extension base="tns:List"/>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/containers/literal/ms.net"
xmlns:xns5="urn:DomainPackage.DomainManager">
  <xsd:import namespace="urn:DomainPackage.DomainManager"/>
  <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
  <xsd:element name="field" type="xsd:string" nillable="true"/>
  <xsd:element name="value" type="xsd:string" nillable="true"/>
  <xsd:element name="ArrayList_Response" type="xns4:ArrayList" nillable="true"/>
  <xsd:element name="myDomain" type="xns5:Domain" nillable="true"/>
  <xsd:element name="boolean_Response" type="xsd:boolean"/>
  <xsd:element name="idDomain" type="xsd:string" nillable="true"/>
  <xsd:element name="myUser" type="xns5:User" nillable="true"/>
  <xsd:element name="myDomain_1" type="xsd:string" nillable="true"/>
  <xsd:element name="userID" type="xsd:string" nillable="true"/>
  <xsd:element name="myDomain_2" type="xsd:string" nillable="true"/>
</xsd:schema>
<xsd:schema targetNamespace="urn:DomainPackage.DomainManager"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:DomainPackage.DomainManager" xmlns:map="http://systinet.com/mapping/">
  <xsd:complexType name="Domain">
    <xsd:sequence>
      <xsd:element name="AXDOM" type="xsd:string" nillable="true"/>
      <xsd:element name="AXID" type="xsd:string" nillable="true"/>
      <xsd:element name="typeOfID" type="xsd:string" nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="User">
    <xsd:sequence/>
  </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="AXCSPProxy">
      <xsd:sequence>
        <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/>
        <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/>
        <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ActionLog">
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="CertificationResult">
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="VerificationResult">
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"

```

```

<xsd:import namespace="http://systinet.com/wsdl/default"/>
<xsd:element name="AXCSProxy_Response" type="xns4:AXCSProxy" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
<xsd:schema targetNamespace="http://systinet.com/wsdl/default" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default"
xmlns:map="http://systinet.com/mapping"/>
<xsd:complexType name="ActionLog">
<xsd:annotation>
<xsd:appinfo>
<map:java-type name="ActionLog"/>
</xsd:appinfo>
</xsd:annotation>
<xsd:sequence/>
</xsd:complexType>
</xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
<xsd:complexType name="List">
<xsd:sequence>
<xsd:element name="Items" type="tns:ArrayOfAnyType"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayList">
<xsd:complexContent mixed="false">
<xsd:extension base="tns:List"/>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="LinkedList">
<xsd:complexContent mixed="false">
<xsd:extension base="tns:List"/>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Vector">
<xsd:complexContent mixed="false">
<xsd:extension base="tns:List"/>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Set">
<xsd:sequence>
<xsd:element name="Items" type="tns:ArrayOfAnyType"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="HashSet">
<xsd:complexContent mixed="false">
<xsd:extension base="tns:Set"/>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SortedSet">
<xsd:complexContent mixed="false">
<xsd:extension base="tns:Set"/>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TreeSet">
<xsd:complexContent mixed="false">
<xsd:extension base="tns:SortedSet"/>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Map">
<xsd:sequence>
<xsd:element name="Keys" type="tns:ArrayOfAnyType"/>
<xsd:element name="Values" type="tns:ArrayOfAnyType"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Hashtable">
<xsd:complexContent mixed="false">
<xsd:extension base="tns:Map"/>
</xsd:complexContent>

```



```

</xsd:complexType>
<xsd:complexType name="HashMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SortedMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TreeMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:SortedMap"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Iterator">
  <xsd:sequence>
    <xsd:element name="Data" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Enumeration">
  <xsd:sequence>
    <xsd:element name="Data" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfAnyType">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsd/default/"
  xmlns:xns5="http://systinet.com/containers/literal/ms.net">
  <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
  <xsd:import namespace="http://systinet.com/wsd/default"/>
  <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
  <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
  <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
  <xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
  <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
  <xsd:element name="boolean_Response" type="xsd:boolean"/>
  <xsd:element name="ActionLog_Response" type="xns4:ActionLog" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsd/default/" elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsd/default/"
    xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="AXCSPProxy">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="AXCSPProxy"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence>
        <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/>
        <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/>
        <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ActionLog">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="ActionLog"/>
        </xsd:appinfo>
      </xsd:annotation>
    </xsd:complexType>
  </xsd:schema>
</wsdl:types>

```

```

</xsd:complexType>
<xsd:complexType name="CertificationResult">
  <xsd:annotation>
    <xsd:appinfo>
      <map:java-type name="CertificationResult"/>
    </xsd:appinfo>
  </xsd:annotation>
</xsd:complexType>
<xsd:complexType name="VerificationResult">
  <xsd:annotation>
    <xsd:appinfo>
      <map:java-type name="VerificationResult"/>
    </xsd:appinfo>
  </xsd:annotation>
</xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wSDL/default/">
  <xsd:import namespace="http://systinet.com/wSDL/default/">
    <xsd:element name="proxyParameters" type="xsd:string" nillable="true"/>
    <xsd:element name="AXCSProxy_Response" type="xns4:AXCSProxy" nillable="true"/>
    <xsd:element name="proxy" type="xns4:AXCSProxy" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="authorisationSupport_authorise_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="authorisationSupport_authorise_1_Request">
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="action" element="ns0:action"/>
  <wsdl:part name="resourceID" element="ns0:resourceID"/>
  <wsdl:part name="authContext" element="ns0:authContext"/>
</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVericator_verifyTemporalLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="context" element="ns0:context"/>
</wsdl:message>
<wsdl:message name="LicenseVericator_verifyCreatedLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVericator_verifyCreatedLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>

```

| | |
|--|--|
| | <pre> <wsdl:message name="LicenseVerificator_verifyTemporalLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="grants" element="ns0:grants"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/> <wsdl:message name="ContentConsumption_retrieveActionLogs_Response"> <wsdl:part name="response" element="ns0:ArrayList_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_insertActionLog_Response"/> <wsdl:message name="ContentConsumption_insertActionLog_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="objectVersion" element="ns0:objectVersion"/> <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/> <wsdl:part name="actionLog" element="ns0:actionLog"/> </wsdl:message> <wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"/> <wsdl:message name="ContentConsumption_deleteCacheContent_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="ProtectionInfoManager_insertProtectionInfo_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="objectVersion" element="ns0:objectVersion"/> <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/> <wsdl:part name="protectionInfo" element="ns0:protectionInfo"/> </wsdl:message> <wsdl:message name="ProtectionInfoManager_insertProtectionInfo_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="objectVersion" element="ns0:objectVersion"/> <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/> </wsdl:message> <wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_Response"> <wsdl:part name="response" element="ns0:ProtectionInformation_Response"/> </wsdl:message> <wsdl:message name="ProtectionInfoManager_generateKey_1_Request"> <wsdl:part name="algorithm" element="ns0:algorithm"/> <wsdl:part name="keyLength" element="ns0:keyLength"/> <wsdl:part name="parameters" element="ns0:parameters"/> </wsdl:message> <wsdl:message name="ProtectionInfoManager_generateKey_Response"> <wsdl:part name="response" element="ns0:KeyAX_Response"/> </wsdl:message> <wsdl:message name="RightsExpressionTranslator_generateTranslation_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </pre> |
|--|--|

```

</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_1_Request">
  <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/>
  <wsdl:part name="destinationLanguage" element="ns0:destinationLanguage"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_1_Request">
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_2"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_1_Request">
  <wsdl:part name="idDomain" element="ns0:idDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_1_Request">
  <wsdl:part name="myUser" element="ns0:myUser"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_1"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_1_Request">
  <wsdl:part name="field" element="ns0:field"/>
  <wsdl:part name="value" element="ns0:value"/>
</wsdl:message>
<wsdl:message name="AXCSProxy_getProxy_1_Request"/>
<wsdl:message name="AXCSProxy_getProxy_Response">
  <wsdl:part name="response" element="ns0:AXCSProxy_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_clearActionLogs_1_Request"/>
<wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/>
<wsdl:message name="ContentConsumption_getLastActionLog_Response">
  <wsdl:part name="response" element="ns0:ActionLog_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_getLastActionLog_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"/>
<wsdl:message name="ContentConsumption_retrieveActionLogs_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_insertActionLog_Response"/>
<wsdl:message name="ContentConsumption_insertActionLog_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>

```

```

<wsdl:part name="actionLog" element="ns0:actionLog"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_clearActionLogs_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_releaseAXCSProxy_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_releaseAXCSProxy_1_Request">
  <wsdl:part name="proxy" element="ns0:proxy"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_getProxy_Response">
  <wsdl:part name="response" element="ns0:AXCSProxy_Response"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_getProxy_1_Request">
  <wsdl:part name="proxyParameters" element="ns0:proxyParameters"/>
</wsdl:message>
<wsdl:portType name="authorisationSupport">
  <wsdl:operation name="authorise" parameterOrder="licenseID userID action resourceID authContext">
    <wsdl:input message="tns:authorisationSupport_authorise_1_Request"/>
    <wsdl:output message="tns:authorisationSupport_authorise_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseManager">
  <wsdl:operation name="retrieveLicense" parameterOrder="AXOID licenseID">
    <wsdl:input message="tns:LicenseManager_retrieveLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_retrieveLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseManager_storeLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_storeLicense_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseVerificator">
  <wsdl:operation name="verifyLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseVerificator_verifyLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerificator_verifyLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyTemporalLicense" parameterOrder="license context">
    <wsdl:input message="tns:LicenseVerificator_verifyTemporalLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerificator_verifyTemporalLicense_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="ProtectionInfoManager">
  <wsdl:operation name="insertProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp protectionInfo">
    <wsdl:input message="tns:ProtectionInfoManager_insertProtectionInfo_1_Request"/>
    <wsdl:output message="tns:ProtectionInfoManager_insertProtectionInfo_Response"/>
  </wsdl:operation>
  <wsdl:operation name="retrieveProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp">
    <wsdl:input message="tns:ProtectionInfoManager_retrieveProtectionInfo_1_Request"/>
    <wsdl:output message="tns:ProtectionInfoManager_retrieveProtectionInfo_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="DomainManager">
  <wsdl:operation name="searchDomains" parameterOrder="field value">
    <wsdl:input message="tns:DomainManager_searchDomains_1_Request"/>
    <wsdl:output message="tns:DomainManager_searchDomains_Response"/>
  </wsdl:operation>
  <wsdl:operation name="createDomain" parameterOrder="myDomain">
    <wsdl:input message="tns:DomainManager_createDomain_1_Request"/>
    <wsdl:output message="tns:DomainManager_createDomain_Response"/>
  </wsdl:operation>
  <wsdl:operation name="deleteDomain" parameterOrder="idDomain">
    <wsdl:input message="tns:DomainManager_deleteDomain_1_Request"/>
    <wsdl:output message="tns:DomainManager_deleteDomain_Response"/>
  </wsdl:operation>
  <wsdl:operation name="updateDomain" parameterOrder="myDomain">
    <wsdl:input message="tns:DomainManager_updateDomain_1_Request"/>
  </wsdl:operation>

```



```

<wsdl:output message="tns:DomainManager_updateDomain_Response"/>
</wsdl:operation>
<wsdl:operation name="registrationRequest" parameterOrder="myUser myDomain">
    <wsdl:input message="tns:DomainManager_registrationRequest_1_Request"/>
    <wsdl:output message="tns:DomainManager_registrationRequest_Response"/>
</wsdl:operation>
<wsdl:operation name="unRegistrationRequest" parameterOrder="userID myDomain">
    <wsdl:input message="tns:DomainManager_unRegistrationRequest_1_Request"/>
    <wsdl:output message="tns:DomainManager_unRegistrationRequest_Response"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="ContentConsumption">
    <wsdl:operation name="insertActionLog" parameterOrder="AXOID objectVersion protectionStamp actionLog">
        <wsdl:input message="tns:ContentConsumption_insertActionLog_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_insertActionLog_Response"/>
    </wsdl:operation>
    <wsdl:operation name="retrieveActionLogs">
        <wsdl:input message="tns:ContentConsumption_retrieveActionLogs_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_retrieveActionLogs_Response"/>
    </wsdl:operation>
    <wsdl:operation name="deleteCacheContent">
        <wsdl:input message="tns:ContentConsumption_deleteCacheContent_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_deleteCacheContent_Response"/>
    </wsdl:operation>
    <wsdl:operation name="clearActionLogs">
        <wsdl:input message="tns:ContentConsumption_clearActionLogs_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_clearActionLogs_Response"/>
    </wsdl:operation>
    <wsdl:operation name="getLastActionLog" parameterOrder="AXOID objectVersion protectionStamp">
        <wsdl:input message="tns:ContentConsumption_getLastActionLog_1_Request"/>
        <wsdl:output message="tns:ContentConsumption_getLastActionLog_Response"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="AXCSProxyManager">
    <wsdl:operation name="getProxy" parameterOrder="proxyParameters">
        <wsdl:input message="tns:AXCSProxyManager_getProxy_1_Request"/>
        <wsdl:output message="tns:AXCSProxyManager_getProxy_Response"/>
    </wsdl:operation>
    <wsdl:operation name="releaseAXCSProxy" parameterOrder="proxy">
        <wsdl:input message="tns:AXCSProxyManager_releaseAXCSProxy_1_Request"/>
        <wsdl:output message="tns:AXCSProxyManager_releaseAXCSProxy_Response"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="authorisationSupport" type="tns:authorisationSupport">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http/">
    <wsdl:operation name="authorise">
        <map:java-operation name="authorise">
signature="KEXqYXZhL2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nOyla"/>
        <soap:operation
soapAction="urn:authorisationSupportauthorisationSupport#authorise#KEXqYXZhL2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nOyla" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="LicenseManager" type="tns:LicenseManager">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http/">
    <wsdl:operation name="retrievelicense">
        <map:java-operation name="retrievelicense">
signature="KEXqYXZhL2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmllZzs="/>
        <soap:operation
soapAction="urn:licenseManagerlicenseManager#retrievelicense#KEXqYXZhL2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nO0xqYXZl2xhbmcvU3RyaW5nOyla" style="document"/>

```

```

hL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="storeLicense">
    <map:java-operation name="storeLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/>
    <soap:operation
      soapAction="urn:LicenseManagerLicenseManager#storeLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="LicenseVerifier" type="tns:LicenseVerifier">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="verifyLicense">
    <map:java-operation name="verifyLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/>
    <soap:operation
      soapAction="urn:LicenseVerifierLicenseVerifier#verifyLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="verifyTemporalLicense">
    <map:java-operation name="verifyTemporalLicense"
      signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmlu
      Zzs="/>
    <soap:operation
      soapAction="urn:LicenseVerifierLicenseVerifier#verifyTemporalLicense#KExqYXZhL2xhbmcvU3RyaW5
      nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmluZzs=" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ProtectionInfoManager" type="tns:ProtectionInfoManager">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="insertProtectionInfo">
    <map:java-operation name="insertProtectionInfo"
      signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n
      O0xQcm90ZWNoaW9uSW5mb3JtYXRpb247KVo="/>
    <soap:operation
      soapAction="urn:ProtectionInfoManagerProtectionInfoManager#insertProtectionInfo#KExqYXZhL2xhbmcvU3R
      yaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWNoaW9uSW5mb3JtYX
      Rpb247KVo=" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="retrieveProtectionInfo">
    <map:java-operation name="retrieveProtectionInfo"
      signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n

```

| | |
|--|---|
| | <pre> OylMUHJvdGVjdGlvbkluZm9ybWFOaW9uOw=="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#retrieveProtectionInfo#KExqYXZhL2xhbmcvU3 RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMUHJvdGVjdGlvbkluZm9ybWFO aW9uOw==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="DomainManager" type="tns:DomainManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="searchDomains"> <map:java-operation name="searchDomains" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5T GlzdDs="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#searchDomains#KExqYXZhL2xhbmcvU3 RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGlzdDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="createDomain"> <map:java-operation name="createDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWVpbjpwWg=="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#createDomain#KExEb21haW5QYWNRyW dlL0RvbWVpbjpwWg==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteDomain"> <map:java-operation name="deleteDomain" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#deleteDomain#KExqYXZhL2xhbmcvU3R yaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="updateDomain"> <map:java-operation name="updateDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWVpbjpwWg=="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#updateDomain#KExEb21haW5QYWNRy WdlL0RvbWVpbjpwWg==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="registrationRequest"> <map:java-operation name="registrationRequest" signature="KExEb21haW5QYWNRyWdlL1VzZXI7TGphdmEvbGFuZy9TdHJpbmc7KVo="/> </pre> |
|--|---|

| | |
|--|--|
| | <pre> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#registrationRequest#KEExEb21haW5QYW NrYWdlL1VzZXI7TGphdmEvdGFuZy9TdHJpbmc7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="unRegistrationRequest"> <map:java-operation name="unRegistrationRequest" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#unRegistrationRequest#KExqYXZhL2xhb mcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="ContentConsumption" type="tns:ContentConsumption"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="insertActionLog"> <map:java-operation name="insertActionLog" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xBy3Rpb25Mb2c7KVY="/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#insertActionLog#KExqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xBy3Rpb25Mb2c7KVY=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveActionLogs"> <map:java-operation name="retrieveActionLogs" signature="KCIMamF2YS91dGlsL0FycmF5TGldDs="/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#retrieveActionLogs#KCIMamF2YS91dGlsL0Fycm F5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteCacheContent"> <map:java-operation name="deleteCacheContent" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#deleteCacheContent#KClA" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="clearActionLogs"> <map:java-operation name="clearActionLogs" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#clearActionLogs#KClA" style="document"/> <wsdl:input> </pre> |
|--|--|

```

        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getLastActionLog">
    <map:java-operation name="getLastActionLog"
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMQWN0aW9uTG9nOw==" />
    <soap:operation
soapAction="urn:ContentConsumptionContentConsumption#getLastActionLog#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMQWN0aW9uTG9nOw=="
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="AXCSProxyManager" type="tns:AXCSProxyManager">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getProxy">
        <map:java-operation name="getProxy"
signature="KExqYXZhL2xhbmcvU3RyaW5nOylMQVhDU1Byb3h5Ow==" />
        <soap:operation
soapAction="urn:AXCSProxyManagerAXCSProxyManager#getProxy#KExqYXZhL2xhbmcvU3RyaW5nOylMQVhDU1Byb3h5Ow==" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="releaseAXCSProxy">
        <map:java-operation name="releaseAXCSProxy" signature="KExBWENTUHJveHk7KVo=" />
        <soap:operation
soapAction="urn:AXCSProxyManagerAXCSProxyManager#releaseAXCSProxy#KExBWENTUHJveHk7KVo="
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="authorisationSupport">
    <wsdl:port name="authorisationSupport" binding="tns:authorisationSupport">
        <soap:address location="http://cjlabs1:6060/authorisationSupport"/>
    </wsdl:port>
</wsdl:service>
<wsdl:service name="LicenseManager">
    <wsdl:port name="LicenseManager" binding="tns:LicenseManager">
        <soap:address location="http://cjlabs1:6060/LicenseManager"/>
    </wsdl:port>
</wsdl:service>
<wsdl:service name="LicenseVerifier">
    <wsdl:port name="LicenseVerifier" binding="tns:LicenseVerifier">
        <soap:address location="http://cjlabs1:6060/LicenseVerifier"/>
    </wsdl:port>
</wsdl:service>
<wsdl:service name="ProtectionInfoManager">
    <wsdl:port name="ProtectionInfoManager" binding="tns:ProtectionInfoManager">
        <soap:address location="http://efpc037:6060/ProtectionInfoManager"/>
    </wsdl:port>
</wsdl:service>

```

| | |
|--|--|
| | <pre> <wsdl:service name="DomainManager"> <wsdl:port name="DomainManager" binding="tns:DomainManager"> <soap:address location="http://EFPC016:6060/DomainManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="ContentConsumption"> <wsdl:port name="ContentConsumption" binding="tns:ContentConsumption"> <soap:address location="http://efpc037:6060/ContentConsumption/"> </wsdl:port> </wsdl:service> <wsdl:service name="AXCSProxyManager"> <wsdl:port name="AXCSProxyManager" binding="tns:AXCSProxyManager"> <soap:address location="http://cjlal1:6060/AXCSProxyManager/"> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|--|--|

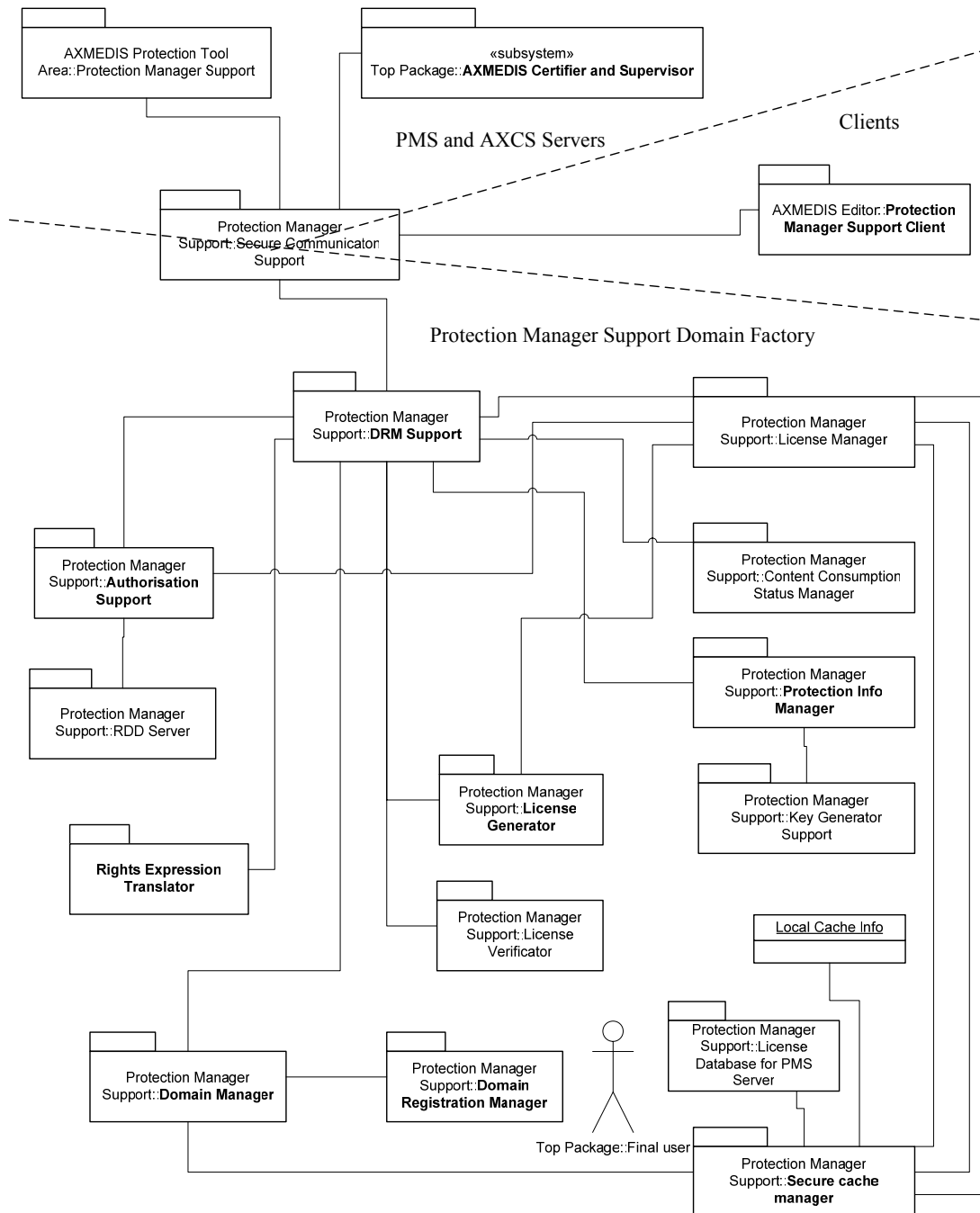
Operations are described in the corresponding modules throughout this section.

Each method of PMS corresponds to a method offered in a submodule. The name of the module corresponds exactly with the name of the “port type” in the WSDL definition above. For instance, if one method is offered by the port “Authorisation Support”, the description of the corresponding method can be found in the module Authorisation Support, and so on.

4.5 Protection Manager Support Domain Factory (FUPF)

PMS Domain Factory is thought for AXMEDIS Objects factories, like an editorial or a digital factory, which have creation and distribution functionalities. In this PMS, it is possible to create licenses, according to the ones owned by the Factory. Nevertheless, in the end, all the actions over objects and licenses should be checked by the PMS Server of reference, which is in contact with the AXMEDIS Certifier and Supervisor.

Protection Manager Support Domain Factory

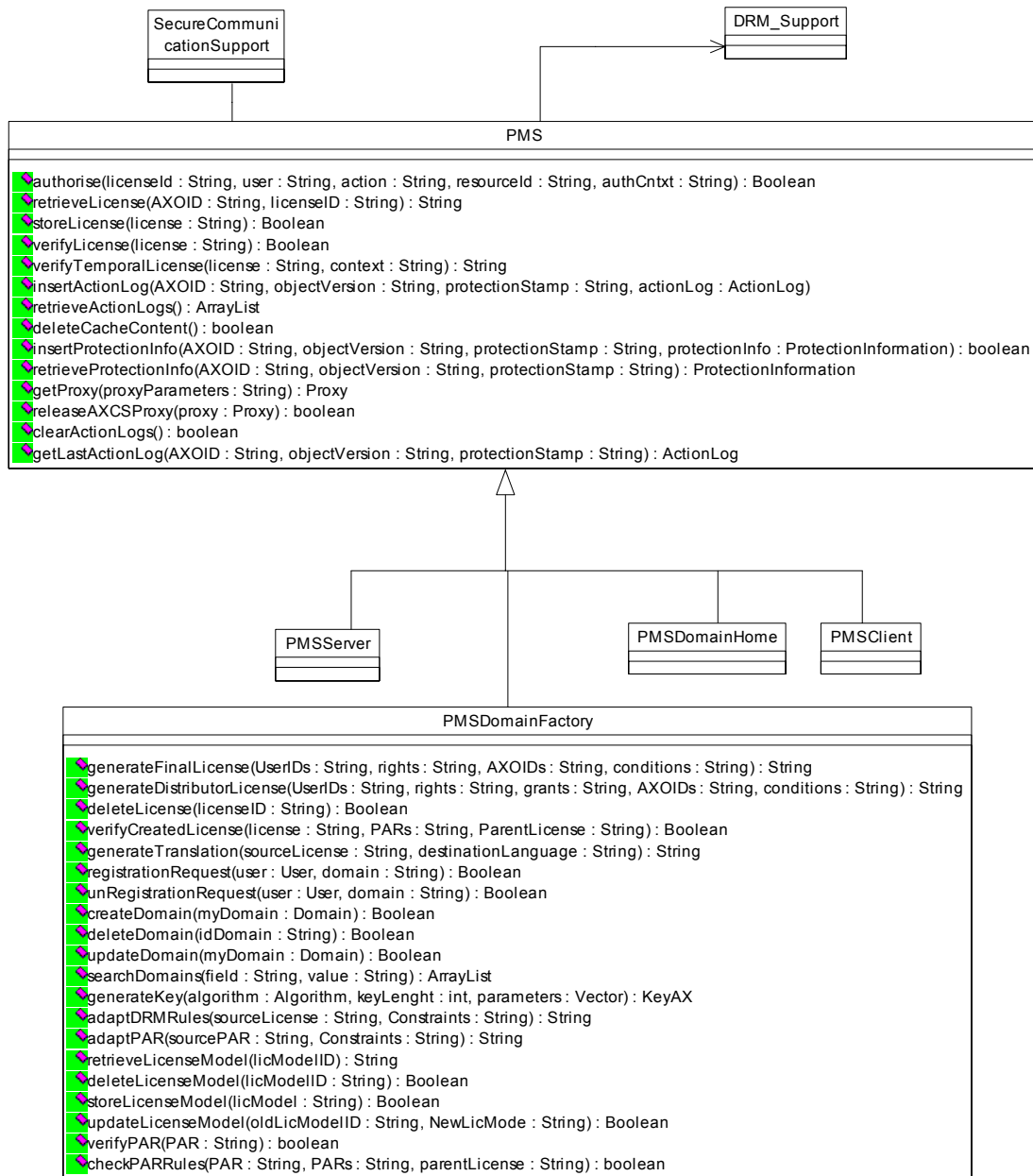


4.5.1 Module profile definition

| Module Profile | | |
|--|-----------------------------------|--|
| Protection Manager Support Domain Factory | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java/C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Secure Communication Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.5.2 Architecture of PMS

Next figure shows the architecture of PMS domain factory module.



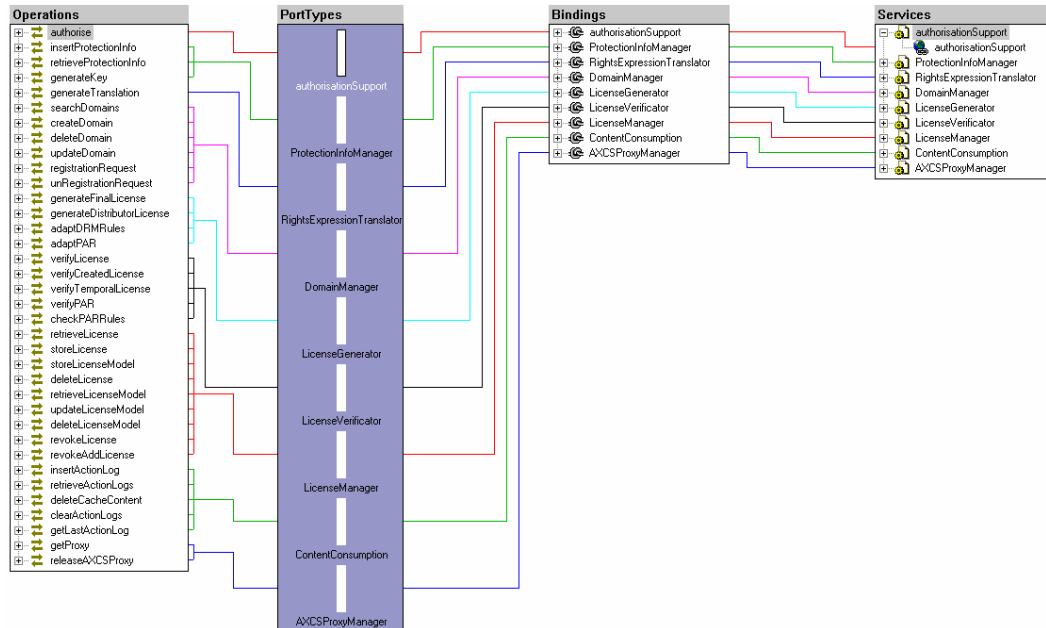
Class diagram of PMS Domain Factory

| PMS Domain Factory | |
|----------------------|--|
| Methods | Description |
| generateFinalLicense | Generates a license for an end-user, according to the userID, right, AXOID |

| | |
|----------------------------|---|
| | and conditions. |
| generateDistributorLicense | Generates a license for a distributor, according to the userID, right, grants that this user will be able to distribute and conditions. |
| authorise | This function authorises AXMEDIS users to perform actions over AXMEDIS objects. It proves that a user has the appropriate license that grants him to exercise a right over a resource if the conditions are met based on the execution context of the client. |
| retrieveLicense | This function retrieves the licenses stored in the license database. It retrieves the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| deleteLicense | This function deletes the licenses stored in the license database. It deletes the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| storeLicense | This function stores a license in the license database. |
| verifyLicense | Verifies a license syntactically against the schemas defined within the license. |
| verifyCreatedLicense | Verifies if the license can be generated according to the PARs and the parent licenses (e.g. Distributor or Creator licenses). |
| verifyTemporalLicense | Verifies that the license generated by the user fulfils the initial desirables requirements of the user. For example, the user can verify that with this license he could exercise the desired action over the AXObject. |
| generateTranslation | This function is used to return a destinationLicense. The system makes a translation from a sourceLicense to a destinationLicense. The destinationLicense, if the translation is possible, will be written in a languageDestination. |
| registrationRequest | This function is used to a registration of an user in a certain domain |
| unRegistrationRequest | This function is used to an unregistration of an user in a certain domain |
| createDomain | This function is used to create a new domain. |
| DeleteDomain | This function is used to delete a domain with an idDomain identificator |
| UpdateDomain | This function is used to update a domain. |
| searchDomains | This function is used to return a list of domains with a certain value in a certain field. |
| insertActionLog | Stores the given action log associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveActionLogs | This method retrieves the action logs stored in the local cache info. |
| deleteCacheContent | This method retrieves the action logs stored in the local cache info. |
| insertProtectionInfo | Stores the given protection information associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveProtectionInfo | This method retrieves the requested protection information. The information needed to retrieve protection information is the AXMEDIS object identifier, the object version and the protection stamp. |
| generateKey | This method permits the creation of a key for protecting an AXMEDIS object. |
| adaptDRMRules | Adapts the given license using the given constraints |
| adaptPAR | Adapts the given PAR using the given constraints |
| verifyPAR | Verifies a PAR syntactically against the schemas defined within the PAR. |
| checkPARRules | Verifies if the PAR can be generated according to the parent licenses and PARs. |
| retrieveLicenseModel | This function returns a licenseModel stored in the License DB |
| deleteLicenseModel | This function deletes a licenseModel from the License DB |
| storeLicenseModel | This function inserts a license model in the License DB |
| updateLicenseModel | This function updates the licenseModel given a licenseModelID and a new description of licenseModel |
| revokeLicense | This function changes the status of a license to revoked in the License DB |

| | |
|------------------|---|
| revokeAddLicense | This function changes the status of a license to revoked and adds the license that substitutes it in the License DB |
| clearActionLogs | Deletes action logs from the cache, after positive authorisation of the user in the connected environment |
| getLastActionLog | Returns the last action log |
| getProxy | Returns a Proxy to establish a secure communication |
| releaseAXCSProxy | Finish the secure communication |

4.5.3 WSDL Interface



| PMS Domain Factory | |
|--------------------|---|
| WSDL | <pre> <wsl:definitions xmlns:tns="urn:authorisationSupport" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:authorisationSupport" name="authorisationSupport"> <wsl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="action" type="xsd:string" nillable="true"/> <xsd:element name="resourceID" type="xsd:string" nillable="true"/> <xsd:element name="authContext" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsl:types> <wsl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> <xsd:element name="license" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsl:types> </pre> |


```

</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="license" type="xsd:string" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="PARs" type="xsd:string" nillable="true"/>
    <xsd:element name="parentLicense" type="xsd:string" nillable="true"/>
    <xsd:element name="context" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="userIds" type="xsd:string" nillable="true"/>
    <xsd:element name="rights" type="xsd:string" nillable="true"/>
    <xsd:element name="AXOIDs" type="xsd:string" nillable="true"/>
    <xsd:element name="conditions" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
    <xsd:element name="grants" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
    xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="ActionLog">
      <!-- Defined in AXMEDIS Supervisor -->
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/containers/literal/ms.net"
    targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
      <xsd:sequence>
        <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
      <xsd:complexContent mixed="false">
        <xsd:extension base="tns:List"/>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"
    xmlns:xns5="http://systinet.com/containers/literal/ms.net">
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:import namespace="http://systinet.com/wsdl/default"/>
    <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
    <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
    <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
    <xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
    <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/containers/literal/ms.net"
    targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
      <xsd:sequence>
        <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>

```

| | |
|--|---|
| | <pre> </xsd:complexType> <xsd:complexType name="Vector"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="ArrayOfAnyType"> <xsd:sequence> <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/" xmlns:xns5="http://systinet.com/containers/literal/ms.net"> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:import namespace="http://systinet.com/wsdl/default"/> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="objectVersion" type="xsd:string" nillable="true"/> <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/> <xsd:element name="protectionInfo" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="ProtectionInformation_Response" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="algorithm" type="xns4:Algorithm" nillable="true"/> <xsd:element name="keyLength" type="xsd:int"/> <xsd:element name="parameters" type="xns5:Vector" nillable="true"/> <xsd:element name="KeyAX_Response" type="xns4:KeyAX" nillable="true"/> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="ProtectionInformation"> <xsd:sequence> <xsd:element name="protectionInfo" type="xsd:string" nillable="true"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="Algorithm"> <xsd:sequence> <xsd:element name="algorithm" type="xsd:int"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="KeyAX"> <xsd:sequence> <xsd:element name="key" type="xsd:base64Binary" nillable="true"/> <xsd:element name="keyByte" type="xsd:base64Binary" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/> <xsd:element name="destinationLanguage" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayList"> </pre> |
|--|---|

| | |
|--|--|
| | <pre> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/containers/literal/ms.net" xmlns:xns5="urn:DomainPackage.DomainManager"> <xsd:import namespace="urn:DomainPackage.DomainManager"/> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:element name="field" type="xsd:string" nillable="true"/> <xsd:element name="value" type="xsd:string" nillable="true"/> <xsd:element name="ArrayList_Response" type="xns4:ArrayList" nillable="true"/> <xsd:element name="myDomain" type="xns5:Domain" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="idDomain" type="xsd:string" nillable="true"/> <xsd:element name="myUser" type="xns5:User" nillable="true"/> <xsd:element name="myDomain_1" type="xsd:string" nillable="true"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="myDomain_2" type="xsd:string" nillable="true"/> </xsd:schema> <xsd:schema targetNamespace="urn:DomainPackage.DomainManager" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:DomainPackage.DomainManager" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="Domain"> <xsd:sequence> <xsd:element name="AXDOM" type="xsd:string" nillable="true"/> <xsd:element name="AXID" type="xsd:string" nillable="true"/> <xsd:element name="typeOfID" type="xsd:string" nillable="true"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="User"> <xsd:sequence/> </xsd:complexType> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="AXCSPProxy"> <xsd:sequence> <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/> <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/> <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ActionLog"> <xsd:sequence/> </xsd:complexType> <xsd:complexType name="CertificationResult"> <xsd:sequence/> </xsd:complexType> <xsd:complexType name="VerificationResult"> <xsd:sequence/> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"> <xsd:import namespace="http://systinet.com/wsdl/default/"> <xsd:element name="AXCSPProxy_Response" type="xns4:AXCSPProxy" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="userIDs" type="xsd:string" nillable="true"/> </pre> |
|--|--|

```

<xsd:element name="rights" type="xsd:string" nillable="true"/>
<xsd:element name="AXOIDs" type="xsd:string" nillable="true"/>
<xsd:element name="conditions" type="xsd:string" nillable="true"/>
<xsd:element name="string_Response" type="xsd:string" nillable="true"/>
<xsd:element name="grants" type="xsd:string" nillable="true"/>
<xsd:element name="sourceLicense" type="xsd:string" nillable="true"/>
<xsd:element name="constraints" type="xsd:string" nillable="true"/>
<xsd:element name="PAR" type="xsd:string" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="license" type="xsd:string" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="PARs" type="xsd:string" nillable="true"/>
    <xsd:element name="parentLicense" type="xsd:string" nillable="true"/>
    <xsd:element name="context" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
    <xsd:element name="PAR" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
    <xsd:element name="licenseID" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
    <xsd:element name="license" type="xsd:string" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="licenseModel" type="xsd:string" nillable="true"/>
    <xsd:element name="licModelID" type="xsd:string" nillable="true"/>
    <xsd:element name="licModelNew" type="xsd:string" nillable="true"/>
    <xsd:element name="newLicense" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
    xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="ActionLog">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="ActionLog"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/containers/literal/ms.net"
    targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
      <xsd:sequence>
        <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
      <xsd:complexContent mixed="false">
        <xsd:extension base="tns:List"/>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="LinkedList">
      <xsd:complexContent mixed="false">
        <xsd:extension base="tns:List"/>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="Vector">

```

```

<xsd:complexContent mixed="false">
  <xsd:extension base="tns:List"/>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Set">
  <xsd:sequence>
    <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="HashSet">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Set"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SortedSet">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Set"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TreeSet">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:SortedSet"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Map">
  <xsd:sequence>
    <xsd:element name="Keys" type="tns:ArrayOfAnyType"/>
    <xsd:element name="Values" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Hashtable">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="HashMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SortedMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TreeMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:SortedMap"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Iterator">
  <xsd:sequence>
    <xsd:element name="Data" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Enumeration">
  <xsd:sequence>
    <xsd:element name="Data" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfAnyType">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wSDL/default/"
  xmlns:xns5="http://systinet.com/containers/literal/ms.net">

```

```

<xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
<xsd:import namespace="http://systinet.com/wsdl/default"/>
<xsd:element name="AXOID" type="xsd:string" nillable="true"/>
<xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
<xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
<xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
<xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
<xsd:element name="boolean_Response" type="xsd:boolean"/>
<xsd:element name="ActionLog_Response" type="xns4:ActionLog" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default"
xmlns:map="http://systinet.com/mapping"/>
    <xsd:complexType name="AXCSPProxy">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="AXCSPProxy"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence>
        <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/>
        <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/>
        <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ActionLog">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="ActionLog"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="CertificationResult">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="CertificationResult"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="VerificationResult">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="VerificationResult"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default"/>
    <xsd:import namespace="http://systinet.com/wsdl/default"/>
    <xsd:element name="proxyParameters" type="xsd:string" nillable="true"/>
    <xsd:element name="AXCSPProxy_Response" type="xns4:AXCSPProxy" nillable="true"/>
    <xsd:element name="proxy" type="xns4:AXCSPProxy" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="authorisationSupport_authorise_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="authorisationSupport_authorise_1_Request">
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="action" element="ns0:action"/>
  <wsdl:part name="resourceID" element="ns0:resourceID"/>

```



```

<wsdl:part name="authContext" element="ns0:authContext"/>
</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyTemporalLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="context" element="ns0:context"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyCreatedLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyCreatedLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyTemporalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>
  <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
  <wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>
  <wsdl:part name="grants" element="ns0:grants"/>
  <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
  <wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/>
<wsdl:message name="ContentConsumption_retrieveActionLogs_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_insertActionLog_Response"/>
<wsdl:message name="ContentConsumption_insertActionLog_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>

```

```

<wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
<wsdl:part name="actionLog" element="ns0:actionLog"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_1_Request">
<wsdl:message name="ContentConsumption_deleteCacheContent_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
  <wsdl:part name="protectionInfo" element="ns0:protectionInfo"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:ProtectionInformation_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_1_Request">
  <wsdl:part name="algorithm" element="ns0:algorithm"/>
  <wsdl:part name="keyLength" element="ns0:keyLength"/>
  <wsdl:part name="parameters" element="ns0:parameters"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_Response">
  <wsdl:part name="response" element="ns0:KeyAX_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_1_Request">
  <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/>
  <wsdl:part name="destinationLanguage" element="ns0:destinationLanguage"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_1_Request">
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_2"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_1_Request">
  <wsdl:part name="idDomain" element="ns0:idDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>

```



```

<wsdl:message name="DomainManager_registrationRequest_1_Request">
  <wsdl:part name="myUser" element="ns0:myUser"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_1"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_1_Request">
  <wsdl:part name="field" element="ns0:field"/>
  <wsdl:part name="value" element="ns0:value"/>
</wsdl:message>
<wsdl:message name="AXCSProxy_getProxy_1_Request"/>
<wsdl:message name="AXCSProxy_getProxy_Response">
  <wsdl:part name="response" element="ns0:AXCSProxy_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_adaptPAR_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_adaptDRMRules_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_adaptPAR_1_Request">
  <wsdl:part name="PAR" element="ns0:PAR"/>
  <wsdl:part name="constraints" element="ns0:constraints"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>
  <wsdl:part name="grants" element="ns0:grants"/>
  <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
  <wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_adaptDRMRules_1_Request">
  <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/>
  <wsdl:part name="constraints" element="ns0:constraints"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>
  <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
  <wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyCreatedLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_checkPARRules_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_checkPARRules_1_Request">
  <wsdl:part name="PAR" element="ns0:PAR"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyCreatedLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyPAR_1_Request">
  <wsdl:part name="PAR" element="ns0:PAR"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyTemporalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>

```

| | |
|--|---|
| | <pre> <wsdl:message name="LicenseVerifier_verifyPAR_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVerifier_verifyTemporalLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> <wsdl:part name="context" element="ns0:context"/> </wsdl:message> <wsdl:message name="LicenseVerifier_verifyLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeLicense_1_Request"> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicenseModel_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_updateLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> <wsdl:part name="licModelNew" element="ns0:licModelNew"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeAddLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeAddLicense_1_Request"> <wsdl:part name="licenseID" element="ns0:licenseID"/> <wsdl:part name="newLicense" element="ns0:newLicense"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicenseModel_1_Request"> <wsdl:part name="licenseModel" element="ns0:licenseModel"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> </wsdl:message> <wsdl:message name="LicenseManager_updateLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </wsdl:message> <wsdl:message name="ContentConsumption_clearActionLogs_1_Request"> <wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/> </pre> |
|--|---|

| | |
|--|--|
| | <pre> <wsdl:message name="ContentConsumption_getLastActionLog_Response"> <wsdl:part name="response" element="ns0:ActionLog_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_getLastActionLog_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="objectVersion" element="ns0:objectVersion"/> <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/> </wsdl:message> <wsdl:message name="ContentConsumption_deleteCacheContent_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"> </wsdl:message> <wsdl:message name="ContentConsumption_retrieveActionLogs_Response"> <wsdl:part name="response" element="ns0:ArrayList_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_insertActionLog_Response"> </wsdl:message> <wsdl:message name="ContentConsumption_insertActionLog_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="objectVersion" element="ns0:objectVersion"/> <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/> <wsdl:part name="actionLog" element="ns0:actionLog"/> </wsdl:message> <wsdl:message name="ContentConsumption_clearActionLogs_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_releaseAXCSProxy_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_releaseAXCSProxy_1_Request"> <wsdl:part name="proxy" element="ns0:proxy"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_getProxy_Response"> <wsdl:part name="response" element="ns0:AXCSProxy_Response"/> </wsdl:message> <wsdl:message name="AXCSProxyManager_getProxy_1_Request"> <wsdl:part name="proxyParameters" element="ns0:proxyParameters"/> </wsdl:message> <wsdl:portType name="authorisationSupport"> <wsdl:operation name="authorise" parameterOrder="licenseID userID action resourceID authContext"> <wsdl:input message="tns:authorisationSupport_authorise_1_Request"/> <wsdl:output message="tns:authorisationSupport_authorise_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="ProtectionInfoManager"> <wsdl:operation name="insertProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp protectionInfo"> <wsdl:input message="tns:ProtectionInfoManager_insertProtectionInfo_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_insertProtectionInfo_Response"/> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp"> <wsdl:input message="tns:ProtectionInfoManager_retrieveProtectionInfo_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_retrieveProtectionInfo_Response"/> </wsdl:operation> <wsdl:operation name="generateKey" parameterOrder="algorithm keyLength parameters"> <wsdl:input message="tns:ProtectionInfoManager_generateKey_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_generateKey_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="RightsExpressionTranslator"> <wsdl:operation name="generateTranslation" parameterOrder="sourceLicense destinationLanguage"> <wsdl:input message="tns:RightsExpressionTranslator_generateTranslation_1_Request"/> <wsdl:output message="tns:RightsExpressionTranslator_generateTranslation_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="DomainManager"> <wsdl:operation name="searchDomains" parameterOrder="field value"> <wsdl:input message="tns:DomainManager_searchDomains_1_Request"/> <wsdl:output message="tns:DomainManager_searchDomains_Response"/> </wsdl:operation> </pre> |
|--|--|

```

<wsdl:operation name="createDomain" parameterOrder="myDomain">
  <wsdl:input message="tns:DomainManager_createDomain_1_Request"/>
  <wsdl:output message="tns:DomainManager_createDomain_Response"/>
</wsdl:operation>
<wsdl:operation name="deleteDomain" parameterOrder="idDomain">
  <wsdl:input message="tns:DomainManager_deleteDomain_1_Request"/>
  <wsdl:output message="tns:DomainManager_deleteDomain_Response"/>
</wsdl:operation>
<wsdl:operation name="updateDomain" parameterOrder="myDomain">
  <wsdl:input message="tns:DomainManager_updateDomain_1_Request"/>
  <wsdl:output message="tns:DomainManager_updateDomain_Response"/>
</wsdl:operation>
<wsdl:operation name="registrationRequest" parameterOrder="myUser myDomain">
  <wsdl:input message="tns:DomainManager_registrationRequest_1_Request"/>
  <wsdl:output message="tns:DomainManager_registrationRequest_Response"/>
</wsdl:operation>
<wsdl:operation name="unRegistrationRequest" parameterOrder="userID myDomain">
  <wsdl:input message="tns:DomainManager_unRegistrationRequest_1_Request"/>
  <wsdl:output message="tns:DomainManager_unRegistrationRequest_Response"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseGenerator">
  <wsdl:operation name="generateFinalLicense" parameterOrder="userIDs rights AXOIDs conditions">
    <wsdl:input message="tns:LicenseGenerator_generateFinalLicense_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_generateFinalLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="generateDistributorLicense" parameterOrder="userIDs rights grants AXOIDs
conditions">
    <wsdl:input message="tns:LicenseGenerator_generateDistributorLicense_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_generateDistributorLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="adaptDRMRules" parameterOrder="sourceLicense constraints">
    <wsdl:input message="tns:LicenseGenerator_adaptDRMRules_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_adaptDRMRules_Response"/>
  </wsdl:operation>
  <wsdl:operation name="adaptPAR" parameterOrder="PAR constraints">
    <wsdl:input message="tns:LicenseGenerator_adaptPAR_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_adaptPAR_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseVerifier">
  <wsdl:operation name="verifyLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseVerifier_verifyLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyCreatedLicense" parameterOrder="license PARs parentLicense">
    <wsdl:input message="tns:LicenseVerifier_verifyCreatedLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyCreatedLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyTemporalLicense" parameterOrder="license context">
    <wsdl:input message="tns:LicenseVerifier_verifyTemporalLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyTemporalLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyPAR" parameterOrder="PAR">
    <wsdl:input message="tns:LicenseVerifier_verifyPAR_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyPAR_Response"/>
  </wsdl:operation>
  <wsdl:operation name="checkPARRules" parameterOrder="PAR PARs parentLicense">
    <wsdl:input message="tns:LicenseVerifier_checkPARRules_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_checkPARRules_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseManager">
  <wsdl:operation name="retrieveLicense" parameterOrder="AXOID licenseID">
    <wsdl:input message="tns:LicenseManager_retrieveLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_retrieveLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseManager_storeLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_storeLicense_Response"/>
  </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="storeLicenseModel" parameterOrder="licenseModel">
  <wsdl:input message="tns:LicenseManager_storeLicenseModel_1_Request"/>
  <wsdl:output message="tns:LicenseManager_storeLicenseModel_Response"/>
</wsdl:operation>
<wsdl:operation name="deleteLicense" parameterOrder="AXOID licenseID">
  <wsdl:input message="tns:LicenseManager_deleteLicense_1_Request"/>
  <wsdl:output message="tns:LicenseManager_deleteLicense_Response"/>
</wsdl:operation>
<wsdl:operation name="retrieveLicenseModel" parameterOrder="licModelID">
  <wsdl:input message="tns:LicenseManager_retrieveLicenseModel_1_Request"/>
  <wsdl:output message="tns:LicenseManager_retrieveLicenseModel_Response"/>
</wsdl:operation>
<wsdl:operation name="updateLicenseModel" parameterOrder="licModelID licModelNew">
  <wsdl:input message="tns:LicenseManager_updateLicenseModel_1_Request"/>
  <wsdl:output message="tns:LicenseManager_updateLicenseModel_Response"/>
</wsdl:operation>
<wsdl:operation name="deleteLicenseModel" parameterOrder="licModelID">
  <wsdl:input message="tns:LicenseManager_deleteLicenseModel_1_Request"/>
  <wsdl:output message="tns:LicenseManager_deleteLicenseModel_Response"/>
</wsdl:operation>
<wsdl:operation name="revokeLicense" parameterOrder="licenseID">
  <wsdl:input message="tns:LicenseManager_revokeLicense_1_Request"/>
  <wsdl:output message="tns:LicenseManager_revokeLicense_Response"/>
</wsdl:operation>
<wsdl:operation name="revokeAddLicense" parameterOrder="licenseID newLicense">
  <wsdl:input message="tns:LicenseManager_revokeAddLicense_1_Request"/>
  <wsdl:output message="tns:LicenseManager_revokeAddLicense_Response"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="ContentConsumption">
  <wsdl:operation name="insertActionLog" parameterOrder="AXOID objectVersion protectionStamp
actionLog">
    <wsdl:input message="tns:ContentConsumption_insertActionLog_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_insertActionLog_Response"/>
  </wsdl:operation>
  <wsdl:operation name="retrieveActionLogs">
    <wsdl:input message="tns:ContentConsumption_retrieveActionLogs_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_retrieveActionLogs_Response"/>
  </wsdl:operation>
  <wsdl:operation name="deleteCacheContent">
    <wsdl:input message="tns:ContentConsumption_deleteCacheContent_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_deleteCacheContent_Response"/>
  </wsdl:operation>
  <wsdl:operation name="clearActionLogs">
    <wsdl:input message="tns:ContentConsumption_clearActionLogs_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_clearActionLogs_Response"/>
  </wsdl:operation>
  <wsdl:operation name="getLastActionLog" parameterOrder="AXOID objectVersion protectionStamp">
    <wsdl:input message="tns:ContentConsumption_getLastActionLog_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_getLastActionLog_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="AXCSPProxyManager">
  <wsdl:operation name="getProxy" parameterOrder="proxyParameters">
    <wsdl:input message="tns:AXCSPProxyManager_getProxy_1_Request"/>
    <wsdl:output message="tns:AXCSPProxyManager_getProxy_Response"/>
  </wsdl:operation>
  <wsdl:operation name="releaseAXCSPProxy" parameterOrder="proxy">
    <wsdl:input message="tns:AXCSPProxyManager_releaseAXCSPProxy_1_Request"/>
    <wsdl:output message="tns:AXCSPProxyManager_releaseAXCSPProxy_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="authorisationSupport" type="tns:authorisationSupport">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="authorise">
    <map:java-operation name="authorise"
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n
O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/>
    <soap:operation

```


| | |
|--|---|
| | <pre> soapAction="urn:authorisationSupportauthorisationSupport#authorise#KExqYXZhL2xhbmvcU3RyaW5nO0xqY XZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbm cvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="ProtectionInfoManager" type="tns:ProtectionInfoManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="insertProtectionInfo"> <map:java-operation name="insertProtectionInfo" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5n O0xQcm90ZWNoaW9uSW5mb3JtYXRpb247KVo="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#insertProtectionInfo#KExqYXZhL2xhbmvcU3R yaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xQcm90ZWNoaW9uSW5mb3JtYX Rpb247KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo"> <map:java-operation name="retrieveProtectionInfo" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5n OylMUHJvdGVjdGlvbkluzm9ybWF0aW9uOw=="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#retrieveProtectionInfo#KExqYXZhL2xhbmvcU3 RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMUHJvdGVjdGlvbkluzm9ybWF0 aW9uOw==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateKey"> <map:java-operation name="generateKey" signature="KExBbGdvcmI0aG07SUxqYXZhL3V0aWwvVmVjdG9yOylMS2V5QVg7"/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#generateKey#KExBbGdvcmI0aG07SUxqYXZh L3V0aWwvVmVjdG9yOylMS2V5QVg7" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="RightsExpressionTranslator" type="tns:RightsExpressionTranslator"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="generateTranslation"> <map:java-operation name="generateTranslation" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmIu Zzs="/> <soap:operation soapAction="urn:translator.RightsExpressionTranslatorRightsExpressionTranslator#generateTranslation#KExq YXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> </pre> |
|--|---|

| | |
|--|--|
| | <pre> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="DomainManager" type="tns:DomainManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="searchDomains"> <map:java-operation name="searchDomains" signature="KExqYXZhL2xhbmvcvU3RyaW5nO0xqYXZhL2xhbmvcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5T GlzdDs="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#searchDomains#KExqYXZhL2xhbmvcvU3 RyaW5nO0xqYXZhL2xhbmvcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGlzdDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="createDomain"> <map:java-operation name="createDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWpjbWpWg=="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#createDomain#KExEb21haW5QYWNRyW dlL0RvbWpjbWpWg==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteDomain"> <map:java-operation name="deleteDomain" signature="KExqYXZhL2xhbmvcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#deleteDomain#KExqYXZhL2xhbmvcvU3R yaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="updateDomain"> <map:java-operation name="updateDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWpjbWpWg=="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#updateDomain#KExEb21haW5QYWNRy WdlL0RvbWpjbWpWg==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="registrationRequest"> <map:java-operation name="registrationRequest" signature="KExEb21haW5QYWNRyWdlL1VzZXI7TGphdmEvbGFuZy9TdHJpbmc7KVo="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#registrationRequest#KExEb21haW5QYW NRyWdlL1VzZXI7TGphdmEvbGFuZy9TdHJpbmc7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </pre> |
|--|--|

| | |
|--|---|
| | <pre> </wsdl:output> </wsdl:operation> <wsdl:operation name="unRegistrationRequest"> <map:java-operation name="unRegistrationRequest" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#unRegistrationRequest#KExqYXZhL2xhb mcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseGenerator" type="tns:LicenseGenerator"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="generateFinalLicense"> <map:java-operation name="generateFinalLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#generateFinalLicense#KExqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2 YS9sYW5nL1N0cmZsZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateDistributorLicense"> <map:java-operation name="generateDistributorLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#generateDistributorLicense#KExqYXZhL2xhbmcvU3Rya W5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqY XZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="adaptDRMRules"> <map:java-operation name="adaptDRMRules" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs Zs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptDRMRules#KExqYXZhL2xhbmcvU3RyaW5nO0xq YXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="adaptPAR"> <map:java-operation name="adaptPAR" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs Zs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptPAR#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL 2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs=" style="document"/> <wsdl:input> </pre> |
|--|---|

| | |
|--|---|
| | <pre> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseVericator" type="tns:LicenseVericator"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="verifyLicense"> <map:java-operation name="verifyLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#verifyLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyCreatedLicense"> <map:java-operation name="verifyCreatedLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#verifyCreatedLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyTemporalLicense"> <map:java-operation name="verifyTemporalLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs="/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#verifyTemporalLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyPAR"> <map:java-operation name="verifyPAR" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#verifyPAR#KExqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="checkPARRules"> <map:java-operation name="checkPARRules" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#checkPARRules#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </pre> |
|--|---|

| | |
|--|---|
| | <pre> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseManager" type="tns:LicenseManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="retrieveLicense"> <map:java-operation name="retrieveLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs="/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#retrieveLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeLicense"> <map:java-operation name="storeLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#storeLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeLicenseModel"> <map:java-operation name="storeLicenseModel" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#storeLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteLicense"> <map:java-operation name="deleteLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#deleteLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveLicenseModel"> <map:java-operation name="retrieveLicenseModel" signature="KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs="/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#retrieveLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> </pre> |
|--|---|

```

        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="updateLicenseModel">
      <map:java-operation name="updateLicenseModel"
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/>
      <soap:operation
soapAction="urn:LicenseManagerLicenseManager#updateLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nO0x
qYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="deleteLicenseModel">
      <map:java-operation name="deleteLicenseModel"
signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/>
      <soap:operation
soapAction="urn:LicenseManagerLicenseManager#deleteLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nOyla
" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="revokeLicense">
      <map:java-operation name="revokeLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/>
      <soap:operation
soapAction="urn:LicenseManagerLicenseManager#revokeLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="revokeAddLicense">
      <map:java-operation name="revokeAddLicense"
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/>
      <soap:operation
soapAction="urn:LicenseManagerLicenseManager#revokeAddLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xq
YXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="ContentConsumption" type="tns:ContentConsumption">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="insertActionLog">
      <map:java-operation name="insertActionLog"
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n
O0xBy3Rpb25Mb2c7KVY="/>
      <soap:operation
soapAction="urn:ContentConsumptionContentConsumption#insertActionLog#KExqYXZhL2xhbmcvU3RyaW5n
O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xBy3Rpb25Mb2c7KVY="
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

```

| | |
|--|--|
| | <pre> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveActionLogs"> <map:java-operation name="retrieveActionLogs" signature="KCIMamF2YS91dGlsL0FycmF5TGldDs="/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#retrieveActionLogs#KCIMamF2YS91dGlsL0Fycm F5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteCacheContent"> <map:java-operation name="deleteCacheContent" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#deleteCacheContent#KClA" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="clearActionLogs"> <map:java-operation name="clearActionLogs" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#clearActionLogs#KClA" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="getLastActionLog"> <map:java-operation name="getLastActionLog" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n OylMQWN0aW9uTG9nOw=="/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#getLastActionLog#KExqYXZhL2xhbmcvU3RyaW 5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMQWN0aW9uTG9nOw==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="AXCSProxyManager" type="tns:AXCSProxyManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="getProxy"> <map:java-operation name="getProxy" signature="KExqYXZhL2xhbmcvU3RyaW5nOylMQVhDU1Byb3h5Ow=="/> <soap:operation soapAction="urn:AXCSProxyManagerAXCSProxyManager#getProxy#KExqYXZhL2xhbmcvU3RyaW5nOylMQ VhDU1Byb3h5Ow==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="releaseAXCSProxy"> <map:java-operation name="releaseAXCSProxy" signature="KExBWENTUHJveHk7KVo="/> </pre> |
|--|--|

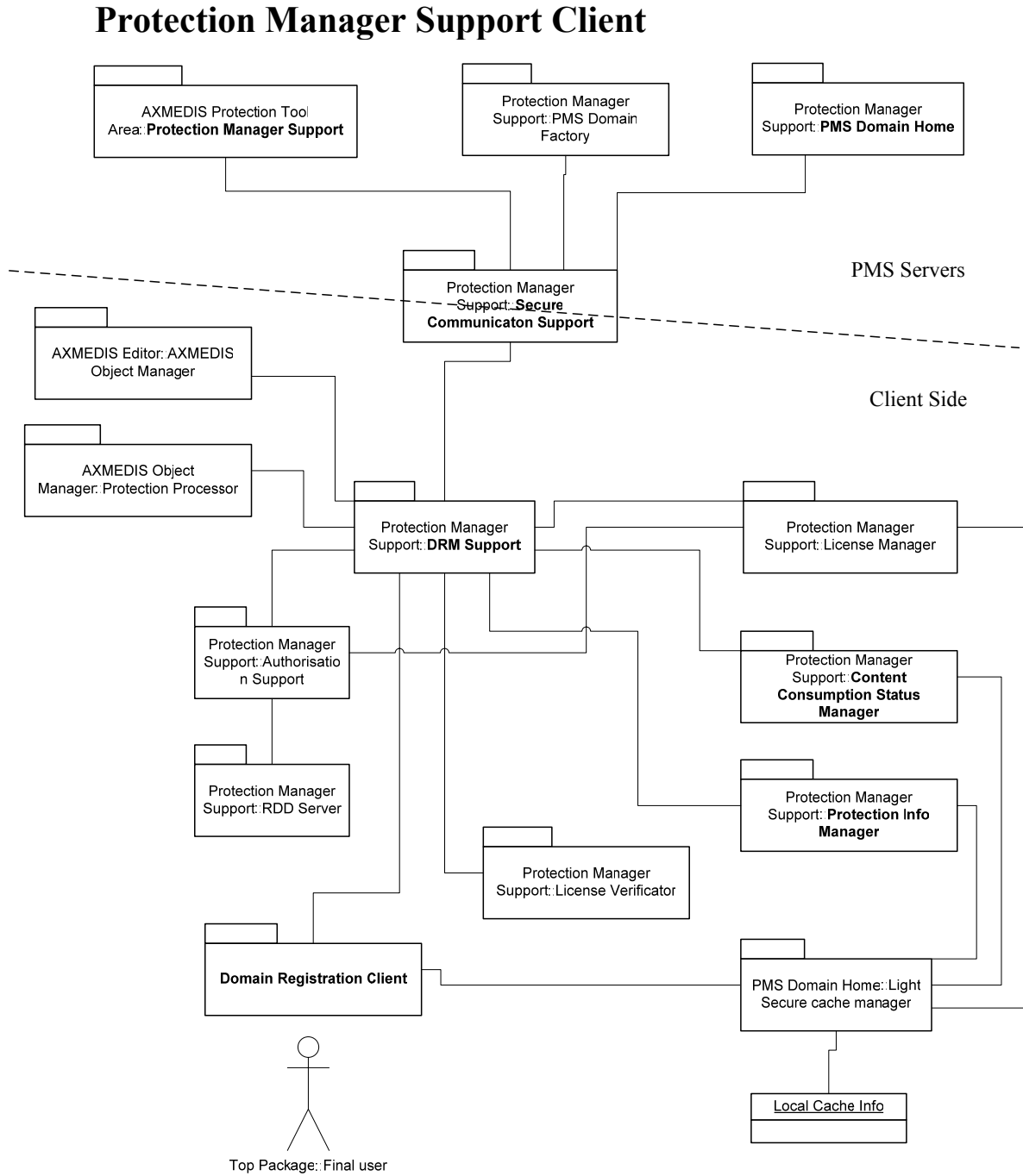
| | |
|--|---|
| | <pre> <soap:operation soapAction="urn:AXCSPProxyManagerAXCSPProxyManager#releaseAXCSPProxy#KExBWENTUHJveHk7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="authorisationSupport"> <wsdl:port name="authorisationSupport" binding="tns:authorisationSupport"> <soap:address location="http://cjlal1:6060/authorisationSupport"/> </wsdl:port> </wsdl:service> <wsdl:service name="ProtectionInfoManager"> <wsdl:port name="ProtectionInfoManager" binding="tns:ProtectionInfoManager"> <soap:address location="http://efpc037:6060/ProtectionInfoManager"/> </wsdl:port> </wsdl:service> <wsdl:service name="RightsExpressionTranslator"> <wsdl:port name="RightsExpressionTranslator" binding="tns:RightsExpressionTranslator"> <soap:address location="http://EFPC016:6060/RightsExpressionTranslator"/> </wsdl:port> </wsdl:service> <wsdl:service name="DomainManager"> <wsdl:port name="DomainManager" binding="tns:DomainManager"> <soap:address location="http://EFPC016:6060/DomainManager"/> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseGenerator"> <wsdl:port name="LicenseGenerator" binding="tns:LicenseGenerator"> <soap:address location="http://efpc037:6060/LicenseGenerator"/> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseVerificator"> <wsdl:port name="LicenseVerificator" binding="tns:LicenseVerificator"> <soap:address location="http://efpc037:6060/LicenseVerificator"/> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseManager"> <wsdl:port name="LicenseManager" binding="tns:LicenseManager"> <soap:address location="http://efpc037:6060/LicenseManager"/> </wsdl:port> </wsdl:service> <wsdl:service name="ContentConsumption"> <wsdl:port name="ContentConsumption" binding="tns:ContentConsumption"> <soap:address location="http://efpc037:6060/ContentConsumption"/> </wsdl:port> </wsdl:service> <wsdl:service name="AXCSPProxyManager"> <wsdl:port name="AXCSPProxyManager" binding="tns:AXCSPProxyManager"> <soap:address location="http://cjlal1:6060/AXCSPProxyManager"/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|--|---|

Operations are described in the corresponding modules throughout this section.

Each method of PMS corresponds to a method offered in a submodule. The name of the module corresponds exactly with the name of the “port type” in the WSDL definition above. For instance, if one method is offered by the port “Authorisation Support”, the description of the corresponding method can be found in the module Authorisation Support, and so on.

4.6 Protection Manager Support Client (FUPF)

PMS Client works in the user side, providing both on-line and off-line functionalities for the final user. It has to provide protection and licensing features in the user side, but it also has to be prepared to communicate with the rest of PMS, Server and Domain.

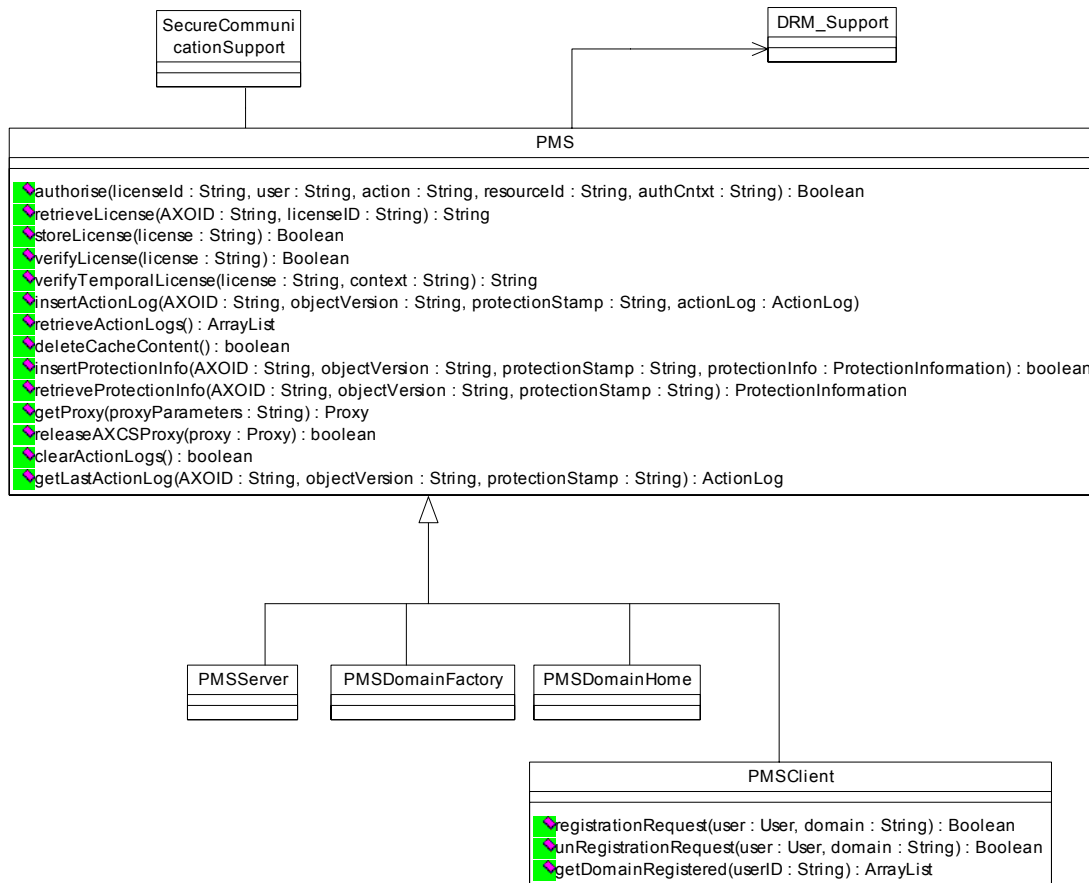


4.6.1 Module profile definition

| Module Profile | | |
|--|-----------------------------------|--|
| Protection Manager Support Client | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Secure Communication Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.6.2 Architecture of PMS

Next figure shows the architecture of PMS client module.



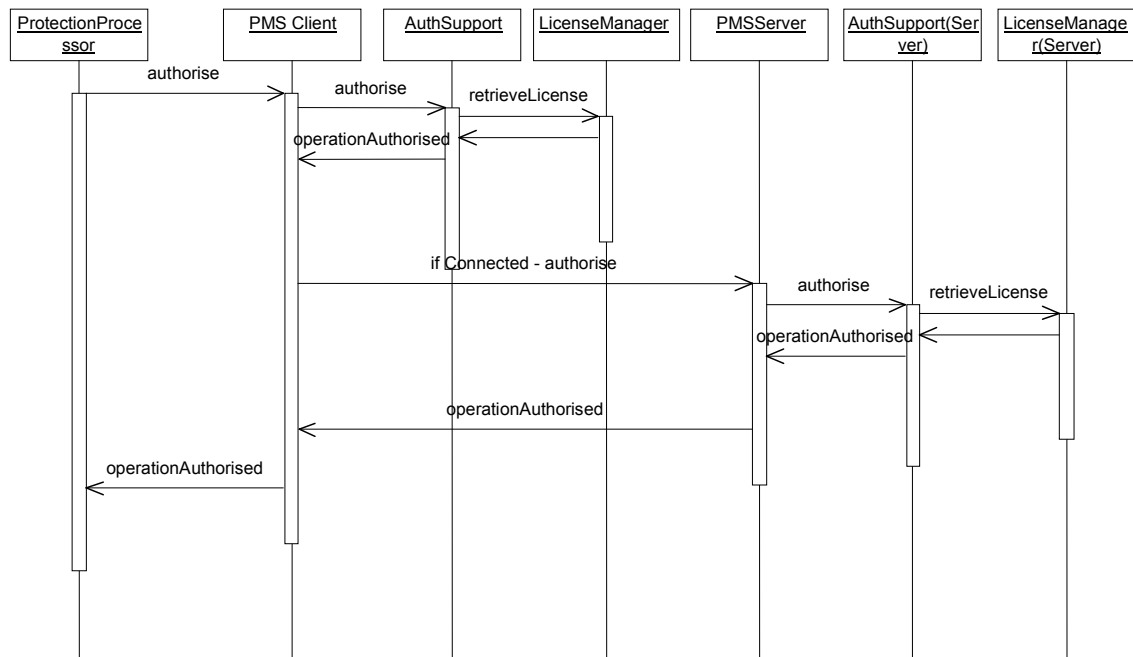
Class diagram of PMS Client

| PMS Client | |
|-----------------------|---|
| Methods | Description |
| authorise | This function authorises AXMEDIS users to perform actions over AXMEDIS objects. It proves that a user has the appropriate license that grants him to exercise a right over a resource if the conditions are met based on the execution context of the client. |
| retrieveLicense | This function retrieves the licenses stored in the license database. It retrieves the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| storeLicense | This function stores a license in the license database. |
| verifyLicense | Verifies a license syntactically against the schemas defined within the license. |
| verifyTemporalLicense | Verifies that the license generated by the user fulfils the initial desirables requirements of the user. For example, the user can verify that with this license he could exercise the desired action over the AXObject. |
| registrationRequest | This function is used to a registration of an user in a certain domain |
| unRegistrationRequest | This function is used to an unregistration of an user in a certain domain |
| getDomainsRegistered | This method returns the domain a user is registered to. |
| insertActionLog | Stores the given action log associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveActionLogs | This method retrieves the action logs stored in the local cache info. |

| | |
|------------------------|--|
| deleteCacheContent | This method retrieves the action logs stored in the local cache info. |
| insertProtectionInfo | Stores the given protection information associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveProtectionInfo | This method retrieves the requested protection information. The information needed to retrieve protection information is the AXMEDIS object identifier, the object version and the protection stamp. |
| clearActionLogs | Deletes action logs from the cache, after positive authorisation of the user in the connected environment |
| getLastActionLog | Returns the last action log |
| getProxy | Returns a Proxy to establish a secure communication |
| releaseAXCSProxy | Finish the secure communication |

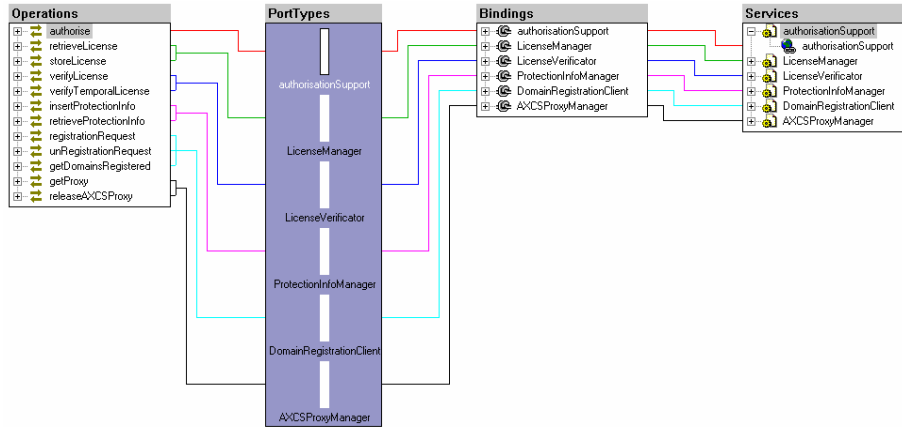
4.6.3 Sequence diagram

The following diagram shows how an authorisation has to be performed either in the on-line and off-line environments.



Sequence diagram of the authorisation between PMS Client and PMS Server

4.6.4 WSDL Interface



| PMS Client | |
|------------|--|
| WSDL | <pre> <wsdl:definitions xmlns:tns="urn:authorisationSupport" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:authorisationSupport" name="authorisationSupport"> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="action" type="xsd:string" nillable="true"/> <xsd:element name="resourceID" type="xsd:string" nillable="true"/> <xsd:element name="authContext" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> <xsd:element name="license" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="license" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="PARs" type="xsd:string" nillable="true"/> <xsd:element name="parentLicense" type="xsd:string" nillable="true"/> <xsd:element name="context" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="userIDs" type="xsd:string" nillable="true"/> <xsd:element name="rights" type="xsd:string" nillable="true"/> <xsd:element name="AXOIDs" type="xsd:string" nillable="true"/> <xsd:element name="conditions" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> <xsd:element name="grants" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> </pre> |

```

</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="ActionLog">
        <!-- Defined in AXMEDIS Supervisor -->
        <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
        <xsd:sequence>
            <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
        <xsd:complexContent mixed="false">
            <xsd:extension base="tns:List"/>
        </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"
xmlns:xns5="http://systinet.com/containers/literal/ms.net">
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:import namespace="http://systinet.com/wsdl/default"/>
    <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
    <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
    <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
    <xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
    <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
        <xsd:sequence>
            <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Vector">
        <xsd:complexContent mixed="false">
            <xsd:extension base="tns:List"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="ArrayOfAnyType">
        <xsd:sequence>
            <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/>
        </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/"
xmlns:xns5="http://systinet.com/containers/literal/ms.net">
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:import namespace="http://systinet.com/wsdl/default"/>
    <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
    <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
    <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
    <xsd:element name="protectionInfo" type="xns4:ProtectionInformation" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="ProtectionInformation_Response" type="xns4:ProtectionInformation"
nillable="true"/>

```

```

<xsd:element name="algorithm" type="xns4:Algorithm" nillable="true"/>
<xsd:element name="keyLength" type="xsd:int"/>
<xsd:element name="parameters" type="xns5:Vector" nillable="true"/>
<xsd:element name="KeyAX_Response" type="xns4:KeyAX" nillable="true"/>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
xmlns:map="http://systinet.com/mapping/">
  <xsd:complexType name="ProtectionInformation">
    <xsd:sequence>
      <xsd:element name="protectionInfo" type="xsd:string" nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Algorithm">
    <xsd:sequence>
      <xsd:element name="algorithm" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="KeyAX">
    <xsd:sequence>
      <xsd:element name="key" type="xsd:base64Binary" nillable="true"/>
      <xsd:element name="keyByte" type="xsd:base64Binary" nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/>
    <xsd:element name="destinationLanguage" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/containers/literal/ms.net"
targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
      <xsd:sequence>
        <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
      <xsd:complexContent mixed="false">
        <xsd:extension base="tns:List"/>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/containers/literal/ms.net"
xmlns:xns5="urn:DomainPackage.DomainManager">
    <xsd:import namespace="urn:DomainPackage.DomainManager"/>
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:element name="field" type="xsd:string" nillable="true"/>
    <xsd:element name="value" type="xsd:string" nillable="true"/>
    <xsd:element name="ArrayList_Response" type="xns4:ArrayList" nillable="true"/>
    <xsd:element name="myDomain" type="xns5:Domain" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="idDomain" type="xsd:string" nillable="true"/>
    <xsd:element name="myUser" type="xns5:User" nillable="true"/>
    <xsd:element name="myDomain_1" type="xsd:string" nillable="true"/>
    <xsd:element name="userID" type="xsd:string" nillable="true"/>
    <xsd:element name="myDomain_2" type="xsd:string" nillable="true"/>
  </xsd:schema>
  <xsd:schema targetNamespace="urn:DomainPackage.DomainManager"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:DomainPackage.DomainManager" xmlns:map="http://systinet.com/mapping/">

```

```

<xsd:complexType name="Domain">
  <xsd:sequence>
    <xsd:element name="AXDOM" type="xsd:string" nillable="true"/>
    <xsd:element name="AXID" type="xsd:string" nillable="true"/>
    <xsd:element name="typeOfID" type="xsd:string" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="User">
  <xsd:sequence/>
</xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="urn:DomainPackage.DomainRegistrationClient"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="urn:DomainPackage.DomainRegistrationClient" xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="User">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="DomainPackage.User"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/"
    xmlns:xns4="urn:DomainPackage.DomainRegistrationClient"
    xmlns:xns5="http://systinet.com/containers/literal/ms.net">
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:import namespace="urn:DomainPackage.DomainRegistrationClient"/>
    <xsd:element name="myUser" type="xns4:User" nillable="true"/>
    <xsd:element name="myDomain" type="xsd:string" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="userID" type="xsd:string" nillable="true"/>
    <xsd:element name="idUserario" type="xsd:string" nillable="true"/>
    <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
  </xsd:schema>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/containers/literal/ms.net"
    targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified">
    <xsd:complexType name="List">
      <xsd:sequence>
        <xsd:element name="Items" type="tns:ArrayOfAnyType"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
      <xsd:complexContent mixed="false">
        <xsd:extension base="tns:List"/>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/"
    xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="AXCSPProxy">
      <xsd:sequence>
        <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/>
        <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/>
        <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ActionLog">
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="CertificationResult">
      <xsd:sequence/>
    </xsd:complexType>
  </wsdl:types>

```

```

</xsd:complexType>
<xsd:complexType name="VerificationResult">
  <xsd:sequence/>
</xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsd/default/"
  <xsd:import namespace="http://systinet.com/wsd/default/" />
  <xsd:element name="AXCSPProxy_Response" type="xns4:AXCSPProxy" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/wsd/default/" elementFormDefault="qualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsd/default/"
    xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="AXCSPProxy">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="AXCSPProxy"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence>
        <xsd:element name="theActionLog" type="tns:ActionLog" nillable="true"/>
        <xsd:element name="theCertificationResult" type="tns:CertificationResult" nillable="true"/>
        <xsd:element name="theVerificationResult" type="tns:VerificationResult" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ActionLog">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="ActionLog"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="CertificationResult">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="CertificationResult"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
    <xsd:complexType name="VerificationResult">
      <xsd:annotation>
        <xsd:appinfo>
          <map:java-type name="VerificationResult"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:sequence/>
    </xsd:complexType>
  </xsd:schema>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsd/default/"
    <xsd:import namespace="http://systinet.com/wsd/default/" />
    <xsd:element name="proxyParameters" type="xsd:string" nillable="true"/>
    <xsd:element name="AXCSPProxy_Response" type="xns4:AXCSPProxy" nillable="true"/>
    <xsd:element name="proxy" type="xns4:AXCSPProxy" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="authorisationSupport_authorise_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="authorisationSupport_authorise_1_Request">
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="action" element="ns0:action"/>

```

| | |
|--|--|
| | <pre> <wsdl:part name="resourceID" element="ns0:resourceID"/> <wsdl:part name="authContext" element="ns0:authContext"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyTemporalLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> <wsdl:part name="context" element="ns0:context"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyCreatedLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyCreatedLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> <wsdl:part name="PARs" element="ns0:PARs"/> <wsdl:part name="parentLicense" element="ns0:parentLicense"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyTemporalLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVerificator_verifyLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="grants" element="ns0:grants"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/> <wsdl:message name="ContentConsumption_retrieveActionLogs_Response"> <wsdl:part name="response" element="ns0:ArrayList_Response"/> </wsdl:message> <wsdl:message name="ContentConsumption_insertActionLog_Response"/> <wsdl:message name="ContentConsumption_insertActionLog_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> </pre> |
|--|--|


```

<wsdl:part name="objectVersion" element="ns0:objectVersion"/>
<wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
<wsdl:part name="actionLog" element="ns0:actionLog"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"/>
<wsdl:message name="ContentConsumption_deleteCacheContent_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
  <wsdl:part name="protectionInfo" element="ns0:protectionInfo"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:ProtectionInformation_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_1_Request">
  <wsdl:part name="algorithm" element="ns0:algorithm"/>
  <wsdl:part name="keyLength" element="ns0:keyLength"/>
  <wsdl:part name="parameters" element="ns0:parameters"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_Response">
  <wsdl:part name="response" element="ns0:KeyAX_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_1_Request">
  <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/>
  <wsdl:part name="destinationLanguage" element="ns0:destinationLanguage"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_1_Request">
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_2"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_1_Request">
  <wsdl:part name="idDomain" element="ns0:idDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>

```



```

</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_1_Request">
  <wsdl:part name="myUser" element="ns0:myUser"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_1"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_1_Request">
  <wsdl:part name="field" element="ns0:field"/>
  <wsdl:part name="value" element="ns0:value"/>
</wsdl:message>
<wsdl:message name="DomainRegistrationClient_unRegistrationRequest_1_Request">
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainRegistrationClient_unRegistrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainRegistrationClient_getDomainsRegistered_1_Request">
  <wsdl:part name="idUserario" element="ns0:idUsuario"/>
</wsdl:message>
<wsdl:message name="DomainRegistrationClient_registrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainRegistrationClient_registrationRequest_1_Request">
  <wsdl:part name="myUser" element="ns0:myUser"/>
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainRegistrationClient_getDomainsRegistered_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="AXCSProxy_getProxy_1_Request"/>
<wsdl:message name="AXCSProxy_getProxy_Response">
  <wsdl:part name="response" element="ns0:AXCSProxy_Response"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_releaseAXCSProxy_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_releaseAXCSProxy_1_Request">
  <wsdl:part name="proxy" element="ns0:proxy"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_getProxy_Response">
  <wsdl:part name="response" element="ns0:AXCSProxy_Response"/>
</wsdl:message>
<wsdl:message name="AXCSProxyManager_getProxy_1_Request">
  <wsdl:part name="proxyParameters" element="ns0:proxyParameters"/>
</wsdl:message>
<wsdl:portType name="authorisationSupport">
  <wsdl:operation name="authorise" parameterOrder="licenseID userID action resourceID authContext">
    <wsdl:input message="tns:authorisationSupport_authorise_1_Request"/>
    <wsdl:output message="tns:authorisationSupport_authorise_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseManager">
  <wsdl:operation name="retrieveLicense" parameterOrder="AXOID licenseID">
    <wsdl:input message="tns:LicenseManager_retrieveLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_retrieveLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseManager_storeLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_storeLicense_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseVerificator">
  <wsdl:operation name="verifyLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseVerificator_verifyLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerificator_verifyLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyTemporalLicense" parameterOrder="license context">
    <wsdl:input message="tns:LicenseVerificator_verifyTemporalLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerificator_verifyTemporalLicense_Response"/>
  </wsdl:operation>
</wsdl:portType>

```

| | |
|--|---|
| | <pre> </wsdl:portType> <wsdl:portType name="ProtectionInfoManager"> <wsdl:operation name="insertProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp protectionInfo"> <wsdl:input message="tns:ProtectionInfoManager_insertProtectionInfo_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_insertProtectionInfo_Response"/> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp"> <wsdl:input message="tns:ProtectionInfoManager_retrieveProtectionInfo_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_retrieveProtectionInfo_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="DomainRegistrationClient"> <wsdl:operation name="registrationRequest" parameterOrder="myUser myDomain"> <wsdl:input message="tns:DomainRegistrationClient_registrationRequest_1_Request"/> <wsdl:output message="tns:DomainRegistrationClient_registrationRequest_Response"/> </wsdl:operation> <wsdl:operation name="unRegistrationRequest" parameterOrder="userID myDomain"> <wsdl:input message="tns:DomainRegistrationClient_unRegistrationRequest_1_Request"/> <wsdl:output message="tns:DomainRegistrationClient_unRegistrationRequest_Response"/> </wsdl:operation> <wsdl:operation name="getDomainsRegistered" parameterOrder="idUserario"> <wsdl:input message="tns:DomainRegistrationClient_getDomainsRegistered_1_Request"/> <wsdl:output message="tns:DomainRegistrationClient_getDomainsRegistered_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:portType name="AXCSProxyManager"> <wsdl:operation name="getProxy" parameterOrder="proxyParameters"> <wsdl:input message="tns:AXCSProxyManager_getProxy_1_Request"/> <wsdl:output message="tns:AXCSProxyManager_getProxy_Response"/> </wsdl:operation> <wsdl:operation name="releaseAXCSProxy" parameterOrder="proxy"> <wsdl:input message="tns:AXCSProxyManager_releaseAXCSProxy_1_Request"/> <wsdl:output message="tns:AXCSProxyManager_releaseAXCSProxy_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="authorisationSupport" type="tns:authorisationSupport"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="authorise"> <map:java-operation name="authorise" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:authorisationSupportauthorisationSupport#authorise#KExqYXZhL2xhbmcvU3RyaW5nO0xqY XZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhb cvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseManager" type="tns:LicenseManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="retrieveLicense"> <map:java-operation name="retrieveLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIu Zzs="/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#retrieveLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZ hL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </pre> |
|--|---|

| | |
|--|--|
| | <pre> </wsdl:operation> <wsdl:operation name="storeLicense"> <map:java-operation name="storeLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#storeLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseVerifier" type="tns:LicenseVerifier"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="verifyLicense"> <map:java-operation name="verifyLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVerifierLicenseVerifier#verifyLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyTemporalLicense"> <map:java-operation name="verifyTemporalLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLu Zzs="/> <soap:operation soapAction="urn:LicenseVerifierLicenseVerifier#verifyTemporalLicense#KExqYXZhL2xhbmvcU3RyaW5 nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="ProtectionInfoManager" type="tns:ProtectionInfoManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="insertProtectionInfo"> <map:java-operation name="insertProtectionInfo" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5n O0xQcm90ZWNoaW9uSW5mb3JtYXRpb247KVo="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#insertProtectionInfo#KExqYXZhL2xhbmvcU3R yaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xQcm90ZWNoaW9uSW5mb3JtYX Rpb247KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo"> <map:java-operation name="retrieveProtectionInfo" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5n OylMUHJvdGVjdGlvbkludm9ybWFOaW9uOW=="> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#retrieveProtectionInfo#KExqYXZhL2xhbmvcU3R yaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMUHJvdGVjdGlvbkludm9ybWFO aW9uOW==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> </pre> |
|--|--|

| | |
|--|--|
| | <pre> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="DomainRegistrationClient" type="tns:DomainRegistrationClient"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="registrationRequest"> <map:java-operation name="registrationRequest" signature="KExEb21haW5QYWNrYWdlL1VzZXI7TGphdmEvdGFuZy9TdHJpbmc7KVo="/> <soap:operation soapAction="urn:DomainPackage.DomainRegistrationClientDomainRegistrationClient#registrationRequest#KE xEb21haW5QYWNrYWdlL1VzZXI7TGphdmEvdGFuZy9TdHJpbmc7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="unRegistrationRequest"> <map:java-operation name="unRegistrationRequest" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainRegistrationClientDomainRegistrationClient#unRegistrationRequest# KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="getDomainsRegistered"> <map:java-operation name="getDomainsRegistered" signature="KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGldDs="/> <soap:operation soapAction="urn:DomainPackage.DomainRegistrationClientDomainRegistrationClient#getDomainsRegistered# KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="AXCSPProxyManager" type="tns:AXCSPProxyManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="getProxy"> <map:java-operation name="getProxy" signature="KExqYXZhL2xhbmcvU3RyaW5nOylMQVhDU1Byb3h5Ow=="/> <soap:operation soapAction="urn:AXCSPProxyManagerAXCSPProxyManager#getProxy#KExqYXZhL2xhbmcvU3RyaW5nOylMQ VhDU1Byb3h5Ow==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="releaseAXCSPProxy"> <map:java-operation name="releaseAXCSPProxy" signature="KExBWENTUHJveHk7KVo="/> <soap:operation soapAction="urn:AXCSPProxyManagerAXCSPProxyManager#releaseAXCSPProxy#KExBWENTUHJveHk7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </pre> |
|--|--|

| | |
|--|--|
| | <pre> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="authorisationSupport"> <wsdl:port name="authorisationSupport" binding="tns:authorisationSupport"> <soap:address location="http://cjlal1:6060/authorisationSupport/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseManager"> <wsdl:port name="LicenseManager" binding="tns:LicenseManager"> <soap:address location="http://cjlal1:6060/LicenseManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseVericator"> <wsdl:port name="LicenseVericator" binding="tns:LicenseVericator"> <soap:address location="http://cjlal1:6060/LicenseVericator/"> </wsdl:port> </wsdl:service> <wsdl:service name="ProtectionInfoManager"> <wsdl:port name="ProtectionInfoManager" binding="tns:ProtectionInfoManager"> <soap:address location="http://efpc037:6060/ProtectionInfoManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="DomainRegistrationClient"> <wsdl:port name="DomainRegistrationClient" binding="tns:DomainRegistrationClient"> <soap:address location="http://EFPC016:6060/DomainRegistrationClient/"> </wsdl:port> </wsdl:service> <wsdl:service name="AXCSProxyManager"> <wsdl:port name="AXCSProxyManager" binding="tns:AXCSProxyManager"> <soap:address location="http://cjlal1:6060/AXCSProxyManager/"> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|--|--|

Operations are described in the corresponding modules throughout this section.

Each method of PMS corresponds to a method offered in a submodule. The name of the module corresponds exactly with the name of the “port type” in the WSDL definition above. For instance, if one method is offered by the port “Authorisation Support”, the description of the corresponding method can be found in the module Authorisation Support, and so on.

4.7 AXCS proxy

This module exposes to the protection processor module inside the AXMEDIS editor the needed functionality to access the AXCS module.

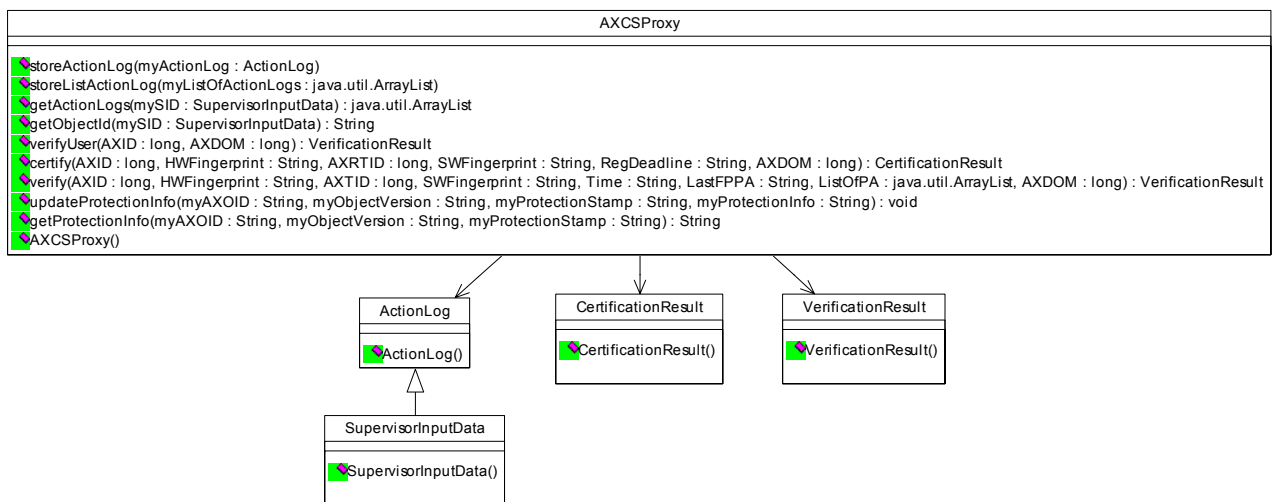
This proxy is provided by the PMS client, which establishes a secure channel with the corresponding PMS Domain and/or PMS server in order to access to the AXCS attached to the PMS server.

| Module Profile | |
|--------------------------------|---------------|
| AXCS proxy | |
| Executable or Library(Support) | Library |
| Single Thread or Multithread | Single thread |
| Language of Development | C++ |

| | | |
|------------------------------|-----------------------------------|--|
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| PMS Client | | |
| Protection Processor | | |
| AXCS | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.7.1 Architecture of the module

The following figure shows the UML diagram of this module, together with the definition of the public operations. ActionLog and SupervisorInputData classes are defined in AXMEDIS Supervisor, while CertificationResult and VerificationResult classes are defined in AXMEDIS Certification and Verification.



AXCS Proxy class diagram

The following table describes the methods offered by the AXCSProxy, available for the protection Processor module:

| AXCSProxy | |
|----------------------|--|
| Method | Description |
| storeActionLog | This function is used to receive the Action Logs and to store them. When an end User requests to perform an action on an AXMEDIS Protected Object and has been certified, verified and authorised, PMS sends to AXCS the action performed (Action Log) and Action Logs are stored by Supervisor. An AXMEDIS user wants to consume an AXMEDIS Object and requests to perform an action on an AXMEDIS Protected Object. Then the user is certified and authorised and the PMS sends to AXCS the action performed. |
| storeListActionLog | This function is a particular case of the function before. In this case the PMS sends to Supervisor a list of Action Logs, not only one. It happens when a user has been off-line for a period of time and has done some actions over an object off-line. When he connects again with the system, the system and the cache of the user have to be synchronised, for what the list of Action Logs is sent to Supervisor. |
| getActionLogs | This function is used to extract Action Logs (from the AXCS DB Interface) and to send it to the AXMEDIS Accounting or reporting tools. When a Distributor wants to recover information on actions performed on the objects he has rights, AXMEDIS Statistic or reporting tools queries AXCS (or SuperAXCS, who recover information from the different AXCSs). Then AXCS extracts the required Action-Logs from the AXCS DB Interface and communicates them to the tools that perform actions to return results in the desired form (These are AXMEDIS Statistic or reporting tools). |
| getObjectId | This function is used to communicate an Object ID when a Content Provider wants to create a new AXMEDIS Object. When a Content Provider wants to create a new AXMEDIS Object, the tool with that is creating the object requests to the AXCS an object ID. Then AXCS interact with Object ID generator to ask an ID and OID generator returns back to AXCS the generated ID. Once the AXCS has the information communicates to the tool the generated ID. |
| verifyUser | This method can be used to verify the status of a user, optionally inside a domain |
| certify | This method is called by the Protection Processor, through the axcsProxy. It is used to certify that the original tool has not been modified and to activate it. It creates a new entry in the CerTools table of the AXCS database associated to the tool user and returns an activation code and tool identifier to the Protection Processor. |
| verify | This method is called by the Protection Processor, through the axcsProxy. It is used to verify that the tool installed on a device has neither been modified nor blocked, that the user is not blocked and that the registered tool is not blocked. It is also responsible for resynchronizing the offline tool operation through the AXSupervisor. |
| getProtectionInfo | This function is used to extract from the Objects Table in the AXCS Objects ID Database the protection information related to an object. |
| updateProtectionInfo | This function is used to update in the Objects Table in the AXCS Objects ID Database the protection information related to an object. |

4.8 DRM Support (FUPF)

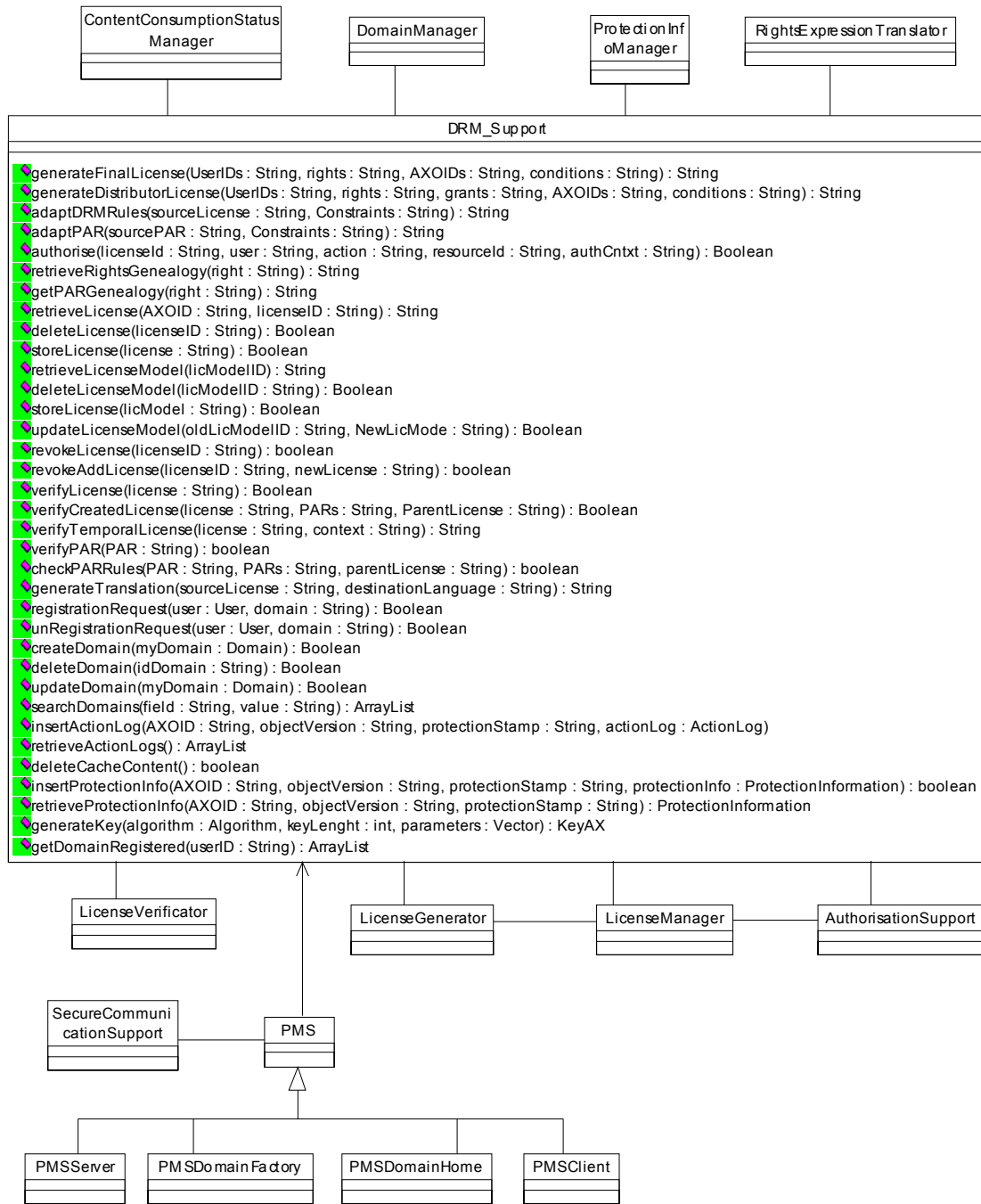
This module is the entry point to the functionality inside each module of PMS. It provides public functions for each module below it, differently for each PMS version.

Module Profile

| DRM Support | | |
|------------------------------------|-----------------------------------|--|
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java, C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Authorisation Support | | |
| PMS Domain | | |
| Domain Manager | | |
| License Manager | | |
| License Generator | | |
| License Verificator | | |
| Content Consumption Status Manager | | |
| Key Manager | | |
| Status Communication Support | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.8.1 Architecture of DRM Support module

Next figure shows the UML diagram of the DRM support module.



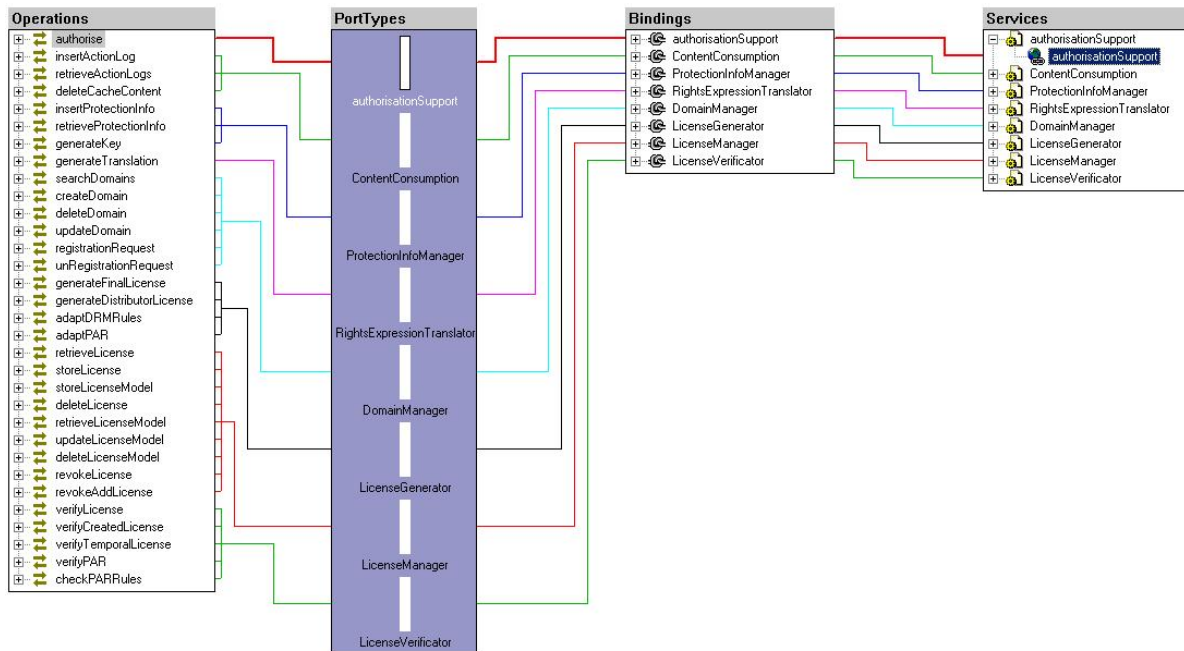
DRM Support class diagram

| DRM Support | |
|----------------------------|--|
| Methods | Description |
| generateFinalLicense | Generates a license for an end-user, according to the userID, right, AXOID and conditions. |
| generateDistributorLicense | Generates a license for a distributor, according to the userID, right, grants that |

| | |
|-------------------------|---|
| | this user will be able to distribute and conditions. |
| authorise | This function authorises AXMEDIS users to perform actions over AXMEDIS objects. It proves that a user has the appropriate license that grants him to exercise a right over a resource if the conditions are met based on the execution context of the client. |
| retrieveRightsGenealogy | Retrieve the right genealogy, parent rights of the input right, according to RDD hierarchy of the rights. |
| getPARGenealogy | Retrieve the right genealogy, child rights of the input right, according to the RDD rights hierarchy. |
| retrieveLicense | This function retrieves the licenses stored in the license database. It retrieves the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| deleteLicense | This function deletes the licenses stored in the license database. It deletes the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| storeLicense | This function stores a license in the license database. |
| verifyLicense | Verifies a license syntactically against the schemas defined within the license. |
| verifyCreatedLicense | Verifies if the license can be generated according to the PARs and the parent licenses (e.g. Distributor or Creator licenses). |
| verifyTemporalLicense | Verifies that the license generated by the user fulfils the initial desirables requirements of the user. For example, the user can verify that with this license he could exercise the desired action over the AXObject. |
| generateTranslation | This function is used to return a destinationLicense. The system makes a translation from a sourceLicense to a destinationLicense. The destinationLicense, if the translation is possible, will be written in a languageDestination. |
| registrationRequest | This function is used to a registration of an user in a certain domain |
| unRegistrationRequest | This function is used to an unregistration of an user in a certain domain |
| createDomain | This function is used to create a new domain. |
| deleteDomain | This function is used to delete a domain with an idDomain identificator |
| updateDomain | This function is used to update a domain. |
| searchDomains | This function is used to return a list of domains with a certain value in a certain field. |
| insertActionLog | Stores the given action log associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveActionLogs | This method retrieves the action logs stored in the local cache info. |
| deleteCacheContent | This method retrieves the action logs stored in the local cache info. |
| insertProtectionInfo | Stores the given protection information associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| retrieveProtectionInfo | This method retrieves the requested protection information. The information needed to retrieve protection information is the AXMEDIS object identifier, the object version and the protection stamp. |
| generateKey | This method permits the creation of a key for protecting an AXMEDIS object. |
| getDomainsRegistered | This method returns the domain where a user is registered |
| adaptDRMRules | Adapts the given license using the given constraints |
| adaptPAR | Adapts the given PAR using the given constraints |
| verifyPAR | Verifies a PAR syntactically against the schemas defined within the PAR. |
| checkPARRules | Verifies if the PAR can be generated according to the parent licenses and PARs. |
| retrieveLicenseModel | This function returns a licenseModel stored in the License DB |
| deleteLicenseModel | This function deletes a licenseModel from the License DB |
| storeLicenseModel | This function inserts a license model in the License DB |

| | |
|--------------------|---|
| updateLicenseModel | This function updates the licenseModel given a licenseModelID and a new description of licenseModel |
| revokeLicense | This function changes the status of a license to revoked in the License DB |
| revokeAddLicense | This function changes the status of a license to revoked and adds the license that substitutes it in the License DB |

4.8.2 WSDL Interface



| DRM Support | |
|-------------|---|
| WSDL | <pre> <wsdl:definitions xmlns:tns="urn:authorisationSupport" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:authorisationSupport" name="authorisationSupport"> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="action" type="xsd:string" nillable="true"/> <xsd:element name="resourceID" type="xsd:string" nillable="true"/> <xsd:element name="authContext" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> </pre> |

| | |
|--|---|
| | <pre> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> <xsd:element name="license" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="license" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="PARs" type="xsd:string" nillable="true"/> <xsd:element name="parentLicense" type="xsd:string" nillable="true"/> <xsd:element name="context" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="userIds" type="xsd:string" nillable="true"/> <xsd:element name="rights" type="xsd:string" nillable="true"/> <xsd:element name="AXOIDs" type="xsd:string" nillable="true"/> <xsd:element name="conditions" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> <xsd:element name="grants" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="ActionLog"> <!-- Defined in AXMEDIS Supervisor --> <xsd:sequence/> </xsd:complexType> </xsd:schema> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayList"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/" xmlns:xns5="http://systinet.com/containers/literal/ms.net"> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:import namespace="http://systinet.com/wsdl/default"/> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="objectVersion" type="xsd:string" nillable="true"/> <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/> <xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/> <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> </pre> |
|--|---|

| | |
|--|---|
| | <pre> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="Vector"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="ArrayOfAnyType"> <xsd:sequence> <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsd/default/" xmlns:xns5="http://systinet.com/containers/literal/ms.net"> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:import namespace="http://systinet.com/wsd/default"/> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="objectVersion" type="xsd:string" nillable="true"/> <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/> <xsd:element name="protectionInfo" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="ProtectionInformation_Response" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="algorithm" type="xns4:Algorithm" nillable="true"/> <xsd:element name="keyLength" type="xsd:int"/> <xsd:element name="parameters" type="xns5:Vector" nillable="true"/> <xsd:element name="KeyAX_Response" type="xns4:KeyAX" nillable="true"/> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/wsd/default/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsd/default/" xmlns:map="http://systinet.com/mapping"> <xsd:complexType name="ProtectionInformation"> <xsd:sequence> <xsd:element name="protectionInfo" type="xsd:string" nillable="true"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="Algorithm"> <xsd:sequence> <xsd:element name="algorithm" type="xsd:int"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="KeyAX"> <xsd:sequence> <xsd:element name="key" type="xsd:base64Binary" nillable="true"/> <xsd:element name="keyByte" type="xsd:base64Binary" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/> <xsd:element name="destinationLanguage" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </pre> |
|--|---|

```

        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayList">
        <xsd:complexContent mixed="false">
            <xsd:extension base="tns:List"/>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/containers/literal/ms.net"
xmlns:xns5="urn:DomainPackage.DomainManager">
    <xsd:import namespace="urn:DomainPackage.DomainManager"/>
    <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
    <xsd:element name="field" type="xsd:string" nillable="true"/>
    <xsd:element name="value" type="xsd:string" nillable="true"/>
    <xsd:element name="ArrayList_Response" type="xns4:ArrayList" nillable="true"/>
    <xsd:element name="myDomain" type="xns5:Domain" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="idDomain" type="xsd:string" nillable="true"/>
    <xsd:element name="myUser" type="xns5:User" nillable="true"/>
    <xsd:element name="myDomain_1" type="xsd:string" nillable="true"/>
    <xsd:element name="userID" type="xsd:string" nillable="true"/>
    <xsd:element name="myDomain_2" type="xsd:string" nillable="true"/>
</xsd:schema>
<xsd:schema targetNamespace="urn:DomainPackage.DomainManager"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:DomainPackage.DomainManager" xmlns:map="http://systinet.com/mapping/">
    <xsd:complexType name="Domain">
        <xsd:sequence>
            <xsd:element name="AXDOM" type="xsd:string" nillable="true"/>
            <xsd:element name="AXID" type="xsd:string" nillable="true"/>
            <xsd:element name="typeOfID" type="xsd:string" nillable="true"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="User">
        <xsd:sequence/>
    </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:types>
    <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
        <xsd:element name="userIDs" type="xsd:string" nillable="true"/>
        <xsd:element name="rights" type="xsd:string" nillable="true"/>
        <xsd:element name="AXOIDs" type="xsd:string" nillable="true"/>
        <xsd:element name="conditions" type="xsd:string" nillable="true"/>
        <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
        <xsd:element name="grants" type="xsd:string" nillable="true"/>
        <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/>
        <xsd:element name="constraints" type="xsd:string" nillable="true"/>
        <xsd:element name="PAR" type="xsd:string" nillable="true"/>
    </xsd:schema>
</wsdl:types>
<wsdl:types>
    <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
        <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
        <xsd:element name="licenseID" type="xsd:string" nillable="true"/>
        <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
        <xsd:element name="license" type="xsd:string" nillable="true"/>
        <xsd:element name="boolean_Response" type="xsd:boolean"/>
        <xsd:element name="licenseModel" type="xsd:string" nillable="true"/>
        <xsd:element name="licModelID" type="xsd:string" nillable="true"/>
        <xsd:element name="licModelNew" type="xsd:string" nillable="true"/>
        <xsd:element name="newLicense" type="xsd:string" nillable="true"/>
    </xsd:schema>
</wsdl:types>

```



```

<wsdl:types>
  <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/"
    elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://systinet.com/xsd/SchemaTypes/">
    <xsd:element name="license" type="xsd:string" nillable="true"/>
    <xsd:element name="boolean_Response" type="xsd:boolean"/>
    <xsd:element name="PARs" type="xsd:string" nillable="true"/>
    <xsd:element name="parentLicense" type="xsd:string" nillable="true"/>
    <xsd:element name="context" type="xsd:string" nillable="true"/>
    <xsd:element name="string_Response" type="xsd:string" nillable="true"/>
    <xsd:element name="PAR" type="xsd:string" nillable="true"/>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="authorisationSupport_authorise_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="authorisationSupport_authorise_1_Request">
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="action" element="ns0:action"/>
  <wsdl:part name="resourceID" element="ns0:resourceID"/>
  <wsdl:part name="authContext" element="ns0:authContext"/>
</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
</wsdl:message>
<wsdl:message name="LicenseManager_retrieveLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseManager_deleteLicense_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="licenseID" element="ns0:licenseID"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseManager_storeLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyTemporalLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="context" element="ns0:context"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyCreatedLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyCreatedLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyTemporalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerificator_verifyLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>

```

```

<wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
<wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request">
  <wsdl:part name="userIDs" element="ns0:userIDs"/>
  <wsdl:part name="rights" element="ns0:rights"/>
  <wsdl:part name="grants" element="ns0:grants"/>
  <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/>
  <wsdl:part name="conditions" element="ns0:conditions"/>
</wsdl:message>
<wsdl:message name="LicenseGenerator_generateDistributorLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/>
<wsdl:message name="ContentConsumption_retrieveActionLogs_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_insertActionLog_Response"/>
<wsdl:message name="ContentConsumption_insertActionLog_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
  <wsdl:part name="actionLog" element="ns0:actionLog"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"/>
<wsdl:message name="ContentConsumption_deleteCacheContent_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
  <wsdl:part name="protectionInfo" element="ns0:protectionInfo"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:ProtectionInformation_Response"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_1_Request">
  <wsdl:part name="algorithm" element="ns0:algorithm"/>
  <wsdl:part name="keyLength" element="ns0:keyLength"/>
  <wsdl:part name="parameters" element="ns0:parameters"/>
</wsdl:message>
<wsdl:message name="ProtectionInfoManager_generateKey_Response">
  <wsdl:part name="response" element="ns0:KeyAX_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="RightsExpressionTranslator_generateTranslation_1_Request">
  <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/>
  <wsdl:part name="destinationLanguage" element="ns0:destinationLanguage"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_1_Request">
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_2"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_1_Request">
  <wsdl:part name="idDomain" element="ns0:idDomain"/>
</wsdl:message>

```


| | |
|--|---|
| | <pre> <wsdl:message name="DomainManager_createDomain_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="DomainManager_updateDomain_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="DomainManager_deleteDomain_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="DomainManager_createDomain_1_Request"> <wsdl:part name="myDomain" element="ns0:myDomain"/> </wsdl:message> <wsdl:message name="DomainManager_unRegistrationRequest_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="DomainManager_updateDomain_1_Request"> <wsdl:part name="myDomain" element="ns0:myDomain"/> </wsdl:message> <wsdl:message name="DomainManager_searchDomains_Response"> <wsdl:part name="response" element="ns0:ArrayList_Response"/> </wsdl:message> <wsdl:message name="DomainManager_registrationRequest_1_Request"> <wsdl:part name="myUser" element="ns0:myUser"/> <wsdl:part name="myDomain" element="ns0:myDomain_1"/> </wsdl:message> <wsdl:message name="DomainManager_searchDomains_1_Request"> <wsdl:part name="field" element="ns0:field"/> <wsdl:part name="value" element="ns0:value"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptPAR_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptDRMRules_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptPAR_1_Request"> <wsdl:part name="PAR" element="ns0:PAR"/> <wsdl:part name="constraints" element="ns0:constraints"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateDistributorLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="grants" element="ns0:grants"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseGenerator_adaptDRMRules_1_Request"> <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/> <wsdl:part name="constraints" element="ns0:constraints"/> </wsdl:message> <wsdl:message name="LicenseGenerator_generateFinalLicense_1_Request"> <wsdl:part name="userIDs" element="ns0:userIDs"/> <wsdl:part name="rights" element="ns0:rights"/> <wsdl:part name="AXOIDs" element="ns0:AXOIDs"/> <wsdl:part name="conditions" element="ns0:conditions"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeLicense_1_Request"> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> </pre> |
|--|---|

| | |
|--|---|
| | <pre> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicenseModel_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_updateLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> <wsdl:part name="licModelNew" element="ns0:licModelNew"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeAddLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_revokeAddLicense_1_Request"> <wsdl:part name="licenseID" element="ns0:licenseID"/> <wsdl:part name="newLicense" element="ns0:newLicense"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicenseModel_1_Request"> <wsdl:part name="licenseModel" element="ns0:licenseModel"/> </wsdl:message> <wsdl:message name="LicenseManager_revokelicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicenseModel_1_Request"> <wsdl:part name="licModelID" element="ns0:licModelID"/> </wsdl:message> <wsdl:message name="LicenseManager_updateLicenseModel_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_deleteLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseManager_retrieveLicense_1_Request"> <wsdl:part name="AXOID" element="ns0:AXOID"/> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseManager_storeLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> </wsdl:message> <wsdl:message name="LicenseVerifier_verifyCreatedLicense_1_Request"> <wsdl:part name="license" element="ns0:license"/> <wsdl:part name="PARs" element="ns0:PARs"/> <wsdl:part name="parentLicense" element="ns0:parentLicense"/> </wsdl:message> <wsdl:message name="LicenseVerifier_checkPARRules_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVerifier_checkPARRules_1_Request"> <wsdl:part name="PAR" element="ns0:PAR"/> <wsdl:part name="PARs" element="ns0:PARs"/> <wsdl:part name="parentLicense" element="ns0:parentLicense"/> </wsdl:message> <wsdl:message name="LicenseVerifier_verifyCreatedLicense_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="LicenseVerifier_verifyPAR_1_Request"> <wsdl:part name="PAR" element="ns0:PAR"/> </wsdl:message> <wsdl:message name="LicenseVerifier_verifyTemporalLicense_Response"> </pre> |
|--|---|

```

<wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVericator_verifyLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseVericator_verifyPAR_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVericator_verifyTemporalLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="context" element="ns0:context"/>
</wsdl:message>
<wsdl:message name="LicenseVericator_verifyLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:portType name="authorisationSupport">
  <wsdl:operation name="authorise" parameterOrder="licenseID userID action resourceID authContext">
    <wsdl:input message="tns:authorisationSupport_authorise_1_Request"/>
    <wsdl:output message="tns:authorisationSupport_authorise_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="ContentConsumption">
  <wsdl:operation name="insertActionLog" parameterOrder="AXOID objectVersion protectionStamp
actionLog">
    <wsdl:input message="tns:ContentConsumption_insertActionLog_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_insertActionLog_Response"/>
  </wsdl:operation>
  <wsdl:operation name="retrieveActionLogs">
    <wsdl:input message="tns:ContentConsumption_retrieveActionLogs_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_retrieveActionLogs_Response"/>
  </wsdl:operation>
  <wsdl:operation name="deleteCacheContent">
    <wsdl:input message="tns:ContentConsumption_deleteCacheContent_1_Request"/>
    <wsdl:output message="tns:ContentConsumption_deleteCacheContent_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="ProtectionInfoManager">
  <wsdl:operation name="insertProtectionInfo" parameterOrder="AXOID objectVersion protectionStamp
protectionInfo">
    <wsdl:input message="tns:ProtectionInfoManager_insertProtectionInfo_1_Request"/>
    <wsdl:output message="tns:ProtectionInfoManager_insertProtectionInfo_Response"/>
  </wsdl:operation>
  <wsdl:operation name="retrieveProtectionInfo" parameterOrder="AXOID objectVersion
protectionStamp">
    <wsdl:input message="tns:ProtectionInfoManager_retrieveProtectionInfo_1_Request"/>
    <wsdl:output message="tns:ProtectionInfoManager_retrieveProtectionInfo_Response"/>
  </wsdl:operation>
  <wsdl:operation name="generateKey" parameterOrder="algorithm keyLength parameters">
    <wsdl:input message="tns:ProtectionInfoManager_generateKey_1_Request"/>
    <wsdl:output message="tns:ProtectionInfoManager_generateKey_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="RightsExpressionTranslator">
  <wsdl:operation name="generateTranslation" parameterOrder="sourceLicense destinationLanguage">
    <wsdl:input message="tns:RightsExpressionTranslator_generateTranslation_1_Request"/>
    <wsdl:output message="tns:RightsExpressionTranslator_generateTranslation_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="DomainManager">
  <wsdl:operation name="searchDomains" parameterOrder="field value">
    <wsdl:input message="tns:DomainManager_searchDomains_1_Request"/>
    <wsdl:output message="tns:DomainManager_searchDomains_Response"/>
  </wsdl:operation>
  <wsdl:operation name="createDomain" parameterOrder="myDomain">
    <wsdl:input message="tns:DomainManager_createDomain_1_Request"/>
    <wsdl:output message="tns:DomainManager_createDomain_Response"/>
  </wsdl:operation>
  <wsdl:operation name="deleteDomain" parameterOrder="idDomain">
    <wsdl:input message="tns:DomainManager_deleteDomain_1_Request"/>
    <wsdl:output message="tns:DomainManager_deleteDomain_Response"/>
  </wsdl:operation>
</wsdl:portType>

```

```

</wsdl:operation>
<wsdl:operation name="updateDomain" parameterOrder="myDomain">
  <wsdl:input message="tns:DomainManager_updateDomain_1_Request"/>
  <wsdl:output message="tns:DomainManager_updateDomain_Response"/>
</wsdl:operation>
<wsdl:operation name="registrationRequest" parameterOrder="myUser myDomain">
  <wsdl:input message="tns:DomainManager_registrationRequest_1_Request"/>
  <wsdl:output message="tns:DomainManager_registrationRequest_Response"/>
</wsdl:operation>
<wsdl:operation name="unRegistrationRequest" parameterOrder="userID myDomain">
  <wsdl:input message="tns:DomainManager_unRegistrationRequest_1_Request"/>
  <wsdl:output message="tns:DomainManager_unRegistrationRequest_Response"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseGenerator">
  <wsdl:operation name="generateFinalLicense" parameterOrder="userIDs rights AXOIDs conditions">
    <wsdl:input message="tns:LicenseGenerator_generateFinalLicense_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_generateFinalLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="generateDistributorLicense" parameterOrder="userIDs rights grants AXOIDs
conditions">
    <wsdl:input message="tns:LicenseGenerator_generateDistributorLicense_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_generateDistributorLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="adaptDRMRules" parameterOrder="sourceLicense constraints">
    <wsdl:input message="tns:LicenseGenerator_adaptDRMRules_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_adaptDRMRules_Response"/>
  </wsdl:operation>
  <wsdl:operation name="adaptPAR" parameterOrder="PAR constraints">
    <wsdl:input message="tns:LicenseGenerator_adaptPAR_1_Request"/>
    <wsdl:output message="tns:LicenseGenerator_adaptPAR_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="LicenseManager">
  <wsdl:operation name="retrieveLicense" parameterOrder="AXOID licenseID">
    <wsdl:input message="tns:LicenseManager_retrieveLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_retrieveLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseManager_storeLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_storeLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="storeLicenseModel" parameterOrder="licenseModel">
    <wsdl:input message="tns:LicenseManager_storeLicenseModel_1_Request"/>
    <wsdl:output message="tns:LicenseManager_storeLicenseModel_Response"/>
  </wsdl:operation>
  <wsdl:operation name="deleteLicense" parameterOrder="AXOID licenseID">
    <wsdl:input message="tns:LicenseManager_deleteLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_deleteLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="retrieveLicenseModel" parameterOrder="licModelID">
    <wsdl:input message="tns:LicenseManager_retrieveLicenseModel_1_Request"/>
    <wsdl:output message="tns:LicenseManager_retrieveLicenseModel_Response"/>
  </wsdl:operation>
  <wsdl:operation name="updateLicenseModel" parameterOrder="licModelID licModelNew">
    <wsdl:input message="tns:LicenseManager_updateLicenseModel_1_Request"/>
    <wsdl:output message="tns:LicenseManager_updateLicenseModel_Response"/>
  </wsdl:operation>
  <wsdl:operation name="deleteLicenseModel" parameterOrder="licModelID">
    <wsdl:input message="tns:LicenseManager_deleteLicenseModel_1_Request"/>
    <wsdl:output message="tns:LicenseManager_deleteLicenseModel_Response"/>
  </wsdl:operation>
  <wsdl:operation name="revokeLicense" parameterOrder="licenseID">
    <wsdl:input message="tns:LicenseManager_revokeLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_revokeLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="revokeAddLicense" parameterOrder="licenseID newLicense">
    <wsdl:input message="tns:LicenseManager_revokeAddLicense_1_Request"/>
    <wsdl:output message="tns:LicenseManager_revokeAddLicense_Response"/>
  </wsdl:operation>
</wsdl:portType>

```

```
</wsdl:portType>
<wsdl:portType name="LicenseVerifier">
  <wsdl:operation name="verifyLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseVerifier_verifyLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyCreatedLicense" parameterOrder="license PARs parentLicense">
    <wsdl:input message="tns:LicenseVerifier_verifyCreatedLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyCreatedLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyTemporalLicense" parameterOrder="license context">
    <wsdl:input message="tns:LicenseVerifier_verifyTemporalLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyTemporalLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyPAR" parameterOrder="PAR">
    <wsdl:input message="tns:LicenseVerifier_verifyPAR_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyPAR_Response"/>
  </wsdl:operation>
  <wsdl:operation name="checkPARRules" parameterOrder="PAR PARs parentLicense">
    <wsdl:input message="tns:LicenseVerifier_checkPARRules_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_checkPARRules_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="authorisationSupport" type="tns:authorisationSupport">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="authorise">
    <map:java-operation name="authorise"
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
    <soap:operation
soapAction="urn:authorisationSupportauthorisationSupport#authorise#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
  <wsdl:binding name="ContentConsumption" type="tns:ContentConsumption">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="insertActionLog">
      <map:java-operation name="insertActionLog"
signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xBY3Rpb25Mb2c7KVY=" style="document"/>
      <soap:operation
soapAction="urn:ContentConsumptionContentConsumption#insertActionLog#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xBY3Rpb25Mb2c7KVY=" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="retrieveActionLogs">
      <map:java-operation name="retrieveActionLogs"
signature="KCIMamF2YS91dGlsLFYcmF5TGldDs=" style="document"/>
      <soap:operation
soapAction="urn:ContentConsumptionContentConsumption#retrieveActionLogs#KCIMamF2YS91dGlsLFYcmF5TGldDs=" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

```


| | |
|--|---|
| | <pre> </wsdl:operation> <wsdl:operation name="deleteCacheContent"> <map:java-operation name="deleteCacheContent" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#deleteCacheContent#KClA" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="ProtectionInfoManager" type="tns:ProtectionInfoManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="insertProtectionInfo"> <map:java-operation name="insertProtectionInfo" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5n O0xQcm90ZWN0aW9uSW5mb3JtYXRpb247KVo="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#insertProtectionInfo#KExqYXZhL2xhbmvcU3R yaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xQcm90ZWN0aW9uSW5mb3JtYX Rpb247KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo"> <map:java-operation name="retrieveProtectionInfo" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5n OylMUHJvdGVjdGlwbkluZm9ybWF0aW9uOw==" /> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#retrieveProtectionInfo#KExqYXZhL2xhbmvcU3 RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMUHJvdGVjdGlwbkluZm9ybWF0 aW9uOw==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateKey"> <map:java-operation name="generateKey" signature="KExBbGdvcmI0aG07SUxqYXZhL3V0aWwvVmVjdG9yOylMS2V5QVg7"/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#generateKey#KExBbGdvcmI0aG07SUxqYXZh L3V0aWwvVmVjdG9yOylMS2V5QVg7" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="RightsExpressionTranslator" type="tns:RightsExpressionTranslator"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="generateTranslation"> <map:java-operation name="generateTranslation" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmIu Zzs="/> <soap:operation soapAction="urn:translator.RightsExpressionTranslatorRightsExpressionTranslator#generateTranslation#KExq YXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=" style="document"/> <wsdl:input> </pre> |
|--|---|

| | |
|--|--|
| | <pre> <soap:body use="literal"/> </wsdl:input> </wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="DomainManager" type="tns:DomainManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="searchDomains"> <map:java-operation name="searchDomains" signature="KExqYXZhL2xhbmvcvU3RyaW5nO0xqYXZhL2xhbmvcvU3RyaW5nOyIMamF2YS91dGlsL0FycmF5TGldDs="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#searchDomains#KExqYXZhL2xhbmvcvU3RyaW5nO0xqYXZhL2xhbmvcvU3RyaW5nOyIMamF2YS91dGlsL0FycmF5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="createDomain"> <map:java-operation name="createDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWFpbjSpWg="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#createDomain#KExEb21haW5QYWNRyWdlL0RvbWFpbjSpWg=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteDomain"> <map:java-operation name="deleteDomain" signature="KExqYXZhL2xhbmvcvU3RyaW5nOyIa"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#deleteDomain#KExqYXZhL2xhbmvcvU3RyaW5nOyIa" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="updateDomain"> <map:java-operation name="updateDomain" signature="KExEb21haW5QYWNRyWdlL0RvbWFpbjSpWg="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#updateDomain#KExEb21haW5QYWNRyWdlL0RvbWFpbjSpWg=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="registrationRequest"> <map:java-operation name="registrationRequest" signature="KExEb21haW5QYWNRyWdlL1VzZXI7TGphdmEvdGFuZy9TdHJpbmc7KVo="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#registrationRequest#KExEb21haW5QYWNRyWdlL1VzZXI7TGphdmEvdGFuZy9TdHJpbmc7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> </pre> |
|--|--|

| | |
|--|---|
| | <pre> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="unRegistrationRequest"> <map:java-operation name="unRegistrationRequest" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#unRegistrationRequest#KExqYXZhL2xhb mcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseGenerator" type="tns:LicenseGenerator"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="generateFinalLicense"> <map:java-operation name="generateFinalLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#generateFinalLicense#KExqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2 YS9sYW5nL1N0cmZsZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateDistributorLicense"> <map:java-operation name="generateDistributorLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5n O0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#generateDistributorLicense#KExqYXZhL2xhbmcvU3Rya W5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqY XZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="adaptDRMRules"> <map:java-operation name="adaptDRMRules" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs Zs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptDRMRules#KExqYXZhL2xhbmcvU3RyaW5nO0xq YXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZsZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="adaptPAR"> <map:java-operation name="adaptPAR" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs Zs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptPAR#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL </pre> |
|--|---|

| | |
|--|--|
| | <pre> 2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:binding name="LicenseManager" type="tns:LicenseManager"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="retrieveLicense"> <map:java-operation name="retrieveLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs="/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#retrieveLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeLicense"> <map:java-operation name="storeLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#storeLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="storeLicenseModel"> <map:java-operation name="storeLicenseModel" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#storeLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteLicense"> <map:java-operation name="deleteLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#deleteLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveLicenseModel"> <map:java-operation name="retrieveLicenseModel" signature="KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs="/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#retrieveLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs=" style="document"/> <wsdl:input> </pre> |
|--|--|

```

        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="updateLicenseModel">
    <map:java-operation name="updateLicenseModel"
signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/>
    <soap:operation
soapAction="urn:LicenseManagerLicenseManager#updateLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nO0x
qYXZhL2xhbmvcU3RyaW5nOyla" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="deleteLicenseModel">
    <map:java-operation name="deleteLicenseModel"
signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/>
    <soap:operation
soapAction="urn:LicenseManagerLicenseManager#deleteLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOyla
" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="revokeLicense">
    <map:java-operation name="revokeLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/>
    <soap:operation
soapAction="urn:LicenseManagerLicenseManager#revokeLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="revokeAddLicense">
    <map:java-operation name="revokeAddLicense"
signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/>
    <soap:operation
soapAction="urn:LicenseManagerLicenseManager#revokeAddLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xq
YXZhL2xhbmvcU3RyaW5nOyla" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="LicenseVerifier" type="tns:LicenseVerifier">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="verifyLicense">
        <map:java-operation name="verifyLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/>
        <soap:operation
soapAction="urn:LicenseVerifierLicenseVerifier#verifyLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

| | |
|--|--|
| | <pre> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyCreatedLicense"> <map:java-operation name="verifyCreatedLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#verifyCreatedLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyTemporalLicense"> <map:java-operation name="verifyTemporalLicense" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs="/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#verifyTemporalLicense#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="verifyPAR"> <map:java-operation name="verifyPAR" signature="KExqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#verifyPAR#KExqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="checkPARRules"> <map:java-operation name="checkPARRules" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseVericatorLicenseVericator#checkPARRules#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="authorisationSupport"> <wsdl:port name="authorisationSupport" binding="tns:authorisationSupport"> <soap:address location="http://cjlal1:6060/authorisationSupport"/> </wsdl:port> </wsdl:service> <wsdl:service name="ContentConsumption"> <wsdl:port name="ContentConsumption" binding="tns:ContentConsumption"> <soap:address location="http://efpc037:6060/ContentConsumption"/> </wsdl:port> </wsdl:service> <wsdl:service name="ProtectionInfoManager"> <wsdl:port name="ProtectionInfoManager" binding="tns:ProtectionInfoManager"> <soap:address location="http://efpc037:6060/ProtectionInfoManager"/> </wsdl:port> </wsdl:service> </pre> |
|--|--|

| | |
|--|--|
| | <pre> </wsdl:port> </wsdl:service> <wsdl:service name="RightsExpressionTranslator"> <wsdl:port name="RightsExpressionTranslator" binding="tns:RightsExpressionTranslator"> <soap:address location="http://EFPC016:6060/RightsExpressionTranslator/"> </wsdl:port> </wsdl:service> <wsdl:service name="DomainManager"> <wsdl:port name="DomainManager" binding="tns:DomainManager"> <soap:address location="http://EFPC016:6060/DomainManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseGenerator"> <wsdl:port name="LicenseGenerator" binding="tns:LicenseGenerator"> <soap:address location="http://efpc037:6060/LicenseGenerator/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseManager"> <wsdl:port name="LicenseManager" binding="tns:LicenseManager"> <soap:address location="http://efpc037:6060/LicenseManager/"> </wsdl:port> </wsdl:service> <wsdl:service name="LicenseVerifier"> <wsdl:port name="LicenseVerifier" binding="tns:LicenseVerifier"> <soap:address location="http://efpc037:6060/LicenseVerifier/"> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|--|--|

4.9 Authorisation Support (FUPF)

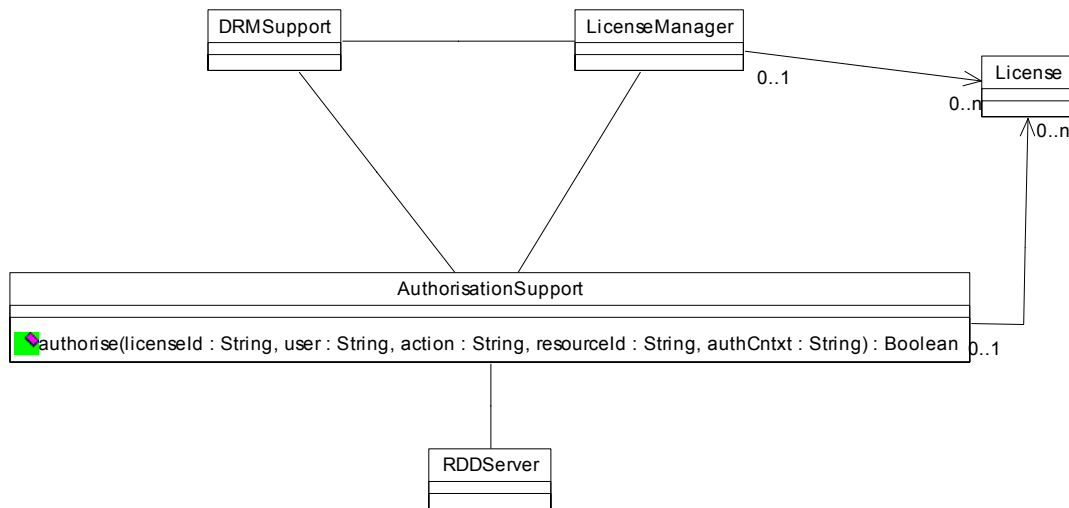
This module provides the functionality of authorising AXMEDIS users to perform actions over AXMEDIS objects. It has access to the license repository and informs the AXMEDIS Supervisor of the result of its actions.

| Module Profile | | |
|--------------------------------|---------------------------------|---|
| Authorisation Support | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | C++/Java | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| DRM support | | |

| | | |
|-------------------|-----------------------------------|--|
| License manager | | |
| RDD Server | | |
| File Formats Used | Shared with | File format name or reference to a section |
| XML | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| NA | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.9.1 Architecture of Authorisation Support module

Next figure shows the UML diagram of the Authorisation Support module.



Authorisation Support class diagram

| Authorisation Support | |
|-----------------------|---|
| Methods | Description |
| authorise | This function authorises AXMEDIS users to perform actions over AXMEDIS objects. It proves that a user has the appropriate license that grants him to exercise a right over a resource if the conditions are met based on the execution context of the client. |

4.9.2 WSDL Interface

[illegible]

| | |
|-------------------------|---|
| | <pre> <wn0:licenseID i:type="d:string">LIC1</wn0:licenseID> <wn0:userID i:type="d:string">AXUID1</wn0:userID> <wn0:action i:type="d:string">play</wn0:action> <wn0:resourceID i:type="d:string">AXOID1</wn0:resourceID> <wn0:authContext i:type="d:string">context1</wn0:authContext> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:string licenseID: identifier of the license (AXLID)</p> <p>xsd:string userID: identifier of the AXMEDIS user (AXUID)</p> <p>xsd:string action: right to exercise</p> <p>xsd:string resourceID: resource (AXOID) to be used</p> <p>xsd:string authContext: properties of the user</p> |
| Output parameters | <p>xsd:Boolean: true if the user is authorised, false if not.</p> |

4.10 RDD Server (FUPF)

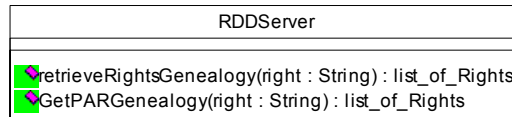
| Module Profile | | |
|--------------------------------|---------------------------------|---|
| RDD Server | | |
| Executable or Library(Support) | Support Library | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Authorisation Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |

| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
|----------------|-----------------------------------|--|
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

RDD Server keeps track of the meaning of the different actions and rights that can be applied to AXMEDIS governed objects.

4.10.1 Architecture of RDD Server module

Next figure shows the UML diagram of the RDDServer module.



RDD Server class diagram

| DRM Support | |
|-------------------------|---|
| Methods | Description |
| retrieveRightsGenealogy | Retrieve the right genealogy, parent rights of the input right, according to RDD hierarchy of the rights. |
| getPARGenealogy | Retrieve the right genealogy, child rights of the input right, according to the RDD rights hierarchy. |

4.10.2 WSDL Interface

| RDD Server | |
|------------|---|
| WSDL | <pre> <wsdl:definitions name='RDDServer' targetNamespace='urn:RDDServer' xmlns:tns='urn:RDDServer' xmlns:ns0='http://systinet.com/xsd/SchemaTypes/' xmlns:map='http://systinet.com/mapping/' xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/' xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'> <wsdl:types> <xsd:schema targetNamespace='http://systinet.com/xsd/SchemaTypes/' elementFormDefault='qualified' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:tns='http://systinet.com/xsd/SchemaTypes/'> <xsd:element name='right' type='xsd:string' nillable='true'/> <xsd:element name='string_Response' type='xsd:string' nillable='true'/> </xsd:schema> </wsdl:types> <wsdl:message name='RDDServer_GetParGenealogy_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='RDDServer_GetParGenealogy_1_Request'> <wsdl:part name='right' element='ns0:right'/> </wsdl:message> <wsdl:message name='RDDServer_retrieveRightsGenealogy_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='RDDServer_retrieveRightsGenealogy_1_Request'> </pre> |

| | |
|-------------------------|---|
| | <pre> <wsdl:part name='right' element='ns0:right'/> </wsdl:message> <wsdl:portType name='RDDServer'> <wsdl:operation name='retrieveRightsGenealogy' parameterOrder='right'> <wsdl:input message='tns:RDDServer_retrieveRightsGenealogy_1_Request'/> <wsdl:output message='tns:RDDServer_retrieveRightsGenealogy_Response'/> </wsdl:operation> <wsdl:operation name='GetParGenealogy' parameterOrder='right'> <wsdl:input message='tns:RDDServer_GetParGenealogy_1_Request'/> <wsdl:output message='tns:RDDServer_GetParGenealogy_Response'/> </wsdl:operation> </wsdl:portType> <wsdl:binding name='RDDServer' type='tns:RDDServer'> <soap:binding transport='http://schemas.xmlsoap.org/soap/http' style='document'/> <wsdl:operation name='retrieveRightsGenealogy'> <map:java-operation name='retrieveRightsGenealogy' signature='KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs='/> <soap:operation soapAction='urn:RDDServerRDDServer#retrieveRightsGenealogy#KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='GetParGenealogy'> <map:java-operation name='GetParGenealogy' signature='KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs='/> <soap:operation soapAction='urn:RDDServerRDDServer#GetParGenealogy#KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name='RDDServer'> <wsdl:port name='RDDServer' binding='tns:RDDServer'> <soap:address location='http://cjlal1:6060/RDDServer'/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
| Method | retrieveRightsGenealogy |
| Description | retrieve the right genealogy, parent rights of the input right, according to RDD hierarchy of the rights. |
| Request Sample Message | <pre> <?xml version='1.0' encoding='UTF-8' standalone='yes'?> <e:Envelope xmlns:e='http://schemas.xmlsoap.org/soap/envelope/' xmlns:d='http://www.w3.org/2001/XMLSchema' xmlns:i='http://www.w3.org/2001/XMLSchema-instance' xmlns:wn0='http://systinet.com/xsd/SchemaTypes/'> <e:Body> <wn0:right i:type='d:string'>play</wn0:right> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version='1.0' encoding='UTF-8' standalone='yes'?> <e:Envelope xmlns:e='http://schemas.xmlsoap.org/soap/envelope/' xmlns:d='http://www.w3.org/2001/XMLSchema' xmlns:i='http://www.w3.org/2001/XMLSchema-instance' xmlns:wn0='http://systinet.com/xsd/SchemaTypes/'> <e:Body> <wn0:string_Response i:type='d:string'>&lt;right&gt;render&lt;right&gt;&lt;right&gt;perform&lt;right& &gt;</wn0:string_Response> </e:Body> </e:Envelope> </pre> |

| | |
|-------------------------|--|
| Input parameters | xsd:string right: Right to be consulted about its parents |
| Output parameters | xsd:string: list of parent rights |
| Method | GetPARGenealogy |
| Description | retrieve the right genealogy, child rights of the input right, according to the RDD rights hierarchy. |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:right i:type="d:string">modify</wn0:right> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:right i:type="d:string">modify</wn0:right> </e:Body> </e:Envelope></pre> |
| Input parameters | xsd:string right: Right to be consulted about its childs |
| Output parameters | xsd:string: list of child rights |

4.11 Secure Communication Support (FUPF)

This module is the responsible of providing a secure communication between two parties.

| Module Profile | | |
|--------------------------------|---------------------------------|---|
| Secure Communication Support | | |
| Executable or Library(Support) | Library (Support) | |
| Single Thread or Multithread | Single Thread | |
| Language of Development | C/C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | PC (Linux / Windows) | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXCS | | |
| DRM Support | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| Any | | |
| | | |
| | | |
| User Interface | Development model, language, | Library used for the development, |

| | | |
|----------------|---------------------------------|--|
| | etc. | platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| Openssl | | |
| | | |
| | | |

4.11.1 Description of the module

This module provides the functionality needed to access to information stored in the Local Cache. This module is internally used by the DRM support corresponding to each PMS to stored information that is used for authorising user to perform actions when it is not connected.

The information managed by secure cache manager has to be stored in a secure way, in order to avoid as much as possibly user manipulation of the action logs and general status information. To do so, openssl cryptographic functions will be used.

Not all the information has to be ciphered before storing it, as then it will be very difficult to find it. For instance, the AXOID should be in clear when we want to request the protection information associated to an object, but the protection information itself should be ciphered for storing it.

The most efficient way of performing the ciphering is to use symmetric ciphering. The symmetric key cannot be stored in the user's device, but it should be calculated when some ciphering operation is requested.

The API of OpenSSL serves as an interface to a range of security functions. In this module we will use the ones that provide cryptographic functionality. In next section, OpenSSL library is explained briefly.

4.11.1.1 OpenSSL

The OpenSSL Project (<http://www.openssl.org>) is a collaborative effort to develop a robust, commercial-grade, full-featured and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan and develop the OpenSSL toolkit and its related documentation.

OpenSSL is based on the SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit is licensed under an Apache-style licence, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions.

4.11.1.2 Windows Version of OpenSSL

The Windows version of OpenSSL library can be downloaded from <http://www.slproweb.com/products/Win32OpenSSL.html>

4.11.1.3 Secure session functionalities offered by OpenSSL

The OpenSSL ssl library implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. It provides a rich API which is documented in <http://www.openssl.org/docs/ssl/ssl.html>.

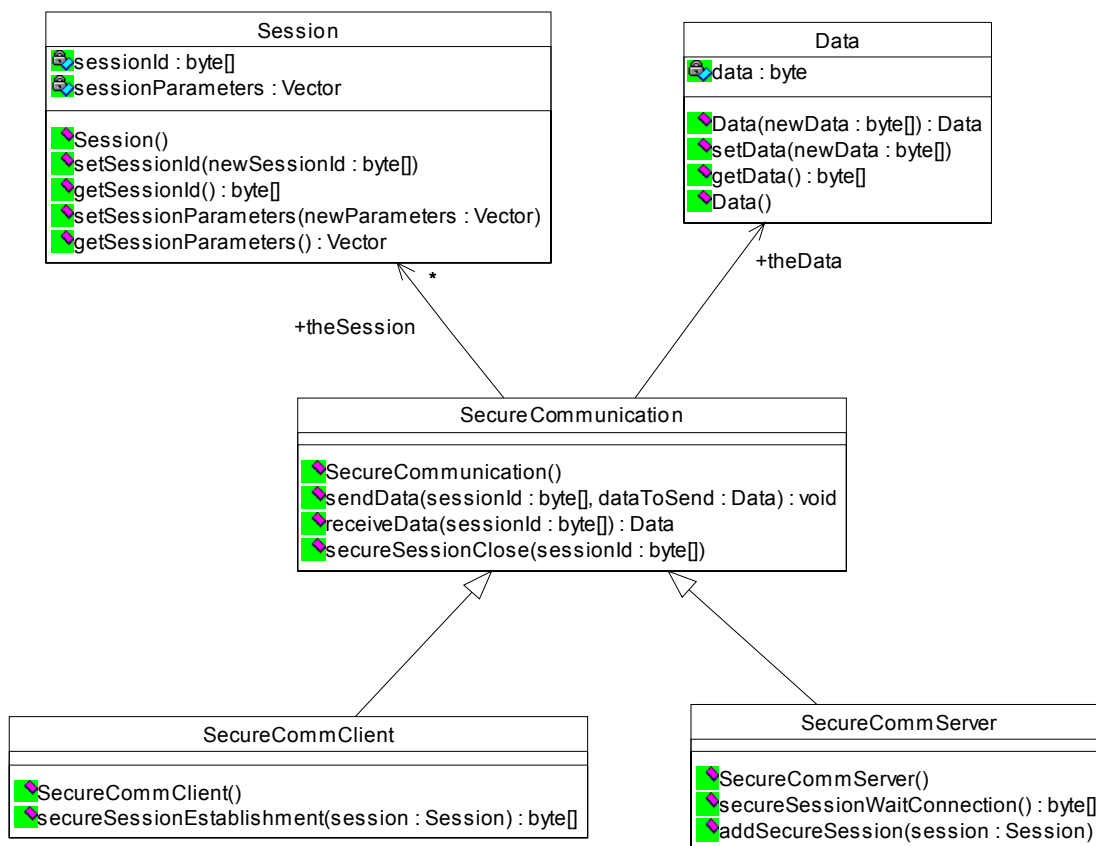
At first the library must be initialized, using `SSL_library_init()`. Then an `SSL_CTX` object is created as a framework to establish TLS/SSL enabled connections (see `SSL_CTX_new()`). Various options regarding certificates, algorithms etc. can be set in this object.

When a network connection has been created, it can be assigned to an SSL object. After the SSL object has been created using `SSL_new()`, `SSL_set_fd()` or `SSL_set_bio()` can be used to associate the network connection with the object.

Then the TLS/SSL handshake is performed using `SSL_accept()` or `SSL_connect()` respectively. `SSL_read()` and `SSL_write()` are used to read and write data on the TLS/SSL connection. `SSL_shutdown()` can be used to shut down the TLS/SSL connection.

4.11.2 Architecture of the Secure communication support module

Next figure shows the UML diagram describing the Secure Communication Support module. Nevertheless, this module is based on OpenSSL, so it is not object oriented in nature, but C++ interface could be provided.



Secure communication support class diagram

`SecureCommunication` class models the common methods between client and server of a secure communication. It provides a list of sessions, that can be managed using the `sessionId`.

`SecureCommClient` offers the client side functionality.

`SecureCommServer` provides the server side functionality, allowing to wait for connections and adding new sessions when needed.

Session and Data classes are used to manage the secure session and the data to be sent through the secure sessions, respectively.

| SecureCommunication | |
|----------------------------|---|
| Method | sendData |
| Description | Sends the given Data through the session identified by sessionId |
| Input parameters | SessionId: byte [], the identifier of the session to be used for the communication DataToSend: Data, the data to be transferred through the secure session |
| Output parameters | None |
| Method | receiveData |
| Description | Receives the given Data through the session identified by sessionId |
| Input parameters | SessionId: byte [], the identifier of the session to be used for the communication |
| Output parameters | Data, the data received through the secure session |
| Method | secureSessionClose |
| Description | This method closes an open session. It will close the underlying SSL session |
| Input parameters | SessionId: byte [], the identifier of the session to be closed |
| Output parameters | None |

| SecureCommClient | |
|-------------------------|---|
| Method | secureSessionEstablishment |
| Description | This method tries to establish a secure session with the parameters inside the Session that is passed as parameter. If it is successful, then it returns the identifier of the newly created session. Data sending and reception is done through this session by using the corresponding identifier |
| Input parameters | session: Session, an instance of session class with the preferred parameters to be used for the secure session |
| Output parameters | SessionId: byte [], the identifier of the session to be used for the communication |

| SecureCommServer | |
|-------------------------|---|
| Method | secureSessionWaitConnection |
| Description | This method is for waiting for connections in the server side. When a new connection is established, it returns the identifier of the new session, that is stored into the current connected sessions list. |
| Input parameters | None |
| Output parameters | SessionId: byte [], the identifier of the new session established |
| Method | addSecureSession |
| Description | This method allows the insertion of a new session into the current connected sessions list. |
| Input parameters | session: Session, an instance of session class that has to be added to the session list |
| Output parameters | None |

| Session | |
|----------------|---|
| Method | getSessionId |
| Description | This method retrieves the identifier of this secure session |

| | |
|-------------------|---|
| Input parameters | None |
| Output parameters | byte[], the session id |
| Method | setSessionId |
| Description | This method stores the identifier of this secure session |
| Input parameters | newSessionId: byte[], the new session identifier |
| Output parameters | None |
| Method | getSessionParameters |
| Description | This method retrieves the parameters of the secure session |
| Input parameters | None |
| Output parameters | Vector: the current parameters of the session |
| Method | setSessionParameters |
| Description | This method stores the parameters of the secure session. They will depend on the openssl options to be used |
| Input parameters | newParameters: Vector, the parameters to be used in the session |
| Output parameters | None |

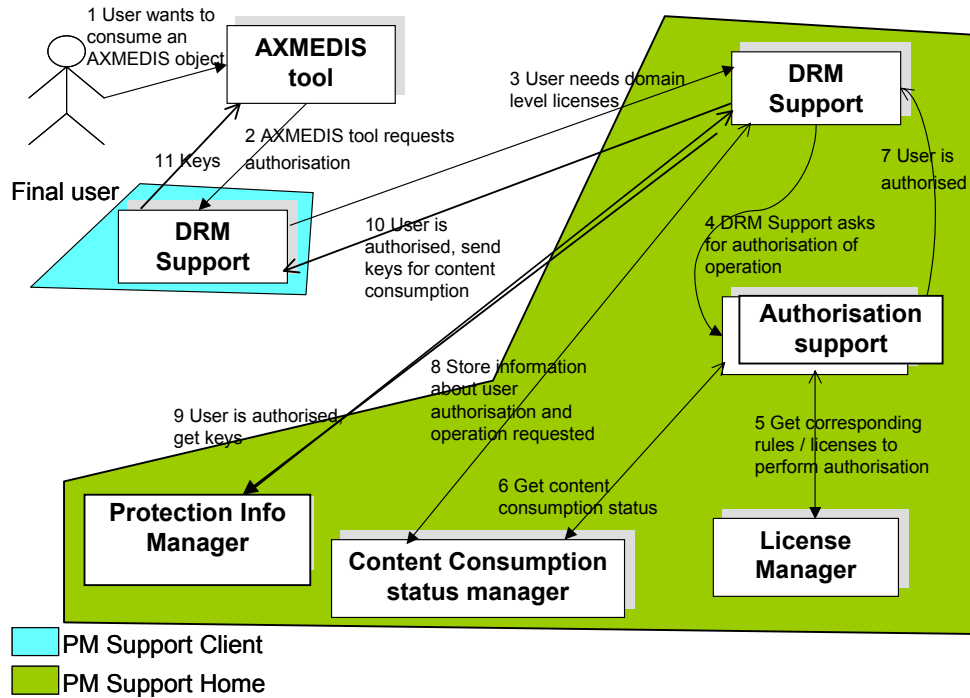
| Data | |
|-------------------|--|
| Method | Data |
| Description | Constructor of the class which receives as parameter the data to be sent/received. |
| Input parameters | NewData: byte[], an array of bytes containing the data |
| Output parameters | A new instance of the Data class |
| Method | getData |
| Description | This method requests the data stored inside this class. |
| Input parameters | None |
| Output parameters | byte[], the byte array representing the data contained inside this class |
| Method | setData |
| Description | This method allows setting new data inside this class. |
| Input parameters | byte[], the byte array representing the data contained inside this class |
| Output parameters | None |

4.12 Content Consumption Status (FUPF)

Content Consumption status module keeps track of the actions performed by the user when he is working in an unconnected environment.

4.12.1 Content Consumption Scenario

The scenario describes the use of content consumption status in PMS Domain Home.



Use of content consumption status

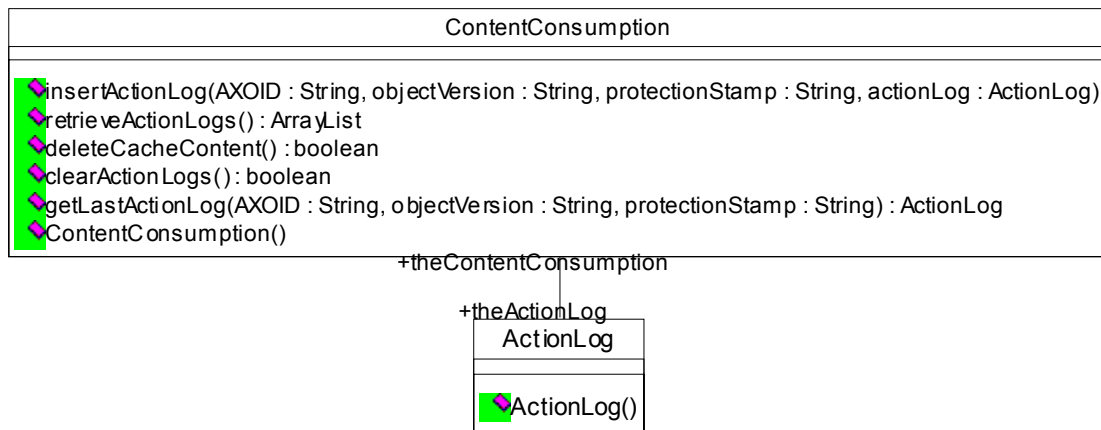
| Module Profile | | |
|--------------------------------|---------------------------------|---|
| Content Consumption Status | | |
| Executable or Library(Support) | Library | |
| Single Thread or Multithread | Single thread | |
| Language of Development | C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | PC (Linux/Windows) | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| DRM Support | | |
| Authorisation Support | | |
| License Interpreter | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |

| | | |
|----------------|-----------------------------------|--|
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

This module provides content consumption caching functionalities.

4.12.2 Architecture of the module

The following figure shows the UML diagram of the Content Consumption module.



Content Consumption class diagram

ContentConsumption:

This module provides the functionality regarding storage of status information regarding objects usage inside the Local Cache Info by the Secure Cache Manager. This module is used by the DRM support.

| ContentConsumption | |
|--------------------|---|
| Method | Description |
| insertActionLog | This method inserts an action log inside the local cache info through the secure cache manager. The action log is identified by the AXMEDIS Object, Version and protection stamp. |
| retrieveActionLogs | This method retrieves all the action logs inside the local cache info when the user connects to the PMS server in order to verify and synchronise the actions performed off-line with the previously performed actions. |
| deleteCacheContent | This method is for deleting the contents of the cache. It can be used when the tool cannot be verified because of illegal manipulation. |
| clearActionLogs | Deletes action logs from the cache, after positive authorisation of the user in the connected environment |
| getLastActionLog | Returns the last action log |

| | ContentConsumption |
|------|---|
| WSDL | <pre> <?xml version="1.0"?> <wsdl:definitions xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:tns="urn:ContentConsumption" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:ContentConsumption" name="ContentConsumption"> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/wsdl/default/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/wsdl/default/" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="ActionLog"> <xsd:annotation> <xsd:appinfo> <map:java-type name="ActionLog"/> </xsd:appinfo> </xsd:annotation> <xsd:sequence/> </xsd:complexType> </xsd:schema> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayList"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="LinkedList"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="Vector"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="Set"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="HashSet"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:Set"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="SortedSet"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:Set"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="TreeSet"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:SortedSet"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="Map"> <xsd:sequence> <xsd:element name="Keys" type="tns:ArrayOfAnyType"/> <xsd:element name="Values" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> </pre> |

```

<xsd:complexType name="Hashtable">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="HashMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SortedMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:Map"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TreeMap">
  <xsd:complexContent mixed="false">
    <xsd:extension base="tns:SortedMap"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Iterator">
  <xsd:sequence>
    <xsd:element name="Data" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Enumeration">
  <xsd:sequence>
    <xsd:element name="Data" type="tns:ArrayOfAnyType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfAnyType">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
<xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wSDL/default/"
  xmlns:xns5="http://systinet.com/containers/literal/ms.net">
  <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/>
  <xsd:import namespace="http://systinet.com/wSDL/default"/>
  <xsd:element name="AXOID" type="xsd:string" nillable="true"/>
  <xsd:element name="objectVersion" type="xsd:string" nillable="true"/>
  <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/>
  <xsd:element name="actionLog" type="xns4:ActionLog" nillable="true"/>
  <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/>
  <xsd:element name="boolean_Response" type="xsd:boolean"/>
  <xsd:element name="ActionLog_Response" type="xns4:ActionLog" nillable="true"/>
</xsd:schema>
</wsdl:types>
<wsdl:message name="ContentConsumption_clearActionLogs_1_Request"/>
<wsdl:message name="ContentConsumption_retrieveActionLogs_1_Request"/>
<wsdl:message name="ContentConsumption_getLastActionLog_Response">
  <wsdl:part name="response" element="ns0:ActionLog_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_getLastActionLog_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_deleteCacheContent_1_Request"/>
<wsdl:message name="ContentConsumption_retrieveActionLogs_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="ContentConsumption_insertActionLog_Response"/>
<wsdl:message name="ContentConsumption_insertActionLog_1_Request"/>

```

[illegible]

| | |
|--|---|
| | <pre> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="clearActionLogs"> <map:java-operation name="clearActionLogs" signature="KClA"/> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#clearActionLogs#KClA" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="getLastActionLog"> <map:java-operation name="getLastActionLog" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyIMQWN0aW9uTG9nOw==" /> <soap:operation soapAction="urn:ContentConsumptionContentConsumption#getLastActionLog#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyIMQWN0aW9uTG9nOw==" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="ContentConsumption"> <wsdl:port name="ContentConsumption" binding="tns:ContentConsumption"> <soap:address location="http://efpc037:6060/ContentConsumption"/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|--|---|

| | |
|-------------------------|---|
| Method | insertActionLog |
| Description | Stores the given action log associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsdl/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXOID i:type="d:string">AXOID</wn0:AXOID> <wn0:objectVersion i:type="d:string">32</wn0:objectVersion> <wn0:protectionStamp i:type="d:string">protection</wn0:protectionStamp> <wn0:actionLog i:type="wn1:ActionLog"/> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:string AXOID: Identifier of the AXMEDIS object whose protection information is going to be stored</p> <p>xsd:string version: Version of the AXMEDIS object whose protection information is going to be stored</p> <p>xsd:string protectionStamp: Identifier of the protection information</p> |

| | |
|-------------------------|--|
| | xsd:actionLog: The action log to be stored. This information is already defined in the AXMEDIS Supervisor section |
| Output parameters | None |
| Method | retrieveActionLogs |
| Description | This method retrieves the action logs stored in the local cache info. |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:ArrayList_Response i:nil="true"/> </e:Body> </e:Envelope></pre> |
| Input parameters | None |
| Output parameters | xsd:arrayList: A list of all the action logs stored for this user. |
| Method | DeleteCacheContent |
| Description | This method retrieves the action logs stored in the local cache info. |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:ArrayList_Response i:nil="true"/> </e:Body> </e:Envelope></pre> |
| Input parameters | None |
| Output parameters | xsd:arrayList: A list of all the action logs stored for this user. |
| Method | clearActionLogs |
| Description | Deletes action logs from the cache, after positive authorisation of the user in the connected environment |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <e:Body/> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body></pre> |

| | |
|-------------------------|--|
| | <code></e:Envelope></code> |
| Input parameters | None |
| Output parameters | Xsd:boolean, indicating that the operation was correct. |
| Method | getLastActionLog |
| Description | Returns the last action log |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXOID i:type="d:string">AXOID</wn0:AXOID> <wn0:objectVersion i:type="d:string">32</wn0:objectVersion> <wn0:protectionStamp i:type="d:string">protection</wn0:protectionStamp> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXOID i:type="d:string">AXOID</wn0:AXOID> <wn0:objectVersion i:type="d:string">32</wn0:objectVersion> <wn0:protectionStamp i:type="d:string">protection</wn0:protectionStamp> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:string AXOID: Identifier of the AXMEDIS object whose protection information is going to be retrieved xsd:string version: Version of the AXMEDIS object whose protection information is going to be retrieved xsd:string protectionStamp: Identifier of the protection information |
| Output parameters | xsd:actionLog: The action log to be returned |

4.13 License Manager (FUPF)

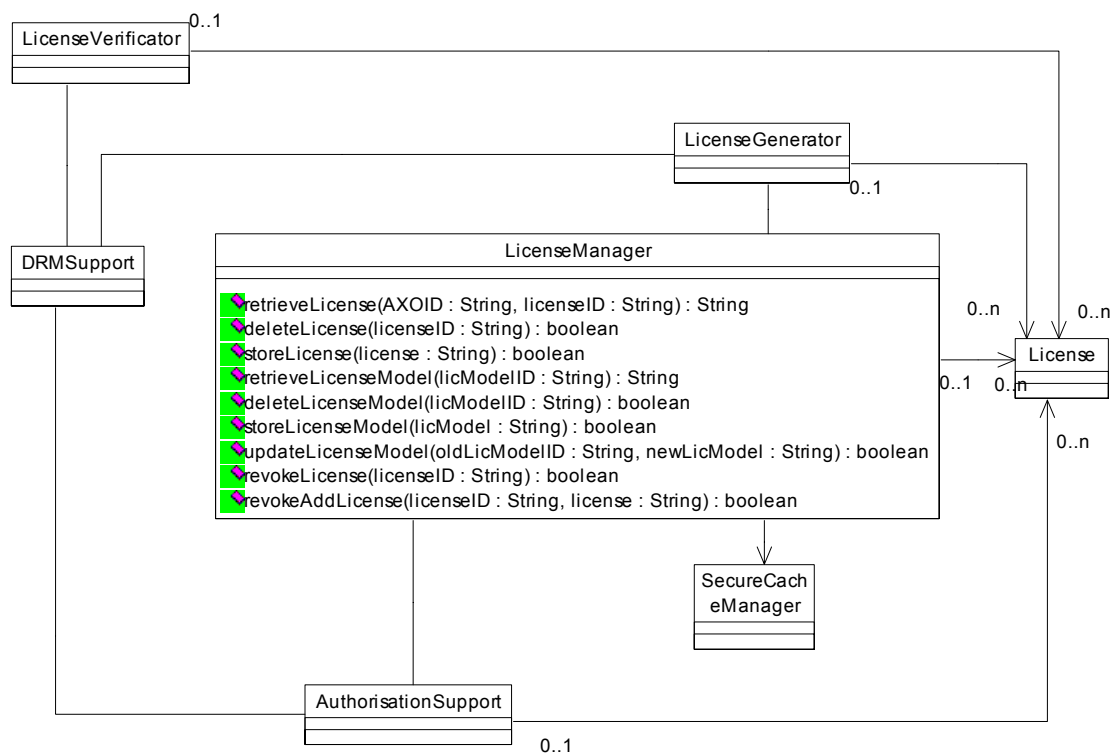
| Module Profile | | |
|--------------------------------|---------------------------------|---|
| License Manager | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| DRM Support | | |
| License Manager | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a |

| | | section |
|----------------|-----------------------------------|--|
| XML | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

This module provides license managing and caching functionalities.

4.13.1 Architecture of License Manager module

Next figure shows the UML diagram of this module.



License Manager class diagram

| LicenseManager | |
|-----------------|--|
| Methods | Description |
| retrieveLicense | This function retrieves the licenses stored in the license database. It retrieves the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| deleteLicense | This function deletes the licenses stored in the license database. It deletes the |

| | |
|----------------------|---|
| | license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| storeLicense | This function stores a license in the license database. |
| retrieveLicenseModel | This function returns a licenseModel stored in the License DB |
| deleteLicenseModel | This function deletes a licenseModel from the License DB |
| storeLicenseModel | This function inserts a license model in the License DB |
| updateLicenseModel | This function updates the licenseModel given a licenseModelID and a new description of licenseModel |
| revokeLicense | This function changes the status of a license to revoked in the License DB |
| revokeAddLicense | This function changes the status of a license to revoked and adds the license that substitutes it in the License DB |

4.13.2 WSDL Interface

| License Manager | |
|-----------------|---|
| WSDL | <pre> <?xml version='1.0'?> <wsdl:definitions name='LicenseManager' targetNamespace='urn:LicenseManager' xmlns:map='http://systinet.com/mapping/' xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/' xmlns:ns0='http://systinet.com/xsd/SchemaTypes/' xmlns:tns='urn:LicenseManager' xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'> <wsdl:types> <xsd:schema targetNamespace='http://systinet.com/xsd/SchemaTypes/' elementFormDefault='qualified' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:tns='http://systinet.com/xsd/SchemaTypes/'> <xsd:element name='AXOID' type='xsd:string' nillable='true'> <xsd:element name='licenseID' type='xsd:string' nillable='true'> <xsd:element name='string_Response' type='xsd:string' nillable='true'> <xsd:element name='license' type='xsd:string' nillable='true'> <xsd:element name='boolean_Response' type='xsd:boolean'> <xsd:element name='licenseModel' type='xsd:string' nillable='true'> <xsd:element name='licModelID' type='xsd:string' nillable='true'> <xsd:element name='licModelNew' type='xsd:string' nillable='true'> <xsd:element name='newLicense' type='xsd:string' nillable='true'> </xsd:schema> </wsdl:types> <wsdl:message name='LicenseManager_storeLicenseModel_Response'> <wsdl:part name='response' element='ns0:boolean_Response'> </wsdl:message> <wsdl:message name='LicenseManager_revokeLicense_1_Request'> <wsdl:part name='licenseID' element='ns0:licenseID'> </wsdl:message> <wsdl:message name='LicenseManager_deleteLicenseModel_1_Request'> <wsdl:part name='licModelID' element='ns0:licModelID'> </wsdl:message> <wsdl:message name='LicenseManager_retrieveLicenseModel_Response'> <wsdl:part name='response' element='ns0:string_Response'> </wsdl:message> <wsdl:message name='LicenseManager_retrieveLicense_Response'> <wsdl:part name='response' element='ns0:string_Response'> </wsdl:message> <wsdl:message name='LicenseManager_storeLicense_Response'> <wsdl:part name='response' element='ns0:boolean_Response'> </wsdl:message> <wsdl:message name='LicenseManager_updateLicenseModel_1_Request'> <wsdl:part name='licModelID' element='ns0:licModelID'> <wsdl:part name='licModelNew' element='ns0:licModelNew'> </wsdl:message> <wsdl:message name='LicenseManager_revokeAddLicense_Response'> <wsdl:part name='response' element='ns0:boolean_Response'> </pre> |


```

</wsdl:message>
<wsdl:message name='LicenseManager_deleteLicense_1_Request'>
  <wsdl:part name='AXOID' element='ns0:AXOID'/>
  <wsdl:part name='licenseID' element='ns0:licenseID'/>
</wsdl:message>
<wsdl:message name='LicenseManager_deleteLicenseModel_Response'>
  <wsdl:part name='response' element='ns0:boolean_Response'/>
</wsdl:message>
<wsdl:message name='LicenseManager_revokeAddLicense_1_Request'>
  <wsdl:part name='licenseID' element='ns0:licenseID'/>
  <wsdl:part name='newLicense' element='ns0:newLicense'/>
</wsdl:message>
<wsdl:message name='LicenseManager_storeLicenseModel_1_Request'>
  <wsdl:part name='licenseModel' element='ns0:licenseModel'/>
</wsdl:message>
<wsdl:message name='LicenseManager_revokeLicense_Response'>
  <wsdl:part name='response' element='ns0:boolean_Response'/>
</wsdl:message>
<wsdl:message name='LicenseManager_retrieveLicenseModel_1_Request'>
  <wsdl:part name='licModelID' element='ns0:licModelID'/>
</wsdl:message>
<wsdl:message name='LicenseManager_updateLicenseModel_Response'>
  <wsdl:part name='response' element='ns0:boolean_Response'/>
</wsdl:message>
<wsdl:message name='LicenseManager_deleteLicense_Response'>
  <wsdl:part name='response' element='ns0:boolean_Response'/>
</wsdl:message>
<wsdl:message name='LicenseManager_retrieveLicense_1_Request'>
  <wsdl:part name='AXOID' element='ns0:AXOID'/>
  <wsdl:part name='licenseID' element='ns0:licenseID'/>
</wsdl:message>
<wsdl:message name='LicenseManager_storeLicense_1_Request'>
  <wsdl:part name='license' element='ns0:license'/>
</wsdl:message>
<wsdl:portType name='LicenseManager'>
  <wsdl:operation name='retrieveLicense' parameterOrder='AXOID licenseID'>
    <wsdl:input message='tns:LicenseManager_retrieveLicense_1_Request'/>
    <wsdl:output message='tns:LicenseManager_retrieveLicense_Response'/>
  </wsdl:operation>
  <wsdl:operation name='storeLicense' parameterOrder='license'>
    <wsdl:input message='tns:LicenseManager_storeLicense_1_Request'/>
    <wsdl:output message='tns:LicenseManager_storeLicense_Response'/>
  </wsdl:operation>
  <wsdl:operation name='storeLicenseModel' parameterOrder='licenseModel'>
    <wsdl:input message='tns:LicenseManager_storeLicenseModel_1_Request'/>
    <wsdl:output message='tns:LicenseManager_storeLicenseModel_Response'/>
  </wsdl:operation>
  <wsdl:operation name='deleteLicense' parameterOrder='AXOID licenseID'>
    <wsdl:input message='tns:LicenseManager_deleteLicense_1_Request'/>
    <wsdl:output message='tns:LicenseManager_deleteLicense_Response'/>
  </wsdl:operation>
  <wsdl:operation name='retrieveLicenseModel' parameterOrder='licModelID'>
    <wsdl:input message='tns:LicenseManager_retrieveLicenseModel_1_Request'/>
    <wsdl:output message='tns:LicenseManager_retrieveLicenseModel_Response'/>
  </wsdl:operation>
  <wsdl:operation name='updateLicenseModel' parameterOrder='licModelID licModelNew'>
    <wsdl:input message='tns:LicenseManager_updateLicenseModel_1_Request'/>
    <wsdl:output message='tns:LicenseManager_updateLicenseModel_Response'/>
  </wsdl:operation>
  <wsdl:operation name='deleteLicenseModel' parameterOrder='licModelID'>
    <wsdl:input message='tns:LicenseManager_deleteLicenseModel_1_Request'/>
    <wsdl:output message='tns:LicenseManager_deleteLicenseModel_Response'/>
  </wsdl:operation>
  <wsdl:operation name='revokeLicense' parameterOrder='licenseID'>
    <wsdl:input message='tns:LicenseManager_revokeLicense_1_Request'/>
    <wsdl:output message='tns:LicenseManager_revokeLicense_Response'/>
  </wsdl:operation>
  <wsdl:operation name='revokeAddLicense' parameterOrder='licenseID newLicense'>
    <wsdl:input message='tns:LicenseManager_revokeAddLicense_1_Request'/>
    <wsdl:output message='tns:LicenseManager_revokeAddLicense_Response'/>
  </wsdl:operation>

```

| | |
|--|---|
| | <pre> </wsdl:operation> </wsdl:portType> <wsdl:binding name='LicenseManager' type='tns:LicenseManager'> <soap:binding transport='http://schemas.xmlsoap.org/soap/http' style='document'/> <wsdl:operation name='retrieveLicense'> <map:java-operation name='retrieveLicense' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZ zs='/> <soap:operation soapAction='urn:LicenseManagerLicenseManager#retrieveLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZ hL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='storeLicense'> <map:java-operation name='storeLicense' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'> <soap:operation soapAction='urn:LicenseManagerLicenseManager#storeLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='storeLicenseModel'> <map:java-operation name='storeLicenseModel' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'> <soap:operation soapAction='urn:LicenseManagerLicenseManager#storeLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='deleteLicense'> <map:java-operation name='deleteLicense' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla'> <soap:operation soapAction='urn:LicenseManagerLicenseManager#deleteLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZh L2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='retrieveLicenseModel'> <map:java-operation name='retrieveLicenseModel' signature='KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs='/> <soap:operation soapAction='urn:LicenseManagerLicenseManager#retrieveLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nOyl MamF2YS9sYW5nL1N0cmLuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='updateLicenseModel'> <map:java-operation name='updateLicenseModel' </pre> |
|--|---|

| | |
|------------------------|---|
| | <pre> signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#updateLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteLicenseModel"> <map:java-operation name="deleteLicenseModel" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#deleteLicenseModel#KExqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="revokeLicense"> <map:java-operation name="revokeLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#revokeLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="revokeAddLicense"> <map:java-operation name="revokeAddLicense" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:LicenseManagerLicenseManager#revokeAddLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqY XZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="LicenseManager"> <wsdl:port name="LicenseManager" binding="tns:LicenseManager"> <soap:address location="http://efpc037:6060/LicenseManager"/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
| Method | retrieveLicense |
| Description | this function retrieve the licenses stored in the license database. It retrieves the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| Request Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:AXOID i:type="d:string">AXOID1</wn0:AXOID> <wn0:licenseID i:type="d:string">AXLID1</wn0:licenseID> </e:Body> </e:Envelope> </pre> |

| | |
|-------------------------|---|
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:string_Response i:type="d:string">&lt;it;license/&gt;</wn0:string_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | xsd:string: AXOID: AXMEDIS object identifier xsd:string: licenseID: license identifier |
| Output parameters | xsd:string: returns the requested license |
| Method | deleteLicense |
| Description | this function deletes the licenses stored in the license database. It deletes the license with the licenseID set as a parameter, or the licenses associated to an AXOID. |
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:license i:type="d:string">&lt;it;license/&gt;</wn0:license> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | xsd:string: licenseID: license identifier |
| Output parameters | xsd:Boolean: true if the license has been deleted, false if not. |
| Method | storeLicense |
| Description | this function stores a license in the license database. |
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:license i:type="d:string">&lt;it;license/&gt;</wn0:license> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | xsd:string: license: license to be stored |
| Output parameters | xsd:boolean: true if the license has been stored, false if not. |
| Method | retrieveLicenseModel |
| Description | This function returns a licenseModel stored in the License DB |
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/></pre> |

| | |
|-------------------------|--|
| | <pre> <e:Body> <wn0:licModelID i:type="d:string">licModel001</wn0:licModelID> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:string_Response i:type="d:string">&lt;it;licenseModel/&gt;</wn0:string_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:string licModelID: This parameter contains the identifier of the license Model that has to be retrieved |
| Output parameters | xsd:string: This function returns a String with the whole XML document representing the License Model |
| Method | storeLicenseModel |
| Description | This function inserts a license model in the License DB |
| Request Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:licenseModel i:type="d:string">&lt;it;licenseModel/&gt;</wn0:licenseModel> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:string license: It contains the whole XML document representing the license model to be stored in the database |
| Output parameters | xsd:boolean: This function returns true if the insert has been finished successfully |
| Method | deleteLicenseModel |
| Description | This function deletes a licenseModel from the License DB |
| Request Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:licModelID i:type="d:string">licModel001</wn0:licModelID> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:string licModelID: It contains the identifier of the licenseModel to be deleted |
| Output parameters | xsd:boolean: This function returns true if the delete has been finished successfully |
| Method | updateLicenseModel |
| Description | This function updates the licenseModel given a licenseModelID and a new description of licenseModel |

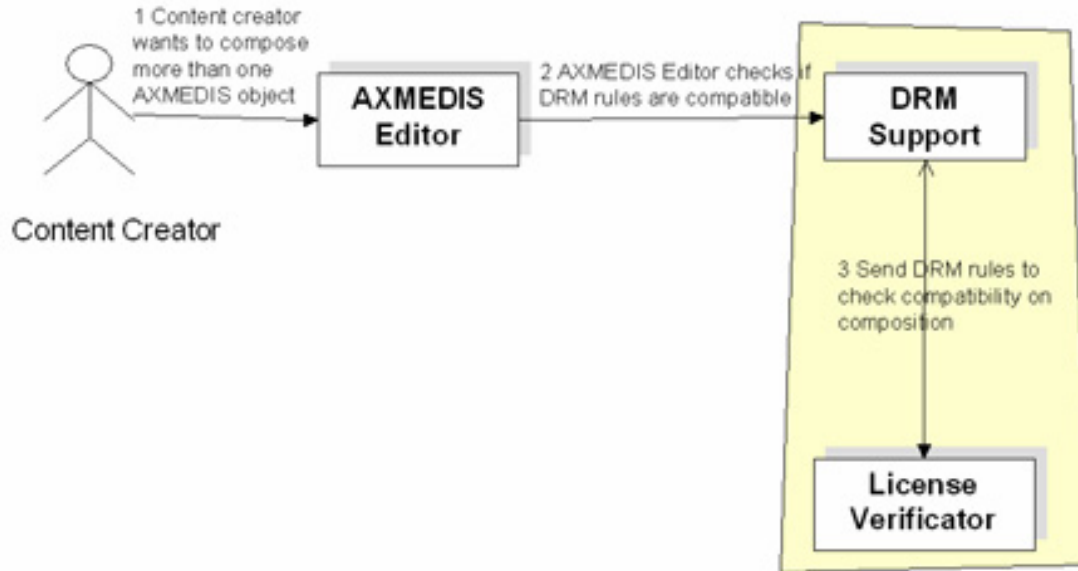
| | |
|-------------------------|--|
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:licModelID i:type="d:string">licModel001</wn0:licModelID> <wn0:licModelNew i:type="d:string">&lt;licenseModel/&gt;</wn0:licModelNew> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | <p>xsd:string licModelID: It contains the identifier of the licenseModel to be updated</p> <p>xsd:string licModelNew: It contains the new licenseModel</p> |
| Output parameters | xsd:boolean: This function returns true if the update has been finished successfully |
| Method | revokeLicense |
| Description | This function changes the status of a license to revoked in the License DB |
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:licenseID i:type="d:string">lic001</wn0:licenseID> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | xsd:string PARID: It contains the identifier of the license that has been revoked |
| Output parameters | xsd:boolean: This function returns true if the license status has been changed to revoked successfully |
| Method | revokeAddLicense |
| Description | This function changes the status of a license to revoked and adds the license that substitutes it in the License DB |
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:licenseID i:type="d:string">lic001</wn0:licenseID> <wn0:newLicense i:type="d:string">&lt;license/&gt;</wn0:newLicense> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | <p>xsd:string PARID: It contains the identifier of the license that has been revoked</p> <p>xsd:string newLicense: license that substitutes the revoked license</p> |

| | |
|-------------------|--|
| Output parameters | xsd:boolean: This function returns true if the license status has been changed to revoked successfully |
|-------------------|--|

4.14 License Vericator (FUPF)

4.14.1 License Vericator scenario

The scenario how an AXMEDIS tool (for instance, AXMEDIS editor), interacts with License Vericator.



License verification

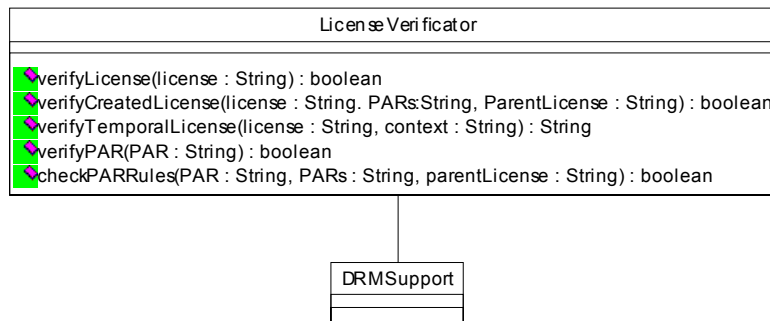
| Module Profile | | |
|--------------------------------|---------------------------------|---|
| License Verificator | | |
| Executable or Library(Support) | Support Library | |
| Single Thread or Multithread | Single Thread | |
| Language of Development | C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| DRM Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, | Library used for the development, |

| | | |
|----------------|---------------------------------|--|
| | etc. | platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

This module provides license syntactic verification functionalities.

4.14.2 Architecture of License Vericator module

Next figure shows the UML diagram of this module.



License vericator class diagram

| License vericator | |
|-----------------------|--|
| Methods | Description |
| verifyLicense | Verifies a license syntactically against the schemas defined within the license. |
| verifyCreatedLicense | Verifies if the license can be generated according to the PARs and the parent licenses (e.g. Distributor or Creator licenses). |
| verifyTemporalLicense | Verifies that the license generated by the user fulfils the initial desirables requirements of the user. For example, the user can verify that with this license he could exercise the desired action over the AXObject. |
| verifyPAR | Verifies a PAR syntactically against the schemas defined within the PAR. |
| checkPARRules | Verifies if the PAR can be generated according to the parent licenses and PARs. |

4.14.3 WSDL Interface

| License Vericator | |
|-------------------|--|
| WSDL | <pre> <?xml version='1.0'?> <wsdl:definitions name='LicenseVericator' targetNamespace='urn:LicenseVericator' xmlns:map='http://systinet.com/mapping/' xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/' xmlns:ns0='http://systinet.com/xsd/SchemaTypes/' xmlns:tns='urn:LicenseVericator' xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'> <wsdl:types> <xsd:schema targetNamespace='http://systinet.com/xsd/SchemaTypes/' elementFormDefault='qualified' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:tns='http://systinet.com/xsd/SchemaTypes/'> <xsd:element name='license' type='xsd:string' nillable='true'/> </xsd:schema> </wsdl:types> </pre> |


```

<xsd:element name="boolean_Response" type="xsd:boolean"/>
<xsd:element name="PARs" type="xsd:string" nillable="true"/>
<xsd:element name="parentLicense" type="xsd:string"
  nillable="true"/>
<xsd:element name="context" type="xsd:string" nillable="true"/>
<xsd:element name="string_Response" type="xsd:string"
  nillable="true"/>
<xsd:element name="PAR" type="xsd:string" nillable="true"/>
</xsd:schema>

</wsdl:types>
<wsdl:message name="LicenseVerifier_verifyCreatedLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_checkPARRules_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_checkPARRules_1_Request">
  <wsdl:part name="PAR" element="ns0:PAR"/>
  <wsdl:part name="PARs" element="ns0:PARs"/>
  <wsdl:part name="parentLicense" element="ns0:parentLicense"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_verifyCreatedLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_verifyPAR_1_Request">
  <wsdl:part name="PAR" element="ns0:PAR"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_verifyTemporalLicense_Response">
  <wsdl:part name="response" element="ns0:string_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_verifyLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_verifyPAR_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_verifyTemporalLicense_1_Request">
  <wsdl:part name="license" element="ns0:license"/>
  <wsdl:part name="context" element="ns0:context"/>
</wsdl:message>
<wsdl:message name="LicenseVerifier_verifyLicense_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:portType name="LicenseVerifier">
  <wsdl:operation name="verifyLicense" parameterOrder="license">
    <wsdl:input message="tns:LicenseVerifier_verifyLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyCreatedLicense" parameterOrder="license PARs parentLicense">
    <wsdl:input message="tns:LicenseVerifier_verifyCreatedLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyCreatedLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyTemporalLicense" parameterOrder="license context">
    <wsdl:input message="tns:LicenseVerifier_verifyTemporalLicense_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyTemporalLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="verifyPAR" parameterOrder="PAR">
    <wsdl:input message="tns:LicenseVerifier_verifyPAR_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_verifyPAR_Response"/>
  </wsdl:operation>
  <wsdl:operation name="checkPARRules" parameterOrder="PAR PARs parentLicense">
    <wsdl:input message="tns:LicenseVerifier_checkPARRules_1_Request"/>
    <wsdl:output message="tns:LicenseVerifier_checkPARRules_Response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="LicenseVerifier" type="tns:LicenseVerifier">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>

```

| | |
|--|---|
| | <pre> <wsdl:operation name='verifyLicense'> <map:java-operation name='verifyLicense' signature='KExqYXZhL2xhbmvcU3RyaW5nOyla'/> <soap:operation soapAction='urn:LicenseVericatorLicenseVericator#verifyLicense#KExqYXZhL2xhbmvcU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='verifyCreatedLicense'> <map:java-operation name='verifyCreatedLicense' signature='KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO Oyla'/> <soap:operation soapAction='urn:LicenseVericatorLicenseVericator#verifyCreatedLicense#KExqYXZhL2xhbmvcU3RyaW5nO 0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='verifyTemporalLicense'> <map:java-operation name='verifyTemporalLicense' signature='KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZ zs='/> <soap:operation soapAction='urn:LicenseVericatorLicenseVericator#verifyTemporalLicense#KExqYXZhL2xhbmvcU3RyaW5n O0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='verifyPAR'> <map:java-operation name='verifyPAR' signature='KExqYXZhL2xhbmvcU3RyaW5nOyla'/> <soap:operation soapAction='urn:LicenseVericatorLicenseVericator#verifyPAR#KExqYXZhL2xhbmvcU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='checkPARRules'> <map:java-operation name='checkPARRules' signature='KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5n Oyla'/> <soap:operation soapAction='urn:LicenseVericatorLicenseVericator#checkPARRules#KExqYXZhL2xhbmvcU3RyaW5nO0xq YXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name='LicenseVericator'> <wsdl:port name='LicenseVericator' binding='tns:LicenseVericator'> <soap:address location='http://efpc037:6060/LicenseVericator'/> </wsdl:port> </wsdl:service> </pre> |
|--|---|

| | |
|-------------------------|--|
| | <pre> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
| Method | verifyLicense |
| Description | Verifies a license syntactically against the schemas defined within the license. |
| Request Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:license i:type="d:string">&lt;it;license/&gt;</wn0:license> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:string license: the license to be verified |
| Output parameters | xsd:boolean: true if the license is correct, false if not. |
| Method | verifyCreatedLicense |
| Description | Verifies if the license can be generated according to the PARs and the parent licenses (e.g. Distributor or Creator licenses). |
| Request Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:license i:type="d:string">&lt;it;license/&gt;</wn0:license> <wn0:PARs i:type="d:string">&lt;it;PotentialAvailableRights/&gt;</wn0:PARs> <wn0:parentLicense i:type="d:string">&lt;it;license/&gt;</wn0:parentLicense> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:string license: the license to be verified</p> <p>xsd:string PARs: PARs of the corresponding AXMEDIS object</p> <p>xsd:string parentLicense: distributor license</p> |
| Output parameters | xsd:boolean: true if the license can be generated, false if not. |
| Method | verifyTemporalLicense |
| Description | Verifies that the license generated by the user fulfils the initial desirables requirements of the user. For example, the user can verify that with this license he could exercise the desired action over the AXObject. |
| Request Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:license i:type="d:string">&lt;it;license/&gt;</wn0:license> </e:Body> </pre> |

| | |
|-------------------------|---|
| | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:string i:type="d:string">&lt;it;authorisationContext/&gt;</wn0:string> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:string_Response i:type="d:string">Yes: If 4 Euros are paid.</wn0:string_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:string license: the license to be verified</p> <p>xsd:string context: user conditions</p> |
| Output parameters | xsd: string: Additional conditions that the user must fulfill. |
| Method | verifyPAR |
| Description | Verifies a PAR syntactically against the schemas defined within the PAR. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:PAR i:type="d:string">&lt;it;license/&gt;</wn0:PAR> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:string PAR: the PAR to be verified |
| Output parameters | xsd:boolean: true if the PAR is correct, false if not. |
| Method | checkPARRules |
| Description | Verifies if the PAR can be generated according to the parent licenses and PARs. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:PAR i:type="d:string">PAR</wn0:PAR> <wn0:PARs i:type="d:string">PARParent</wn0:PARs> <wn0:parentLicense i:type="d:string">&lt;it;license/&gt;</wn0:parentLicense> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:string PAR: the PAR to be verified</p> <p>xsd:string PARs: PARs of the parent AXMEDIS object</p> <p>xsd:string parentLicense: distributor license</p> |

| | |
|-------------------|---|
| Output parameters | xsd:boolean: true if the PARs can be generated, false if not. |
|-------------------|---|

4.15 Protection Info Manager(FUPF)

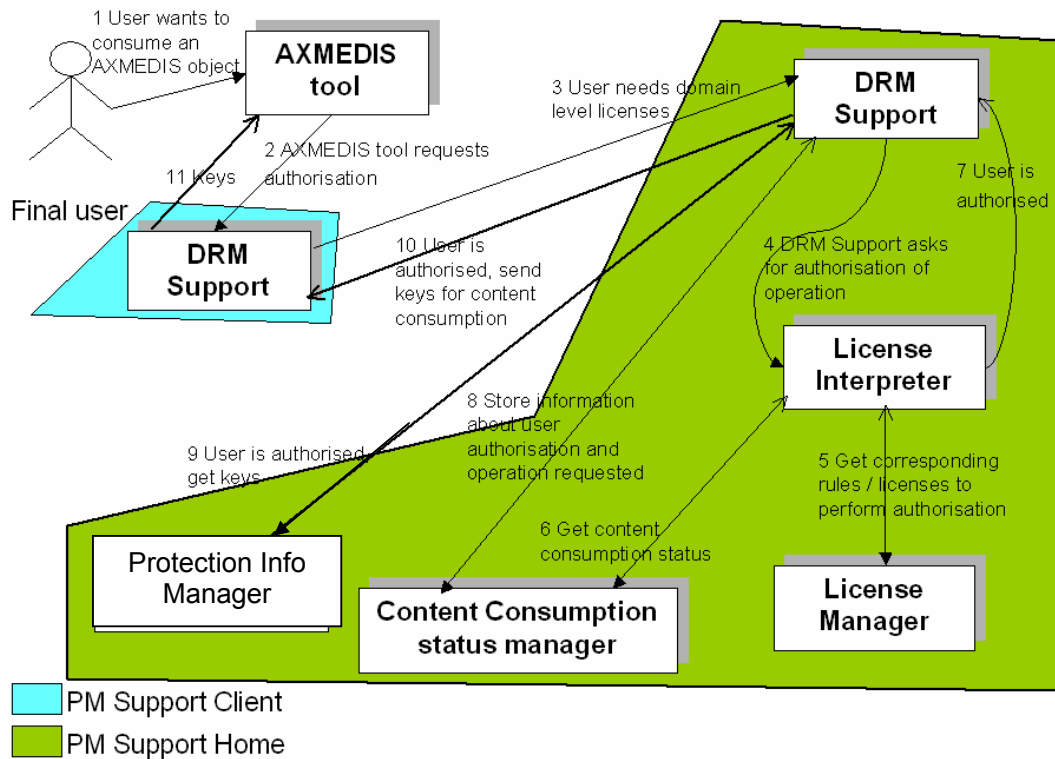
The Protection Info Manager has the responsibility of dealing with protection information when it is transmitted to the final users. It also provides the entry point to the key generation functionality by means of the Key Generator module.

DRM Support makes use of it, in order to access to the protection information stored in the local cache info and for generating keys for protecting AXMEDIS objects.

In the rest of the section it is described in more detail the functionality provided by this module.

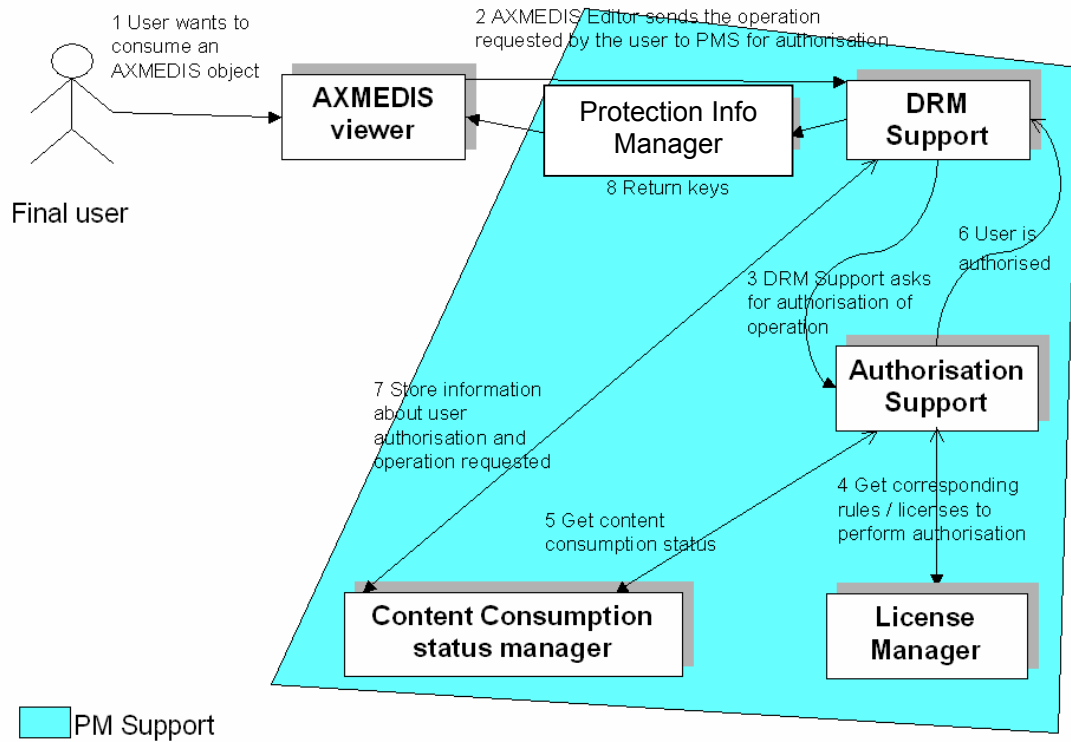
4.15.1 Protection Info Manager Scenarios

The scenario describes the use of protection info manager in the PMS Domain Home.



Use of protection info manager PMS Domain Home

The scenario in the next figure describes the use of protection info manager in the PMS Client.



Use of protection info manager PMS Client

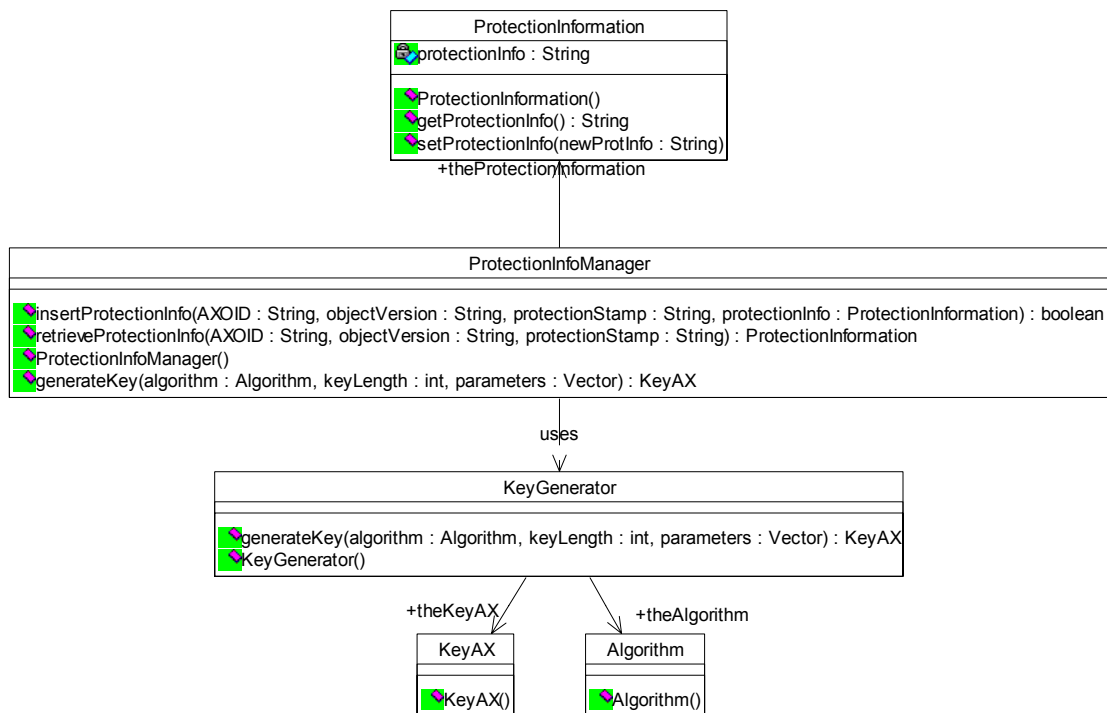
| Module Profile | | |
|--------------------------------|-----------------------------------|---|
| Protection info Manager | | |
| Executable or Library(Support) | Library (Support) | |
| Single Thread or Multithread | Single Thread | |
| Language of Development | C / C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | PC (Linux / Windows) | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| Any | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |

| | | |
|----------------|---------------------------------|--|
| | | |
| Used Libraries | Name of the library and version | License status: GPL, LGPL, PEK, proprietary, authorized or not |
| Openssl | | |
| | | |
| | | |

This module provides key managing and caching functionalities.

4.15.2 Protection Info Manager module architecture

Next figure shows the UML diagram describing the Protection Info Manager module. Nevertheless, this module is based on OpenSSL, so it is not object oriented in nature, but C++ interface could be provided.



Protection Info manager class diagram

ProtectionInfoManager:

This module provides the functionality regarding protection information. It also allows the generation of keys, by using the Key Generator class.

| ProtectionInfoManager | |
|-----------------------------|---|
| Method | Description |
| insertProtectionInformation | This method stores the given protection information associated to an AXMEDIS object identifier, the object version and the protection stamp. Protection Information Manager will not physically store this information, but it will call to the Light / Secure Cache Manager module that will store it into the Local Cache Info. |

| | |
|-------------------------------|--|
| retrieveProtectionInformation | This method retrieves the requested protection information. The information needed to retrieve the protection information is the AXMEDIS object identifier, the object version and the protection stamp. This information will be requested to the Light / Secure Cache Manager module, which is in charge of retrieving it in the Local Cache Info. |
| generateKey | This method permits the creation of a key for protecting an AXMEDIS object. It needs several parameters like key length, algorithm used and other algorithm dependent. |

| ProtectionInfoManager | |
|-----------------------|---|
| WSDL | <pre> <?xml version="1.0" encoding="UTF-8"?> <wsdl:definitions name="ProtectionInfoManager" targetNamespace="urn:ProtectionInfoManager" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:tns="urn:ProtectionInfoManager" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="Vector"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="ArrayOfAnyType"> <xsd:sequence> <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/" xmlns:xns5="http://systinet.com/containers/literal/ms.net"> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:import namespace="http://systinet.com/wsdl/default"/> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="objectVersion" type="xsd:string" nillable="true"/> <xsd:element name="protectionStamp" type="xsd:string" nillable="true"/> <xsd:element name="protectionInfo" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="ProtectionInformation_Response" type="xns4:ProtectionInformation" nillable="true"/> <xsd:element name="algorithm" type="xns4:Algorithm" nillable="true"/> <xsd:element name="keyLength" type="xsd:int"/> <xsd:element name="parameters" type="xns5:Vector" nillable="true"/> <xsd:element name="KeyAX_Response" type="xns4:KeyAX" nillable="true"/> </xsd:schema> </pre> |


```

<xsd:schema targetNamespace="http://systinet.com/wsdl/default/"
  elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://systinet.com/wsdl/default/"
  xmlns:map="http://systinet.com/mapping/">
  <xsd:complexType name="ProtectionInformation">
    <xsd:sequence>
      <xsd:element name="protectionInfo" type="xsd:string"
        nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Algorithm">
    <xsd:sequence>
      <xsd:element name="algorithm" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="KeyAX">
    <xsd:sequence>
      <xsd:element name="key" type="xsd:base64Binary"
        nillable="true"/>
      <xsd:element name="keyByte" type="xsd:base64Binary"
        nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
</wsdl:types>

<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
  <wsdl:part name="protectionInfo" element="ns0:protectionInfo"/>
</wsdl:message>

<wsdl:message name="ProtectionInfoManager_insertProtectionInfo_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>

<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_1_Request">
  <wsdl:part name="AXOID" element="ns0:AXOID"/>
  <wsdl:part name="objectVersion" element="ns0:objectVersion"/>
  <wsdl:part name="protectionStamp" element="ns0:protectionStamp"/>
</wsdl:message>

<wsdl:message name="ProtectionInfoManager_retrieveProtectionInfo_Response">
  <wsdl:part name="response"
    element="ns0:ProtectionInformation_Response"/>
</wsdl:message>

<wsdl:message name="ProtectionInfoManager_generateKey_1_Request">
  <wsdl:part name="algorithm" element="ns0:algorithm"/>
  <wsdl:part name="keyLength" element="ns0:keyLength"/>
  <wsdl:part name="parameters" element="ns0:parameters"/>
</wsdl:message>

<wsdl:message name="ProtectionInfoManager_generateKey_Response">
  <wsdl:part name="response" element="ns0:KeyAX_Response"/>
</wsdl:message>

<wsdl:portType name="ProtectionInfoManager">
  <wsdl:operation name="insertProtectionInfo"
    parameterOrder="AXOID objectVersion protectionStamp protectionInfo">
    <wsdl:input
      message="tns:ProtectionInfoManager_insertProtectionInfo_1_Request"/>
    <wsdl:output
      message="tns:ProtectionInfoManager_insertProtectionInfo_Response"/>
  </wsdl:operation>

  <wsdl:operation name="retrieveProtectionInfo"
    parameterOrder="AXOID objectVersion protectionStamp">

```

| | |
|--|---|
| | <pre> <wsdl:input message="tns:ProtectionInfoManager_retrieveProtectionInfo_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_retrieveProtectionInfo_Response"/> </wsdl:operation> <wsdl:operation name="generateKey" parameterOrder="algorithm keyLength parameters"> <wsdl:input message="tns:ProtectionInfoManager_generateKey_1_Request"/> <wsdl:output message="tns:ProtectionInfoManager_generateKey_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="ProtectionInfoManager" type="tns:ProtectionInfoManager"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/> <wsdl:operation name="insertProtectionInfo"> <map:java-operation name="insertProtectionInfo" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWN0aW9uSW5mb3JtYXRpb247KVo="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#insertProtectionInfo#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWN0aW9uSW5mb3JtYXRpb247KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="retrieveProtectionInfo"> <map:java-operation name="retrieveProtectionInfo" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWN0aW9uSW5mb3JtYXRpb247KVo="/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#retrieveProtectionInfo#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xQcm90ZWN0aW9uSW5mb3JtYXRpb247KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateKey"> <map:java-operation name="generateKey" signature="KExBbGdvcmI0aG07SUxqYXZhL3V0aWwVjG9yOylMS2V5QVg7"/> <soap:operation soapAction="urn:ProtectionInfoManagerProtectionInfoManager#generateKey#KExBbGdvcmI0aG07SUxqYXZhL3V0aWwVjG9yOylMS2V5QVg7" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="ProtectionInfoManager"> <wsdl:port name="ProtectionInfoManager" binding="tns:ProtectionInfoManager"> <soap:address </pre> |
|--|---|

| | |
|--|--|
| | <pre> location="http://efpc037:6060/ProtectionInfoManager/"> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
|--|--|

| | |
|-------------------------|--|
| Method | insertProtectionInfo |
| Description | Stores the given protection information associated to an AXMEDIS object identifier, the object version and the protection stamp. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsdl/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXOID i:type="d:string">AXOID</wn0:AXOID> <wn0:objectVersion i:type="d:string">objectVersion</wn0:objectVersion> <wn0:protectionStamp i:type="d:string">protectionStamp</wn0:protectionStamp> <wn0:protectionInfo i:type="wn1:ProtectionInformation"> <wn1:protectionInfo i:nil="true"/> </wn0:protectionInfo> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsdl/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:string AXOID: Identifier of the AXMEDIS object whose protection information is going to be stored</p> <p>xsd:string version: Version of the AXMEDIS object whose protection information is going to be stored</p> <p>xsd:string protectionStamp: Identifier of the protection information</p> <p>xsd:protectionInfo: The protection information to be stored. It is represented with a class, but maybe only a string will be needed</p> |
| Output parameters | xsd:boolean : The function returns true if the operation was successfully done, false otherwise |
| Method | retrieveProtectionInfo |
| Description | This method retrieves the requested protection information. The information needed to retrieve protection information is the AXMEDIS object identifier, the object version and the protection stamp. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsdl/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:AXOID i:type="d:string">AXOID</wn0:AXOID> <wn0:objectVersion i:type="d:string">objectVersion</wn0:objectVersion> <wn0:protectionStamp i:type="d:string">protectionStamp</wn0:protectionStamp> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" </pre> |

| | |
|-------------------------|--|
| | <pre> xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:ProtectionInformation_Response i:nil="true"/> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>xsd:string AXOID: Identifier of the AXMEDIS object whose protection information is going to be stored</p> <p>xsd:string version: Version of the AXMEDIS object whose protection information is going to be stored</p> <p>xsd:string protectionStamp: Identifier of the protection information</p> |
| Output parameters | xsd:protectionInfo: The protection information to be stored. It is represented with a class, but maybe only a string will be needed |
| Method | generateKey |
| Description | This method permits the creation of a key for protecting an AXMEDIS object. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:algorithm i:type="wn1:Algorithm"> <wn1:algorithm i:type="d:int">0</wn1:algorithm> </wn0:algorithm> <wn0:keyLength i:type="d:int">32</wn0:keyLength> <wn0:parameters i:type="wn2:Vector"> <wn2:Items/> </wn0:parameters> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/wsd/default/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:KeyAX_Response i:nil="true"/> </e:Body> </e:Envelope> </pre> |
| Input parameters | <p>Algorithm: The ciphering / deciphering algorithm</p> <p>xsd:int keyLength: The length of the key to be generated</p> <p>Vector: A vector of the parameters needed to generate the key. They depend on the algorithm used</p> |
| Output parameters | KeyAX: The newly generated key. It will probably could be modeled with a byte array in order to simplify the WS |

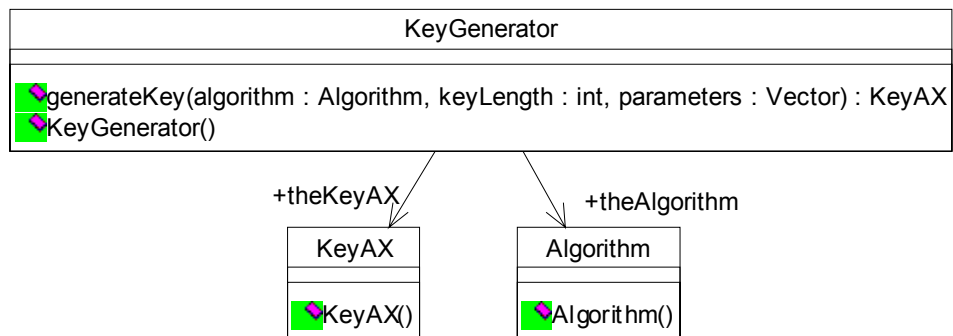
4.16 Key Generator (FUPF)

| Module Profile | |
|--------------------------------|----------------------|
| Key Generator | |
| Executable or Library(Support) | Executable |
| Single Thread or Multithread | Single Thread |
| Language of Development | C / C++ |
| Responsible Name | |
| Responsible Partner | FUPF |
| Status (proposed/approved) | Proposed |
| Platforms supported | PC (Linux / Windows) |

| | | |
|------------------------------|-----------------------------------|--|
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| Any | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| Openssl | | |
| | | |
| | | |

4.16.1 Key Generator module architecture

Next figure shows the UML diagram describing the Key Generator module. Nevertheless, this module is based on OpenSSL, so it is not object oriented in nature, but C++ interface could be provided. The functionality of this module has already been explained in the previous section, Protection Info Manager.



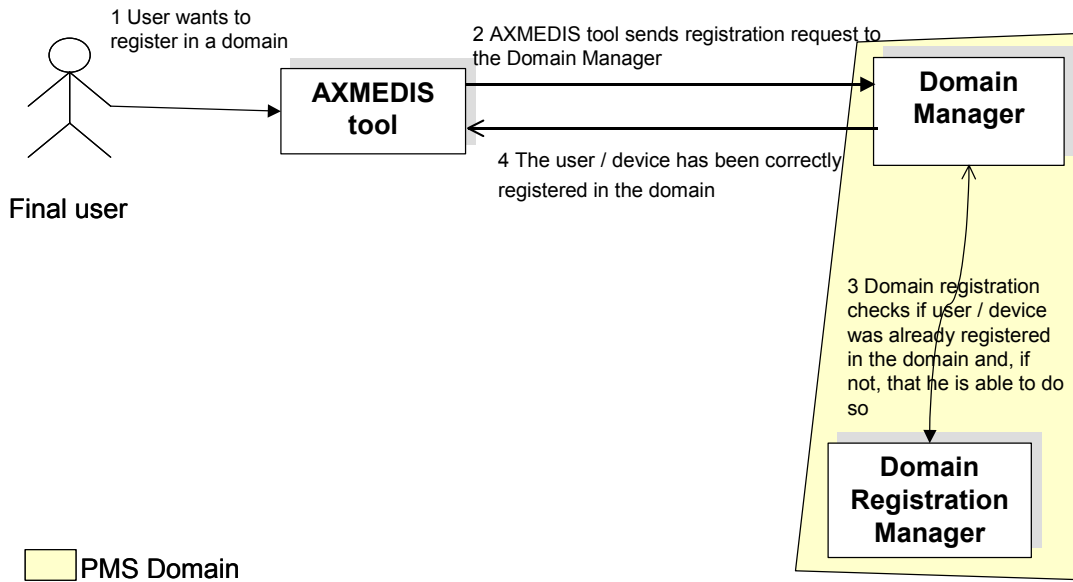
Key generator class diagram

4.17 Domain Manager (FUPF)

This module, together with the domain registration manager, keeps track of the users that are associated to a domain, giving them the possibility to register, unregister and, in general, to manage the domains available for a user.

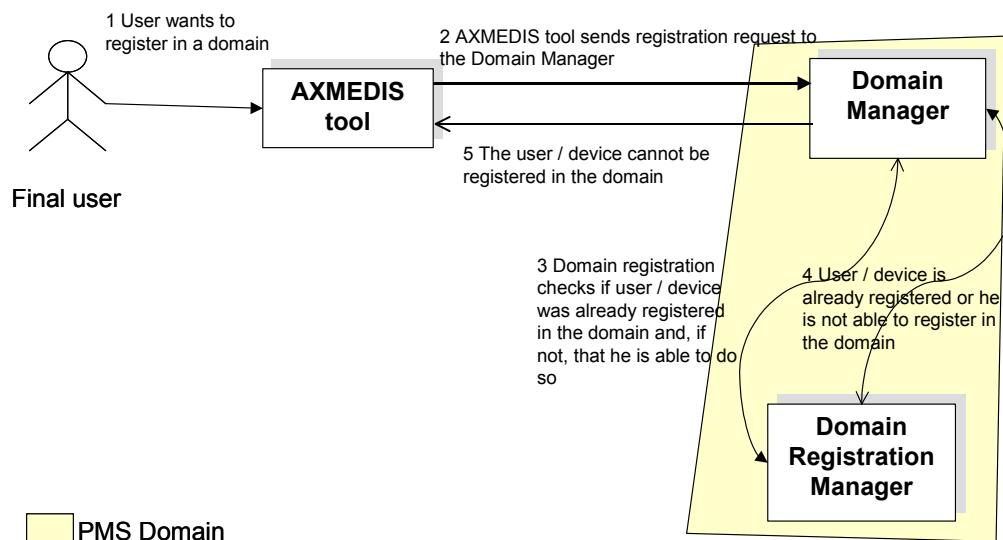
4.17.1 Domain Manager scenarios

The scenario described defines how a user can be registered in a domain.



Successful registration in a domain

The scenario described next shows how a user cannot be registered in a domain.



Unsuccessful registration in a domain

4.17.2 Module Profile definition

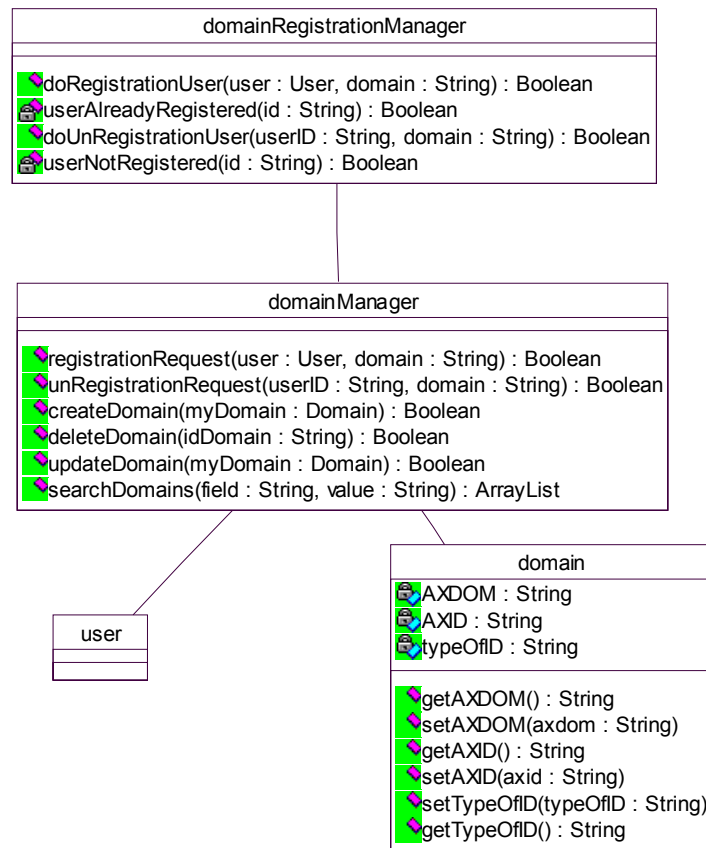
| Module Profile | | |
|--------------------------------|-----------------------------------|--|
| Domain Manager | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java/C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | proposed | |
| Platforms supported | PC | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| DRM Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| XML | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |

| Module Profile | | |
|------------------------------------|---------------------------------|---|
| Domain Registration Manager | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java/C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | proposed | |
| Platforms supported | Any supported by Java | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |

| | | |
|----------------|-----------------------------------|--|
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.17.3 Architecture of Domain Manager and Domain Registration Manager Module

Next figure describes the UML diagram of the Domain management related modules.



Domain Manager and Domain Registration manager modules

4.17.4 Domain Manager WEB Service formalisation

This section is provided for clarity, as the Domain Manager is not accessible as a web service but from the corresponding DRM support. The exposed operations are defined at the DRM support section, but here we explain in detail the operations for Domain Manager module.

| DomainManager | |
|-----------------------|--|
| Methods | Description |
| searchDomains | This function is used to return a list of domains with a certain value in a certain field. |
| createDomain | This function is used to create a new domain. |
| deleteDomain | This function is used to delete a domain with the idDomain selected. |
| updateDomain | This function is used to update a domain that already exists |
| registrationRequest | This function is used to register a certain user in a domain with the name selected. |
| unRegistrationRequest | This function is used to unregister a certain user in a domain with the name selected. |

| DomainManager | |
|---------------|--|
| WSDL | <pre> <?xml version="1.0" encoding="UTF-8"?> <wsdl:definitions name="DomainPackage.DomainManager" targetNamespace="urn:DomainPackage.DomainManager" xmlns:tns="urn:DomainPackage.DomainManager" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayList"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/containers/literal/ms.net" xmlns:xns5="urn:DomainPackage.DomainManager"> <xsd:import namespace="urn:DomainPackage.DomainManager"/> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:element name="field" type="xsd:string" nillable="true"/> <xsd:element name="value" type="xsd:string" nillable="true"/> <xsd:element name="ArrayList_Response" type="xns4:ArrayList" nillable="true"/> <xsd:element name="myDomain" type="xns5:Domain" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="idDomain" type="xsd:string" nillable="true"/> <xsd:element name="myUser" type="xns5:User" nillable="true"/> <xsd:element name="myDomain_1" type="xsd:string" nillable="true"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="myDomain_2" type="xsd:string" nillable="true"/> </xsd:schema> <xsd:schema targetNamespace="urn:DomainPackage.DomainManager" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:DomainPackage.DomainManager" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="Domain"> <xsd:sequence> <xsd:element name="AXDOM" type="xsd:string" nillable="true"/> <xsd:element name="AXID" type="xsd:string" nillable="true"/> <xsd:element name="typeOfID" type="xsd:string" nillable="true"/> </xsd:sequence> </xsd:complexType> </xsd:schema> </wsdl:types> </pre> |

```

</xsd:complexType>
<xsd:complexType name="User">
  <xsd:sequence/>
</xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="DomainManager_registrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_1_Request">
  <wsdl:part name="userID" element="ns0:userID"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_2"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_1_Request">
  <wsdl:part name="idDomain" element="ns0:idDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_deleteDomain_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_createDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_unRegistrationRequest_Response">
  <wsdl:part name="response" element="ns0:boolean_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_updateDomain_1_Request">
  <wsdl:part name="myDomain" element="ns0:myDomain"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_Response">
  <wsdl:part name="response" element="ns0:ArrayList_Response"/>
</wsdl:message>
<wsdl:message name="DomainManager_registrationRequest_1_Request">
  <wsdl:part name="myUser" element="ns0:myUser"/>
  <wsdl:part name="myDomain" element="ns0:myDomain_1"/>
</wsdl:message>
<wsdl:message name="DomainManager_searchDomains_1_Request">
  <wsdl:part name="field" element="ns0:field"/>
  <wsdl:part name="value" element="ns0:value"/>
</wsdl:message>
<wsdl:portType name="DomainManager">
  <wsdl:operation name="searchDomains" parameterOrder="field value">
    <wsdl:input message="tns:DomainManager_searchDomains_1_Request"/>
    <wsdl:output message="tns:DomainManager_searchDomains_Response"/>
  </wsdl:operation>
  <wsdl:operation name="createDomain" parameterOrder="myDomain">
    <wsdl:input message="tns:DomainManager_createDomain_1_Request"/>
    <wsdl:output message="tns:DomainManager_createDomain_Response"/>
  </wsdl:operation>
  <wsdl:operation name="deleteDomain" parameterOrder="idDomain">
    <wsdl:input message="tns:DomainManager_deleteDomain_1_Request"/>
    <wsdl:output message="tns:DomainManager_deleteDomain_Response"/>
  </wsdl:operation>
  <wsdl:operation name="updateDomain" parameterOrder="myDomain">
    <wsdl:input message="tns:DomainManager_updateDomain_1_Request"/>
    <wsdl:output message="tns:DomainManager_updateDomain_Response"/>
  </wsdl:operation>
  <wsdl:operation name="registrationRequest" parameterOrder="myUser myDomain">
    <wsdl:input message="tns:DomainManager_registrationRequest_1_Request"/>
    <wsdl:output message="tns:DomainManager_registrationRequest_Response"/>
  </wsdl:operation>
  <wsdl:operation name="unRegistrationRequest" parameterOrder="userID myDomain">
    <wsdl:input message="tns:DomainManager_unRegistrationRequest_1_Request"/>
    <wsdl:output message="tns:DomainManager_unRegistrationRequest_Response"/>
  </wsdl:operation>

```

| | |
|--|---|
| | <pre> </wsdl:portType> <wsdl:binding name="DomainManager" type="tns:DomainManager"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/> <wsdl:operation name="searchDomains"> <map:java-operation name="searchDomains" signature="KExqYXZhL2xhbmvcvU3RyaW5nO0xqYXZhL2xhbmvcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGlzdDs="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#searchDomains#KExqYXZhL2xhbmvcvU3RyaW5nO0xqYXZhL2xhbmvcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGlzdDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="createDomain"> <map:java-operation name="createDomain" signature="KExEb21haW5QYWNrYWdlL0RvbWFPbjspWg="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#createDomain#KExEb21haW5QYWNrYWdlL0RvbWFPbjspWg=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="deleteDomain"> <map:java-operation name="deleteDomain" signature="KExqYXZhL2xhbmvcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#deleteDomain#KExqYXZhL2xhbmvcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="updateDomain"> <map:java-operation name="updateDomain" signature="KExEb21haW5QYWNrYWdlL0RvbWFPbjspWg="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#updateDomain#KExEb21haW5QYWNrYWdlL0RvbWFPbjspWg=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="registrationRequest"> <map:java-operation name="registrationRequest" signature="KExEb21haW5QYWNrYWdlL1VzZXI7TGphdmEvbGFuZy9TdHJpbmc7KVo="/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#registrationRequest#KExEb21haW5QYWNrYWdlL1VzZXI7TGphdmEvbGFuZy9TdHJpbmc7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="unRegistrationRequest"> <map:java-operation name="unRegistrationRequest" </pre> |
|--|---|

| | |
|-------------------------|--|
| | <pre>signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainManagerDomainManager#unRegistrationRequest#KExqYXZhL2xhb mcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="DomainManager"> <wsdl:port name="DomainManager" binding="tns:DomainManager"> <soap:address location="http://EFPC016:6060/DomainManager"/> </wsdl:port> </wsdl:service> </wsdl:definitions></pre> |
| Method | searchDomains |
| Description | This function is used to return a list of domains with a certain value in a certain field. |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:ArrayList_Response i:type="wn2:ArrayList"> <wn2:Items> <wn2:Item i:type="wn0:Domain"/> <wn2:Item i:type="wn0:Domain"/> </wn2:Items> </wn1:ArrayList_Response> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:ArrayList_Response i:type="wn2:ArrayList"> <wn2:Items> <wn2:Item i:type="wn0:Domain"/> <wn2:Item i:type="wn0:Domain"/> </wn2:Items> </wn1:ArrayList_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | <p>xsd:String field : field of the domain to do the search</p> <p>xsd:String value : value of the field</p> |
| Output parameters | <p>Sequence of ax:domain that is a complex type formed by:</p> <p>xsd: string AXDOM xsd: string AXID xsd: string typeOfID</p> <p>xsd: string getAXDOM xsd: string getAXID xsd: string getTypeOfID</p> <p>setAXDOM xsd: String setAXID xsd: String setTypeOfID xsd : String</p> |

| | |
|-------------------------|---|
| | |
| Method | createDomain |
| Description | This function is used to create a new domain. |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myDomain i:type="wn0:Domain"> <wn0:AXDOM i:nil="true"/> <wn0:AXID i:nil="true"/> <wn0:typeOfID i:nil="true"/> </wn1:myDomain> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:boolean_Response i:type="d:boolean">true</wn1:boolean_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | ax:Domain is already defined |
| Output parameters | xsd: boolean : 0 means OK |
| Method | updateDomain |
| Description | This function is used to update a domain. |
| Input parameters | ax:Domain is already defined |
| Output parameters | xsd: boolean : 0 means OK |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myDomain i:type="wn0:Domain"> <wn0:AXDOM i:nil="true"/> <wn0:AXID i:nil="true"/> <wn0:typeOfID i:nil="true"/> </wn1:myDomain> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:boolean_Response i:type="d:boolean">true</wn1:boolean_Response> </e:Body> </e:Envelope></pre> |
| Method | registrationRequest |
| Description | This function is used to a registration of an user in a certain domain |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/"</pre> |

| | |
|-------------------------|--|
| | <pre> xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myUser i:type="wn0:User"/> <wn1:myDomain_1 i:type="d:string"/> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:boolean_Response i:type="d:boolean">true</wn1:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <pre> xsd: string domain : name of the domain ax: user </pre> |
| Output parameters | <pre> xsd: boolean : 0 means OK </pre> |
| Method | deleteDomain |
| Description | This function is used to delete a domain with an idDomain identifier |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:idDomain i:type="d:string"/> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:boolean_Response i:type="d:boolean">true</wn1:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | <pre> xsd: string idDomain : domain id </pre> |
| Output parameters | <pre> xsd: boolean : 0 means OK </pre> |
| Method | unRegistrationRequest |
| Description | This function is used to an unregistration of an user in a certain domain |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myUser_1 i:type="d:string"/> <wn1:myDomain_1 i:type="d:string"/> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:boolean_Response i:type="d:boolean">true</wn1:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Input | <pre> xsd: string userID : name of the domain </pre> |

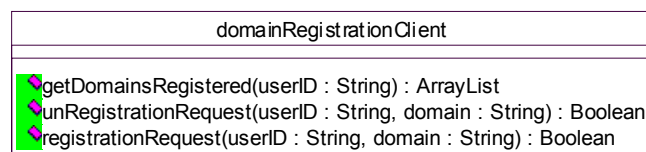
| | |
|-------------------|---|
| parameters | xsd: string domain : name of the domain |
| Output parameters | xsd: boolean : 0 means OK |

4.17.5 Domain registration client

| Module Profile | | |
|--------------------------------|-----------------------------------|--|
| Domain Registration Client | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | proposed | |
| Platforms supported | Windows | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

4.17.5.1 Architecture of Domain Registration Client Module

Next figure describes the UML diagram of the Domain Registration Client module.



Class Diagram of Domain Registration Client module

| Domain Registration Client | |
|----------------------------|--|
| Methods | Description |
| getDomainsRegistered | This function is used to get the domain registered by a certain user. |
| registrationRequest | This function is used to register a certain user in a domain with the name selected. |
| unRegistrationRequest | This function is used to unregister a certain user in a domain with the name selected. |

4.17.5.2 WSDL Description

| domainRegistrationClient | |
|--------------------------|--|
| WSDL | <pre> <?xml version="1.0"?> <wsdl:definitions name="DomainPackage.DomainRegistrationClient" targetNamespace="urn:DomainPackage.DomainRegistrationClient" xmlns:tns="urn:DomainPackage.DomainRegistrationClient" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> <wsdl:types> <xsd:schema targetNamespace="urn:DomainPackage.DomainRegistrationClient" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:DomainPackage.DomainRegistrationClient" xmlns:map="http://systinet.com/mapping/"> <xsd:complexType name="User"> <xsd:annotation> <xsd:appinfo> <map:java-type name="DomainPackage.User"/> </xsd:appinfo> </xsd:annotation> <xsd:sequence/> </xsd:complexType> </xsd:schema> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="urn:DomainPackage.DomainRegistrationClient" xmlns:xns5="http://systinet.com/containers/literal/ms.net"> <xsd:import namespace="http://systinet.com/containers/literal/ms.net"/> <xsd:import namespace="urn:DomainPackage.DomainRegistrationClient"/> <xsd:element name="myUser" type="xns4:User" nillable="true"/> <xsd:element name="myDomain" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="userID" type="xsd:string" nillable="true"/> <xsd:element name="idUserario" type="xsd:string" nillable="true"/> <xsd:element name="ArrayList_Response" type="xns5:ArrayList" nillable="true"/> </xsd:schema> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/containers/literal/ms.net" targetNamespace="http://systinet.com/containers/literal/ms.net" elementFormDefault="qualified"> <xsd:complexType name="List"> <xsd:sequence> <xsd:element name="Items" type="tns:ArrayOfAnyType"/> </xsd:sequence> </xsd:complexType> <xsd:complexType name="ArrayList"> <xsd:complexContent mixed="false"> <xsd:extension base="tns:List"/> </xsd:complexContent> </xsd:complexType> </xsd:schema> </wsdl:types> <wsdl:message name="DomainRegistrationClient_unRegistrationRequest_1_Request"> <wsdl:part name="userID" element="ns0:userID"/> <wsdl:part name="myDomain" element="ns0:myDomain"/> </wsdl:message> </pre> |

| | |
|--|---|
| | <pre> <wsdl:message name="DomainRegistrationClient_unRegistrationRequest_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="DomainRegistrationClient_getDomainsRegistered_1_Request"> <wsdl:part name="idUsuario" element="ns0:idUsuario"/> </wsdl:message> <wsdl:message name="DomainRegistrationClient_registrationRequest_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> <wsdl:message name="DomainRegistrationClient_registrationRequest_1_Request"> <wsdl:part name="myUser" element="ns0:myUser"/> <wsdl:part name="myDomain" element="ns0:myDomain"/> </wsdl:message> <wsdl:message name="DomainRegistrationClient_getDomainsRegistered_Response"> <wsdl:part name="response" element="ns0:ArrayList_Response"/> </wsdl:message> <wsdl:portType name="DomainRegistrationClient"> <wsdl:operation name="registrationRequest" parameterOrder="myUser myDomain"> <wsdl:input message="tns:DomainRegistrationClient_registrationRequest_1_Request"/> <wsdl:output message="tns:DomainRegistrationClient_registrationRequest_Response"/> </wsdl:operation> <wsdl:operation name="unRegistrationRequest" parameterOrder="userID myDomain"> <wsdl:input message="tns:DomainRegistrationClient_unRegistrationRequest_1_Request"/> <wsdl:output message="tns:DomainRegistrationClient_unRegistrationRequest_Response"/> </wsdl:operation> <wsdl:operation name="getDomainsRegistered" parameterOrder="idUsuario"> <wsdl:input message="tns:DomainRegistrationClient_getDomainsRegistered_1_Request"/> <wsdl:output message="tns:DomainRegistrationClient_getDomainsRegistered_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="DomainRegistrationClient" type="tns:DomainRegistrationClient"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/> <wsdl:operation name="registrationRequest"> <map:java-operation name="registrationRequest" signature="KExEb21haW5QYWNrYWdlL1VzZXI7TGphdmEvdGFuZy9TdHJpbmc7KVo="/> <soap:operation soapAction="urn:DomainPackage.DomainRegistrationClientDomainRegistrationClient#registrationRequest#KExEb21haW5QYWNrYWdlL1VzZXI7TGphdmEvdGFuZy9TdHJpbmc7KVo=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="unRegistrationRequest"> <map:java-operation name="unRegistrationRequest" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla"/> <soap:operation soapAction="urn:DomainPackage.DomainRegistrationClientDomainRegistrationClient#unRegistrationRequest#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="getDomainsRegistered"> <map:java-operation name="getDomainsRegistered" signature="KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGldDs="/> <soap:operation soapAction="urn:DomainPackage.DomainRegistrationClientDomainRegistrationClient#getDomainsRegistered#KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS91dGlsL0FycmF5TGldDs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </pre> |
|--|---|

| | |
|-------------------------|--|
| | <pre> </wsdl:operation> </wsdl:binding> <wsdl:service name="DomainRegistrationClient"> <wsdl:port name="DomainRegistrationClient" binding="tns:DomainRegistrationClient"> <soap:address location="http://EFPC016:6060/DomainRegistrationClient/"> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
| Method | registrationRequest |
| Description | This function is used to do a registration of an user in a certain domain |
| Input parameters | ax: user xsd: string domain : name of the domain |
| Output parameters | xsd: boolean : 0 means OK |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:myUser i:type="wn0:User"/> <wn1:myDomain_1 i:type="d:string"/> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="urn:DomainPackage.DomainManager" xmlns:wn1="http://systinet.com/xsd/SchemaTypes/" xmlns:wn2="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn1:boolean_Response i:type="d:boolean">true</wn1:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Method | unRegistrationRequest |
| Description | This function is used to an unregistration of an user in a certain domain |
| Input parameters | xsd: string userID : id of the user xsd: string domain : name of the domain |
| Output parameters | xsd: boolean : 0 means OK |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:userID i:type="d:string"/> <wn0:myDomain i:type="d:string"/> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:boolean_Response i:type="d:boolean">true</wn0:boolean_Response> </e:Body> </e:Envelope> </pre> |
| Method | getDomainsRegistered |
| Description | This function is used to get the domain registered by a certain user |
| Input parameters | xsd: string userID : id of the user |
| Output parameters | Sequence of xsd: String domain |

| | |
|-------------------------|---|
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/containers/literal/ms.net"> <e:Body> <wn0:idUsuario i:type="d:string">userID</wn0:idUsuario> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/" xmlns:wn1="http://systinet.com/containers/literal/ms.net" xmlns:wn2="http://idoox.com/package/"> <e:Body> <wn0:ArrayList_Response i:type="wn1:ArrayList"> <wn1:Items> <wn1:Item i:type="wn2:DomainPackage.Domain"> <AXID i:nil="true"/> <typeOfID i:nil="true"/> <AXDOM i:nil="true"/> </wn1:Item> <wn1:Item i:type="wn2:DomainPackage.Domain"> <AXID i:nil="true"/> <typeOfID i:nil="true"/> <AXDOM i:nil="true"/> </wn1:Item> </wn1:Items> </wn0:ArrayList_Response> </e:Body> </e:Envelope></pre> |

4.18 Secure cache Manager (FUPF)

| Module Profile | | |
|---|-----------------------------------|---|
| Secure Cache Manager | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Single Thread | |
| Language of Development | C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| License Manager, Domain Manager, Domain Registration Client | | |
| Content Consumption status | | |
| Protection Information Manager | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |

| | | |
|----------------|---------------------------------|--|
| | | |
| Used Libraries | Name of the library and version | License status: GPL, LGPL, PEK, proprietary, authorized or not |
| Openssl | | |
| | | |
| | | |

4.18.1 Description of the module

This module provides the functionality needed to access to information stored in the Local Cache. This module is internally used by the DRM support corresponding to each PMS to stored information that is used for authorising user to perform actions when it is not connected.

The information managed by secure cache manager has to be stored in a secure way, in order to avoid as much as possibly user manipulation of the action logs and general status information. To do so, openssl cryptographic functions will be used.

Not all the information has to be ciphered before storing it, as then it will be very difficult to find it. For instance, the AXOID should be in clear when we want to request the protection information associated to an object, but the protection information itself should be ciphered for storing it.

The most efficient way of performing the ciphering is to use symmetric ciphering. The symmetric key cannot be stored in the user's device, but it should be calculated when some ciphering operation is requested.

The API of OpenSSL serves as an interface to a range of security functions. In this module we will use the ones that provide cryptographic functionality. In next section, OpenSSL library is explained briefly.

4.18.1.1 OpenSSL

The OpenSSL Project (<http://www.openssl.org>) is a collaborative effort to develop a robust, commercial-grade, full-featured and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan and develop the OpenSSL toolkit and its related documentation.

OpenSSL is based on the SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit is licensed under an Apache-style licence, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions.

4.18.1.2 Windows Version of OpenSSL

The Windows version of OpenSSL library can be downloaded from <http://www.slproweb.com/products/Win32OpenSSL.html>

4.18.1.3 Cryptographic functions provided by OpenSSL

The OpenSSL crypto library implements a wide range of cryptographic algorithms used in various Internet standards. The services provided by this library are used by the OpenSSL implementations of SSL, TLS and S/MIME, and they have also been used to implement SSH, OpenPGP, and other cryptographic standards.

libcrypto consists of a number of sub-libraries that implement the individual algorithms. The functionality includes symmetric encryption, public key cryptography and key agreement, certificate handling, AXMEDIS Project

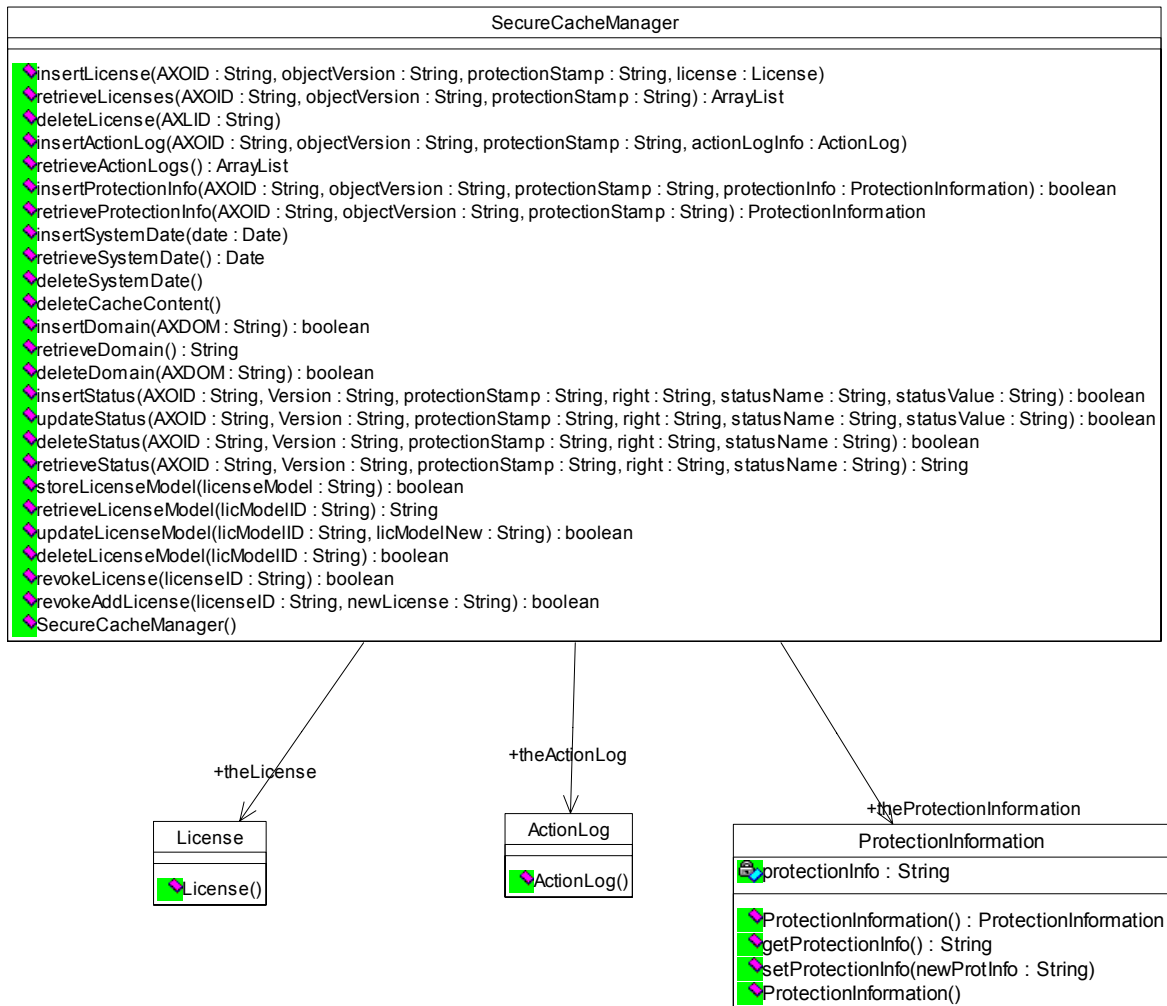
cryptographic hash functions and a cryptographic pseudo-random number generator. It is briefly described next.

- Symmetric ciphers: des, idea, rc2, rc4, rc5.
- Authentication codes, hash functions: hmac, md4, md5, sha.
- Input/Output data encoding: asn1, bio, evp, pem, pkcs7, pkcs12.

The functions provided by each algorithm depend on its nature, and are independently described in the library documentation (for details, see <http://www.openssl.org/docs/crypto/crypto.html>).

4.18.2 Architecture of the module

The following figure shows the UML diagram of the secure cache manager module.



Secure cache manager class diagram

Secure Cache Manager

| Method | Description |
|------------------------|--|
| insertLicense | This method allows the insertion of a license into the local cache by the LicenseManager module. |
| retrieveLicense | This method allows the retrieval of a license from the local cache by the LicenseManager module. |
| deleteLicense | This method allows the deletion of a license from the local cache by the LicenseManager module. |
| insertActionLog | This method inserts an action log into the local cache info. It is called by the Content Consumption status module. |
| retrieveActionLogs | This method retrieves the action logs stored into the local cache info. It is called by the Content Consumption status module. |
| insertProtectionInfo | This method stores the protection information associated to a given object, version and protection stamp. This information has to be stored ciphered. |
| retrieveProtectionInfo | This method retrieves the protection information associated to a given object, version and protection stamp. |
| insertSystemDate | This method stores the current system date in order to perform local checks over the operations done over the local cache info. |
| retrieveSystemDate | This method returns the last system date stored into the local cache info. |
| deleteSystemDate | This method deletes the system date from the local cache info. |
| deleteCacheContent | This method deletes the whole content of the local cache info. It should also invalidate it, as it is called when a tool has not been verified. |
| insertDomain | This method inserts the domain a user has been registered to. This information has to be stored ciphered. |
| retrieveDomain | This method retrieves the current domain a user is registered to. |
| deleteDomain | This method deletes the domain. This operation is called when the user request unregistration from the domain. |
| insertStatus | This method stores some status information associated to AXMEDIS objects usage in order to be able to perform local authorisations. This information has to be stored ciphered. It is used by the Authorisation support module. |
| updateStatus | This method updates some status information associated to AXMEDIS objects usage in order to be able to perform local authorisations. This information has to be stored ciphered. It is used by the Authorisation support module. |
| deleteStatus | This method deletes status information associated to AXMEDIS objects usage. |
| retrieveStatus | This method retrieves status information associated to AXMEDIS objects usage in order to be able to perform local authorisations. It is used by the Authorisation support module. |
| retrieveLicenseModel | This function returns a licenseModel stored in the License DB |
| deleteLicenseModel | This function deletes a licenseModel from the License DB |
| storeLicenseModel | This function inserts a license model in the License DB |
| updateLicenseModel | This function updates the licenseModel given a licenseModelID and a new description of licenseModel |
| revokeLicense | This function changes the status of a license to revoked in the License DB |
| revokeAddLicense | This function changes the status of a license to revoked and adds the license that substitutes it in the License DB |

4.19 Local Cache Information Format (FUPF)

Module Profile

| Local Cache Information Format | | |
|--------------------------------|-----------------------------------|--|
| Executable or Library(Support) | | |
| Single Thread or Multithread | | |
| Language of Development | | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Secure Cache Manager | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

The local cache information format has to deal with different kinds of information:

- Licenses
- Protection information
- Action logs
- Status information
- System information
- Domain information
- Other

For the case of a PC, we propose the use of a small relational database in order to be able to make quick insertions and retrievals of information.

For other kinds of devices, the format of the cache is not so clear, as we may need a “light” version of the Secure Cache Manager and also of the local cache. Depending on the device capabilities, the amount of information to be stored and the way of storing it may vary, and the best option has to be selected for each device.

4.19.1 File format (FUPF)

This is the description of the tables needed inside the cache. For the light version, the stored information may vary.

License

| Columns | Data type | Allow NULLs | Value/Range |
|-----------------|----------------|-------------|-------------|
| AXOID | C-Large Length | Not allowed | |
| ObjectVersion | C-Large Length | Not allowed | |
| ProtectionStamp | C-Large Length | Not allowed | |
| AXLID | C-Large Length | Not allowed | |
| License | C-Large Length | Not allowed | |

Column details

1. AXOID

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object ID.

2. ObjectVersion

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object version.

3. ProtectionStamp

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Indicates the way to protect the related object.

4. AXLID

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Identifier of the stored license.

5. License

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: The whole license.

ProtectionInformation

| Columns | Data type | Allow NULLs | Value/Range |
|-----------------------|----------------|-------------|-------------|
| AXOID | C-Large Length | Not allowed | |
| ObjectVersion | C-Large Length | Not allowed | |
| ProtectionStamp | C-Large Length | Not allowed | |
| ProtectionInformation | C-Large Length | Not allowed | |

Column details

1. AXOID

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object ID.

2. ObjectVersion

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object version.

3. ProtectionStamp

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Indicates the way to protect the related object.

4. ProtectionInformation

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Protection Information associated to the object.

ActionLog

| Columns | Data type | Allow NULLs | Value/Range |
|-----------------|----------------|-------------|-------------|
| AXOID | C-Large Length | Not allowed | |
| ObjectVersion | C-Large Length | Not allowed | |
| ProtectionStamp | C-Large Length | Not allowed | |
| ActionLog | C-Large Length | Not allowed | |

Column details**1. AXOID**

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object ID.

2. ObjectVersion

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object version.

3. ProtectionStamp

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Indicates the way to protect the related object.

4. ActionLog

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Information of an Action log, ciphered in a unique field.

Status

| Columns | Data type | Allow NULLs | Value/Range |
|-----------------|----------------|-------------|-------------|
| AXOID | C-Large Length | Not allowed | |
| ObjectVersion | C-Large Length | Not allowed | |
| ProtectionStamp | C-Large Length | Not allowed | |
| Right | C-Large Length | Not allowed | |
| FieldName | C-Large Length | Not allowed | |
| FieldValue | C-Large Length | Not allowed | |

Column details**1. AXOID**

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object ID.

2. ObjectVersion

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Pertinent Object version.

3. ProtectionStamp

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: Indicates the way to protect the related object.

4. Right

Physical data type: LONGTEXT

Allow NULLs: Not allowed
Notes: Right exercised over the object.

5. FieldName

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Status information name.

6. FieldValue

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Status information name.

SystemDate

| Columns | Data type | Allow NULLs | Value/Range |
|------------|----------------|-------------|-------------|
| SystemDate | C-Large Length | Not allowed | |

Column details**1. SystemDate**

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Stores the ciphered system date.

Domain

| Columns | Data type | Allow NULLs | Value/Range |
|---------|----------------|-------------|-------------|
| AXDOM | C-Large Length | Not allowed | |

Column details**1. AXDOM**

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: Pertinent User AXMEDIS Current Domain (if any).

4.20 License Generator (FUPF)

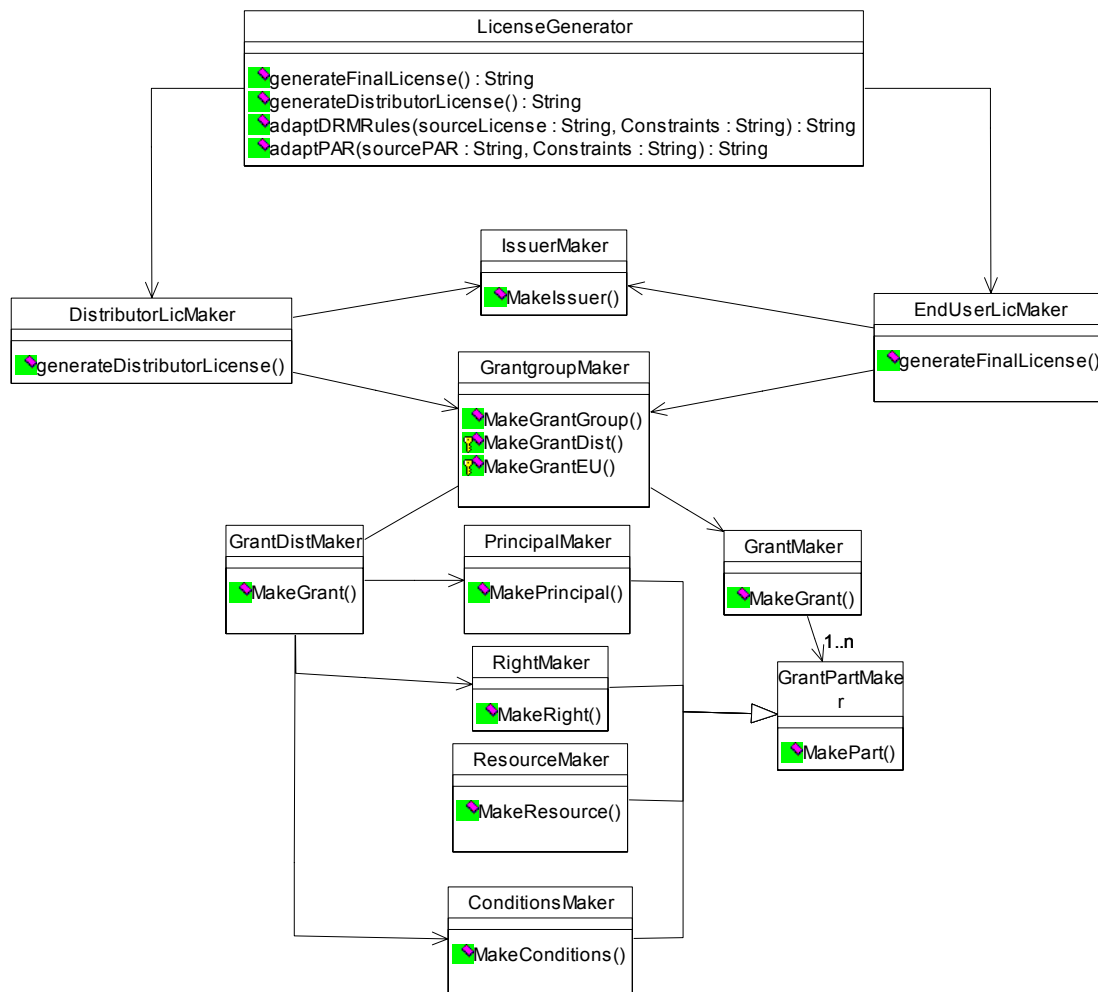
| Module Profile | | |
|--------------------------------|---------------------------------|---|
| License Generator | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | C++/Java | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | | |
| Platforms supported | | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |

| | | |
|----------------|-----------------------------------|--|
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

This module provides the functionality to create new licenses that define the governance of the AXMEDIS objects. It will use MPEG-21 REL as the primary rights expression language, and other languages, mechanisms will be added during the project.

4.20.1 Architecture of the License Generator

Next figure describes the License Generator class diagram.



License Generator class diagram

| License Generator | |
|----------------------------|---|
| Methods | Description |
| generateFinalLicense | Generates a license for an end-user, according to the userID, right, AXOID and conditions. |
| generateDistributorLicense | Generates a license for a distributor, according to the userID, right, grants that this user will be able to distribute and conditions. |
| adaptDRMRules | Adapts the given license using the given constraints |
| adaptPAR | Adapts the given PAR using the given constraints |

4.20.2 WSDL Interface

| License Generator | |
|-------------------|---|
| WSDL | <pre> <?xml version='1.0'?> <wsdl:definitions name='LicenseGenerator' targetNamespace='urn:LicenseGenerator' xmlns:map='http://systinet.com/mapping' xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/' xmlns:ns0='http://systinet.com/xsd/SchemaTypes/' xmlns:tns='urn:LicenseGenerator' xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'> <wsdl:types> <xsd:schema targetNamespace='http://systinet.com/xsd/SchemaTypes/' elementFormDefault='qualified' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:tns='http://systinet.com/xsd/SchemaTypes/'> <xsd:element name='userIDs' type='xsd:string' nillable='true'/> <xsd:element name='rights' type='xsd:string' nillable='true'/> <xsd:element name='AXOIDs' type='xsd:string' nillable='true'/> <xsd:element name='conditions' type='xsd:string' nillable='true'/> <xsd:element name='string_Response' type='xsd:string' nillable='true'/> <xsd:element name='grants' type='xsd:string' nillable='true'/> <xsd:element name='sourceLicense' type='xsd:string' nillable='true'/> <xsd:element name='constraints' type='xsd:string' nillable='true'/> <xsd:element name='PAR' type='xsd:string' nillable='true'/> </xsd:schema> </wsdl:types> <wsdl:message name='LicenseGenerator_adaptPAR_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseGenerator_generateDistributorLicense_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseGenerator_adaptDRMRules_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseGenerator_generateFinalLicense_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseGenerator_adaptPAR_1_Request'> <wsdl:part name='PAR' element='ns0:PAR'/> <wsdl:part name='constraints' element='ns0:constraints'/> </wsdl:message> <wsdl:message name='LicenseGenerator_generateDistributorLicense_1_Request'> </pre> |

```
<wsdl:part name='userIds' element='ns0:userIds'/>
<wsdl:part name='rights' element='ns0:rights'/>
<wsdl:part name='grants' element='ns0:grants'/>
<wsdl:part name='AXOIDs' element='ns0:AXOIDs'/>
<wsdl:part name='conditions' element='ns0:conditions'/>
</wsdl:message>
<wsdl:message name='LicenseGenerator_adaptDRMRules_1_Request'>
  <wsdl:part name='sourceLicense' element='ns0:sourceLicense'/>
  <wsdl:part name='constraints' element='ns0:constraints'/>
</wsdl:message>
<wsdl:message name='LicenseGenerator_generateFinalLicense_1_Request'>
  <wsdl:part name='userIDs' element='ns0:userIDs'/>
  <wsdl:part name='rights' element='ns0:rights'/>
  <wsdl:part name='AXOIDs' element='ns0:AXOIDs'/>
  <wsdl:part name='conditions' element='ns0:conditions'/>
</wsdl:message>

<wsdl:portType name='LicenseGenerator'>
  <wsdl:operation name='generateFinalLicense' parameterOrder='userIDs rights AXOIDs conditions'>
    <wsdl:input message='tns:LicenseGenerator_generateFinalLicense_1_Request'/>
    <wsdl:output message='tns:LicenseGenerator_generateFinalLicense_Response'/>
  </wsdl:operation>
  <wsdl:operation name='generateDistributorLicense' parameterOrder='userIDs rights grants AXOIDs conditions'>
    <wsdl:input message='tns:LicenseGenerator_generateDistributorLicense_1_Request'/>
    <wsdl:output message='tns:LicenseGenerator_generateDistributorLicense_Response'/>
  </wsdl:operation>
  <wsdl:operation name='adaptDRMRules' parameterOrder='sourceLicense constraints'>
    <wsdl:input message='tns:LicenseGenerator_adaptDRMRules_1_Request'/>
    <wsdl:output message='tns:LicenseGenerator_adaptDRMRules_Response'/>
  </wsdl:operation>
  <wsdl:operation name='adaptPAR' parameterOrder='PAR constraints'>
    <wsdl:input message='tns:LicenseGenerator_adaptPAR_1_Request'/>
    <wsdl:output message='tns:LicenseGenerator_adaptPAR_Response'/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name='LicenseGenerator' type='tns:LicenseGenerator'>
  <soap:binding transport='http://schemas.xmlsoap.org/soap/http' style='document'/>
  <wsdl:operation name='generateFinalLicense'>
    <map:java-operation name='generateFinalLicense'
signature=KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs='/'>
      <soap:operation
soapAction='urn:LicenseGeneratorLicenseGenerator#generateFinalLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=' style='document'/>
        <wsdl:input>
          <soap:body use='literal'/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use='literal'/>
        </wsdl:output>
      </wsdl:operation>
    <wsdl:operation name='generateDistributorLicense'>
      <map:java-operation name='generateDistributorLicense'
signature=KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs='/'>
        <soap:operation
soapAction='urn:LicenseGeneratorLicenseGenerator#generateDistributorLicense#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=' style='document'/>
          <wsdl:input>
            <soap:body use='literal'/>
          </wsdl:input>
          <wsdl:output>
            <soap:body use='literal'/>
          </wsdl:output>
        </wsdl:operation>
      <wsdl:operation name='adaptDRMRules'>
        <map:java-operation name='adaptDRMRules'
```

| | |
|-------------------------------|--|
| | <pre> signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptDRMRules#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="adaptPAR"> <map:java-operation name="adaptPAR" signature="KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs="/> <soap:operation soapAction="urn:LicenseGeneratorLicenseGenerator#adaptPAR#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmZs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="LicenseGenerator"> <wsdl:port name="LicenseGenerator" binding="tns:LicenseGenerator"> <soap:address location="http://efpc037:6060/LicenseGenerator"/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
| Method | generateFinalLicense |
| Description | generates a license for an end-user, according to the userID, right, AXOID and conditions. |
| Request Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:userIDs i:type="d:string">AXUID1</wn0:userIDs> <wn0:rights i:type="d:string">play</wn0:rights> <wn0:AXOIDs i:type="d:string">AXOID1</wn0:AXOIDs> <wn0:conditions i:type="d:string">&lt;conditions/&gt;</wn0:conditions> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:string_Response i:type="d:string">&lt;license/&gt;</wn0:string_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:string userIDs: AXMEDIS users identifications xsd:string rights: rights that the AXMDIS user will exercise xsd:string AXOIDs: AXMEDIS objects against which the rights will be exercised xsd:string conditions: condntions that must be achieved |
| Output parameters | xsd:string : the final user license |
| Method | generateDistributorLicense |
| Description | generates a license for a distributor, according to the userID, right, grants that this user will be able to distribute and conditions. |

| | |
|-------------------------|--|
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:userIDs i:type="d:string">AXUID2</wn0:userIDs> <wn0:rights i:type="d:string">issue</wn0:rights> <wn0:grants i:type="d:string">&lt;grant&gt;</wn0:grants> <wn0:AXOIDs i:type="d:string">AXOID2</wn0:AXOIDs> <wn0:conditions i:type="d:string">&lt;conditions&gt;</wn0:conditions> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:string_Response i:type="d:string">&lt;license&gt;</wn0:string_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | <p>xsd:string userIDs: AXMEDIS users identifications</p> <p>xsd:string rights: rights that the AXMDIS user will exercise</p> <p>xsd:string grants: grants that the distributor will issue</p> <p>xsd:string AXOIDs: AXMEDIS objects against which the rights will be exercised</p> <p>xsd:string conditions: condions that must be achieved</p> |
| Output parameters | xsd:string : the distributor license |
| Method | adaptDRMRules |
| Description | Adapts the given license using the given constraints |
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:sourceLicense i:type="d:string">&lt;r:license&gt;</wn0:sourceLicense> <wn0:constrains i:type="d:string">adapt</wn0:constrains> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:string_Response i:type="d:string">&lt;r:license&gt;</wn0:string_Response> </e:Body> </e:Envelope></pre> |
| Input parameters | <p>SourceLicense:String, the license to be adapted</p> <p>Constraints:String, the adaptation constraints to be applied over the license</p> |
| Output parameters | String, the adapted license |
| Method | adaptPAR |
| Description | Adapts the given PAR using the given constraints |
| Request Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"> <e:Body> <wn0:PAR i:type="d:string">&lt;PAR&gt;</wn0:PAR> <wn0:constrains i:type="d:string">adapt</wn0:constrains> </e:Body> </e:Envelope></pre> |
| Response Sample Message | <pre><e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes/"></pre> |

| | |
|-------------------|---|
| | <pre> <e:Body> <wn0:string_Response i:type="d:string">&lt;PAR/&gt;</wn0:string_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | PAR:String, the PAR to be adapted Constraints:String, the adaptation constraints to be applied over the PAR |
| Output parameters | String, the adapted PAR |

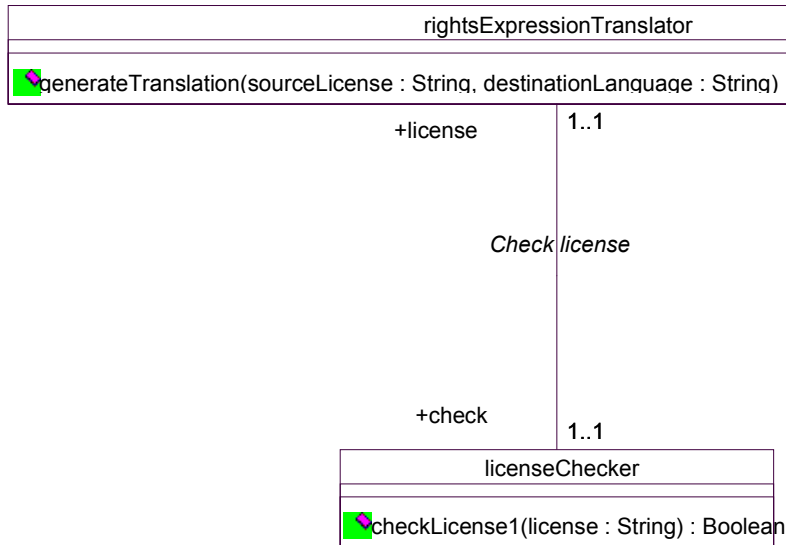
4.21 Rights Expression Translator (FUPF)

| Module Profile | | |
|-------------------------------------|-----------------------------------|--|
| Rights Expression Translator | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | Java/C++ | |
| Responsible Name | | |
| Responsible Partner | FUPF | |
| Status (proposed/approved) | proposed | |
| Platforms supported | Any supported by Java | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| DRM Support | | |
| | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a section |
| XML, XSLT | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| Xerces | | |
| | | |
| | | |

This module provides the functionality to translate licenses from one REL (source) to another one (destination).

4.21.1 Architecture of the Rights Expression Translator

Next figure describes a preliminary version of the Rights Expression Translator class diagram.



Rights Expression Translator class diagram

4.21.2 Rights Expression Translator WEB Service formalisation

| Rights Expression Translator | |
|------------------------------|--|
| Methods | Description |
| generateTranslation | This function is used to return a destinationLicense. The system makes a translation from a sourceLicense to a destinationLicense. The destinationLicense, if the translation is possible, will be written in a languageDestination. |

| Rights Expression Translator | |
|------------------------------|---|
| WSDL | <pre> <?xml version="1.0"?> <wsdl:definitions name="translator.RightsExpressionTranslator" targetNamespace="urn:translator.RightsExpressionTranslator" xmlns:tns="urn:translator.RightsExpressionTranslator" xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:map="http://systinet.com/mapping/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> <wsdl:types> <xsd:schema targetNamespace="http://systinet.com/xsd/SchemaTypes/" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://systinet.com/xsd/SchemaTypes/"> <xsd:element name="sourceLicense" type="xsd:string" nillable="true"/> <xsd:element name="destinationLanguage" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:message name="RightsExpressionTranslator_generateTranslation_Response"> </pre> |

| | |
|-------------------------|---|
| | <pre> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="RightsExpressionTranslator_generateTranslation_1_Request"> <wsdl:part name="sourceLicense" element="ns0:sourceLicense"/> <wsdl:part name="destinationLanguage" element="ns0:destinationLanguage"/> </wsdl:message> <wsdl:portType name="RightsExpressionTranslator"> <wsdl:operation name="generateTranslation" parameterOrder="sourceLicense destinationLanguage"> <wsdl:input message="tns:RightsExpressionTranslator_generateTranslation_1_Request"/> <wsdl:output message="tns:RightsExpressionTranslator_generateTranslation_Response"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="RightsExpressionTranslator" type="tns:RightsExpressionTranslator"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/> <wsdl:operation name="generateTranslation"> <map:java-operation name="generateTranslation" signature="KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmLu Zzs="/> <soap:operation soapAction="urn:translator.RightsExpressionTranslatorRightsExpressionTranslator#generateTranslation#KExq YXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="RightsExpressionTranslator"> <wsdl:port name="RightsExpressionTranslator" binding="tns:RightsExpressionTranslator"> <soap:address location="http://EFPC016:6060/RightsExpressionTranslator"/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre> |
| Method | generateTranslation |
| Description | This function is used to return a destinationLicense. The system makes a translation from a sourceLicense to a destinationLicense. The destinationLicense, if the translation is possible, will be written in a languageDestination. |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:sourceLicense i:type="d:string">Source License</wn0:sourceLicense> <wn0:destinationLanguage i:type="d:string">AXMEDIS-REL</wn0:destinationLanguage> </e:Body> </e:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:wn0="http://systinet.com/xsd/SchemaTypes"/> <e:Body> <wn0:string_Response i:type="d:string">destinationLicense</wn0:string_Response> </e:Body> </e:Envelope> </pre> |
| Input parameters | xsd:String sourceLicense xsd:String languageDestination |
| Output parameters | xsd:String destinationLicense -1 means destinationLanguage doesn't exist -2 translation not possible |

| | |
|--|---|
| | -3 sourceLicense error -4 error generation destination license -5 unknown error |
|--|---|

5 Protection Tool Engine (FHGIGD)

Protection Tool Engine

From the WF is possible to control both the Editor and the Engine.

This allows to pass from Formatting to Protection according to some defined flow. This tool has to:

- protect AXMEDIS objects in large set, adapting the DRM, PAR, Licenses, etc.
- merge a license and an object to create a Governed AXMEDIS Objects
- generate licenses in automatic according to some rules
- etc.

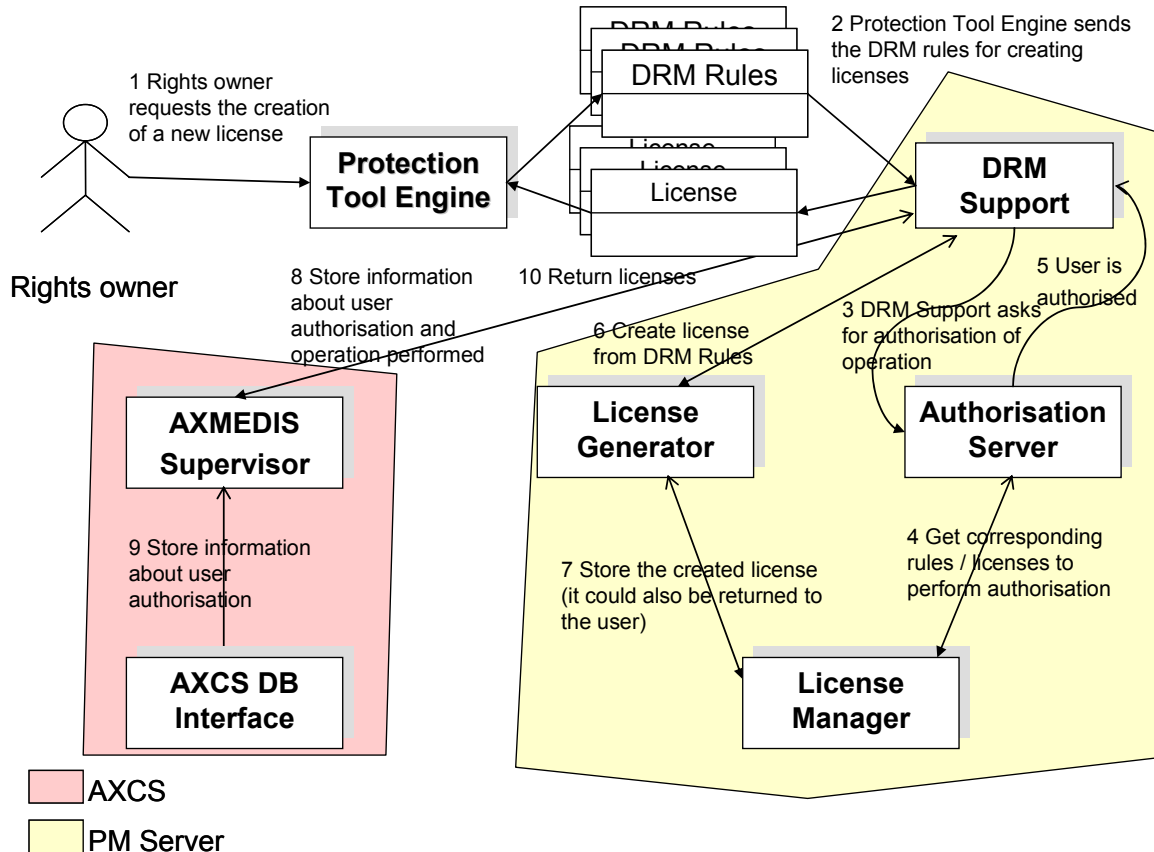
In the January 2005 it has been decided:

- The Protection Tool Engine is an instance of the Comp/Form Engine with specific support to work with protection functionalities
- The Protection Tool Editor is an instance of the Comp/Form Engine with specific support to work with protection functionalities
- The functionalities that have to be available from the Script and from the Editor are mainly:
 - Verification of PAR, internal and external against the License
 - Generation of the License from PAR (internal and or external)
 - Verification of any PAR or License against some RIGHTS written in clear such as: the play on the AXOID 34 in July 2005 for 5 times, the print of AXOID 56 in Spain in May 2006 at least one, etc.
 - Addition/remove of rights from a given License/PAR
 - Adaptation of a license/PAR removing some rights, scaling down...
 - Production of the Protection Information
 - Writing the Protection Information sequence of commands to protect the object (see IPMP standard of MPEG21, plus start-end segment, etc...)
 - Save the Protection Information sequence to the database
 - Save the License into the database
 - Send the License to the PMS
 - Send the Protection Information to the AXCS via the PMS
 - Load from the database of a License Model via the License Model ID
 - Code from 0 the rights and other fields that compose a license
 - Exploitation of libraries for encryption/decrypt, scramble/descramble, compress/uncompress, etc., as native functionalities into the script and for defining the Protection Information sequence of commands
 - Exploitation of lib and DRM support for key generation for Enc/dec Algorithms, couples of keys, etc., different size, etc...
- The functionalities in terms of Editing and Viewing are divided and provided in two tools:
 - DRM Editor and Viewer of FUPF
 - Protection Information Editor and Viewer of FHGIGD
- The functionalities in terms of Engine Processing are provided by adding two modules at the Compositional and Formatting Engine called:
 - JS DRM Support: FHGIGD
 - JS Protection Information: FHGIGD
 -

The description in this section is based on and refers to the description of the composition and formatting engine, rules and rule editor in chapter 2 “AXMEDIS Content Production Area” of Part C of the AXMEDIS framework specification.

In some section where the concepts of the protection and the production tools are the very similar, only a summary of the functionalities that are described and specified in Part C is given. In other sections that are more focused on the specific requirements and properties of the production tool area a detailed description and specification is given. In the latter case the focus is on the modifications and the additional features needed for the protection tool area.

5.1 Scenarios on Content Protection Area



- 1) The Protection Tool Engine is initiated to protect a single object or multiple objects by:
 - a) WF manager
 - b) Protection Information Editor and Viewer
 - c) Automatically via the activated Protection Rules (generated by the Protection Rules Editor)
- 2) Parameters are verified and the protection rule is loaded. The protection rule is expressed in a script language. Its format is described in DE3-1-2C-AXFW (AXMEDIS framework Part C – Content Production)
- 3) If not AXMEDIS objects but a selection of AXMEDIS content is passed in the rule the Query Support retrieves the specific AXOID(s).
- 4) The AXMEDIS objects are accessed via the AXOM according to the given rule. Some or all of the following operations are performed according to the rule:
 - a) License creation and verification:

- i) DRM rules are the basis for the license creation and forwarded to the DRM Support (License Generator)
 - ii) The PARs are estimated on the basis of the licenses of the included resources and then the PAR is included in the AXInfo contained into the protected and non protected parts of the object.
 - iii) PMS Domain factory creates/adapts the license from the rules or the user input.
 - iv) Verification of PAR or License against given rights
- b) PMS Domain factory creates required keys (e.g. for encryption or hash functions).
 - c) Creation of the protection information
 - d) Protection the object (resulting in a new object or a new version of the object). Encryption support (see use case Encryption) is used via the AXOM.
 - e) If the protection is successful and the protection information has been generated to protect this object, the protection information has to be stored (see use case Storage of security information)
 - f) The license and the Protection information are sent to the PMS. The PMS forwards the Protection Information to the AXCS.

5.2 Protection Rule Editor (FHGIGD)

The Protection Rule Editor is an instance of the AXCP rule editor. For a detailed description of the general rule format and Editor GUI please see section 2.7 “AXCP Rule Editor” of Part C. In the following only the protection specific aspects are described.

Protection rules can be used to specify a number of protection operations that are applied to a selection of AXMEDIS objects. This includes the protection of AXMEDIS objects (or parts) using different algorithms, e.g. encryption and fingerprinting and the creation and modification of licenses using the PMS.

Different concepts have to be distinguished within AXMEDIS regarding DRM. These concepts address usage restrictions that include legal and technical aspects. The DRM rules specify rules for the usage of objects. A license states specific rights and restrictions that are granted to principals. A license describes technical access and usage conditions. PAR (Potentially Available Rights) describe the rights which are generally available for an AXMEDIS object and which may be assigned to a principal with a license.

In AXMEDIS, the license concept is derived from the REL definitions in MPEG-21 [MPEG-21]:

Principal:

A principal encapsulates the identification of principals to whom rights are granted. Each principal identifies exactly one party. In contrast, a set of principals, such as the universe of everyone, is not a principal.

A principal denotes the party that it identifies by information unique to that individual. Usefully, this is information that has some associated authentication mechanism by which the principal can prove its identity.

The Principal type supports the following identification technologies:

- A principal that must present multiple credentials, all of which must be simultaneously valid, to be authenticated.
- A keyHolder, meaning someone identified as possessing a secret key such as the private key of a public / private key pair.
- Other identification technologies that may be invented by others.

Right:

A right is the "verb" that a principal can be granted to exercise against some resource under some condition. Typically, a right specifies an action (or activity) or a class of actions that a principal may perform on or using the associated resource.

MPEG REL provides a right element to encapsulate information about rights and provides a set of commonly used, specific rights, notably rights relating to other rights, such as issue, revoke and obtain.

Extensions to MPEG REL could define rights appropriate to using specific types of resource. For instance, the MPEG REL content extension defines rights appropriate to using digital works (e.g., play and print).

Resource:

A resource is the "object" to which a principal can be granted a right. A resource can be a digital work (such as an e-book, an audio or video file, or an image), a service (such as an email service, or B2B transaction service), or even a piece of information that can be owned by a principal (such as a name or an email address).

MPEG REL provides mechanisms to encapsulate the information necessary to identify and use a particular resource or resources that match a certain pattern. The latter allows identification of a collection of resources with some common characteristics. Extensions to MPEG REL could define resources appropriate to specific business models and technical applications.

Condition:

A condition specifies the terms, conditions and obligations under which rights can be exercised. A simple condition is a time interval within which a right can be exercised. A slightly complicated condition is to require the existence of a valid, prerequisite right that has been issued to some principal. Using this mechanism, the eligibility to exercise one right can become dependent on the eligibility to exercise other rights.

MPEG REL defines a condition element to encapsulate information about conditions and some very basic conditions. Extensions to MPEG REL could define conditions appropriate to specific distribution and usage models. For instance, the MPEG REL content extension defines conditions appropriate to using digital works (e.g., watermark, destination, and renderer).

AXMEDIS licenses

Any operation on an AXMEDIS object must be granted by a corresponding license. This license should be a digital AXMEDIS license. However, in real world scenarios digital AXMEDIS licenses are not always available for all digital objects. AXMEDIS licenses include specific rules for operations on objects, a list of operations and their corresponding execution conditions for specific users. The License Manager stores created licenses in the License Database. AXMEDIS licenses are optionally store in Governed AXMEDIS objects. PAR (internal and external) are stored in the AXInfo.

AXMEDIS PAR (Potentially Available Rights)

PAR (Potentially Available Rights) describe the rights which are generally available for an AXMEDIS object and which may be assigned to a principal with a license. Within AXMEDIS PAR are described using the same language as for licenses.

AXMEDIS Protection Rule Editing and Execution

With the AXCP Rule Editor a user can generate, view and edit protection rules that can be activated and executed by the AXCP Engine. AXCP Rules are described in XML and include a script that will be executed when the rule is processed. The script code is written in JavaScript

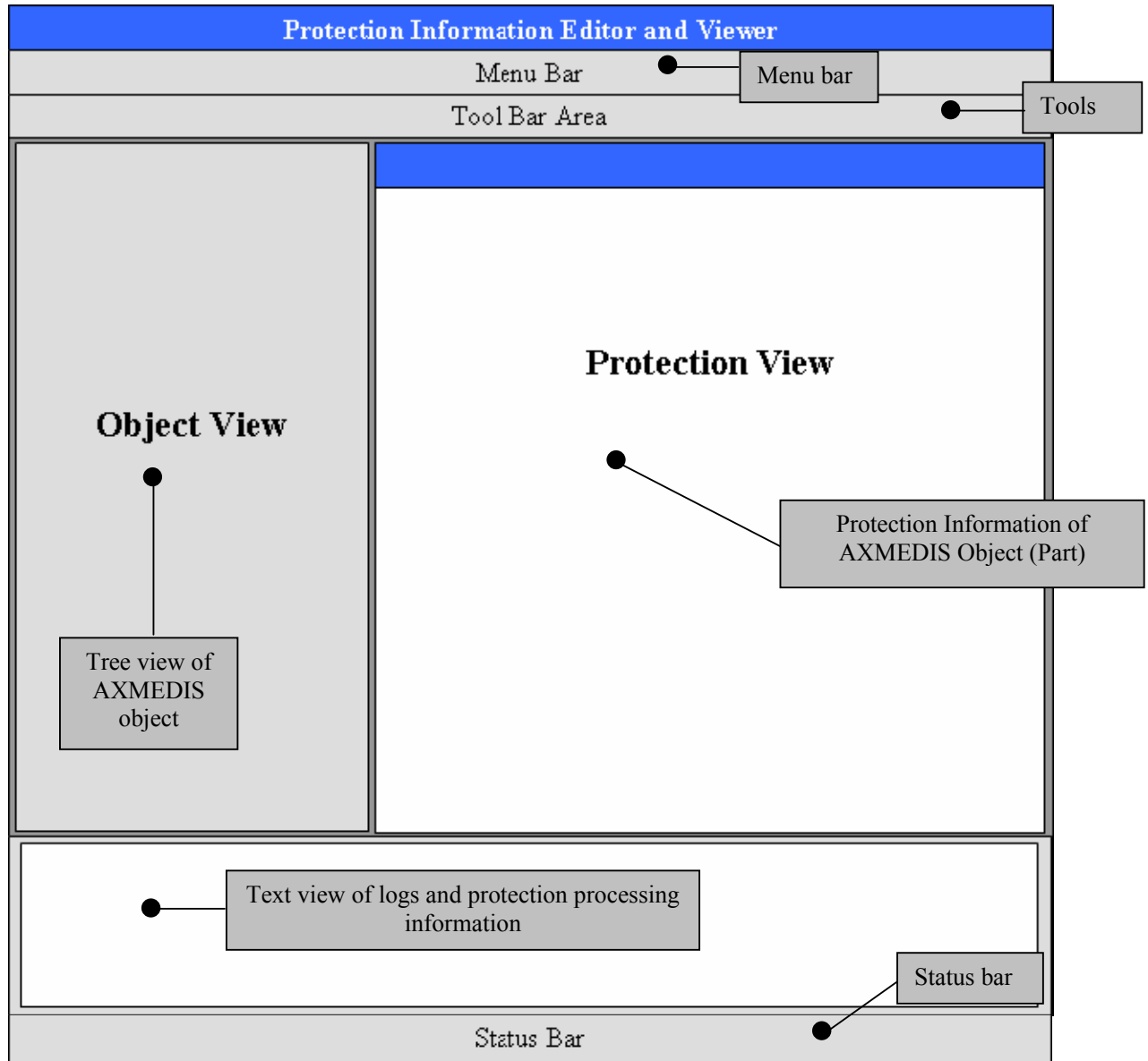
Please see section 3.9 "JS_Protection" and 3.10 "JS_DRM" for a detailed description of the JavaScript objects that are available for the specification of protection rules and the functionalities that are provided.

The functionalities include:

- Applying Protection to AXMEDIS object : encryption, scrambling, compression, FP, ... and creation of new Protection Information
- Sending the Protection Information (keys and parameter, see IPMP standard of MPEG21) to the database of the AXCS via the PMS
- Creating a new governed object (with license)
- Generating a license from license model and additional information (principal, AXOID)

- Check/Verification of an issued License or an existing PAR against some RIGHTS written in clear such as: “the play on the AXOID 34 in July 2005 for 5 times, the print of AXOID 56 in Spain in May 2006 at least one, etc.”
- Check/Verification if it is possible to issue/generate a License with some RIGHTS written in clear such as: “the play on the AXOID 34 in July 2005 for 5 times, the print of AXOID 56 in Spain in May 2006 at least one, etc.”
- Addition of rights or removal from a PAR/license (PAR/license adaptation): Generation of a new license (with new or less rights) AND Revocation of the old licenses in ONE TRANSACION

5.3 Protection Information Editor and Viewer (FHGIGD)



Menu Bar

The menu bar will be constituted of the following entries:

File

- Load – load an AXMEDIS object
- Save – save the current AXMEDIS object
- Delete – delete the selected AXMEDIS object
- Close – Close the current AXMEDIS object
- Exit – Quit the editor

Protection Commands

- Encryption – Encrypt the selected part of the AXMEDIS object
- Scrambling –Scramble the selected part of the AXMEDIS object
- Compression – Compress the selected part of the AXMEDIS object
- Fingerprinting – Create fingerprint the selected part of the AXMEDIS object
- Save Protection Config – Save the present configurations, e.g. encryption algorithm and parameters, to a file
- Load Protection Config – Load a configuration, e.g. encryption algorithm and parameters, from a file

Help

- Help – Call the on line help
- About – Information about the authors, version etc.

Object view area

In this area the structure and the components of the AXMEDIS object are displayed. It will be visualised using a Tree control that will permit to show and browse components according the MPEG-21 XML schema.

Text View Area

This is a text control where protection processing information, log messages, textual description, alert, etc... are displayed.

Protection Information View Area

In this area the contents of the selected structure of the rule is displayed. It will be visualised using a List control that will permit to show and browse components according the rule XML schema.

| Module Profile | | |
|--|---------------------------------|---|
| Protection Information Editor and Viewer | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Single Threaded | |
| Language of Development | C++ | |
| Responsible Name | Peter Ebinger | |
| Responsible Partner | FHGIGD | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | Microsoft Windows, Linux | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXOM | | |
| PMS | | |
| | | |
| File Formats Used | Shared with | File format name or reference to a |

| | | section |
|----------------|-----------------------------------|--|
| | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| GUI | | |
| | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| wxWidgets | | |
| | | |
| | | |

6 AXMEDIS Accounting Area (EXITECH)

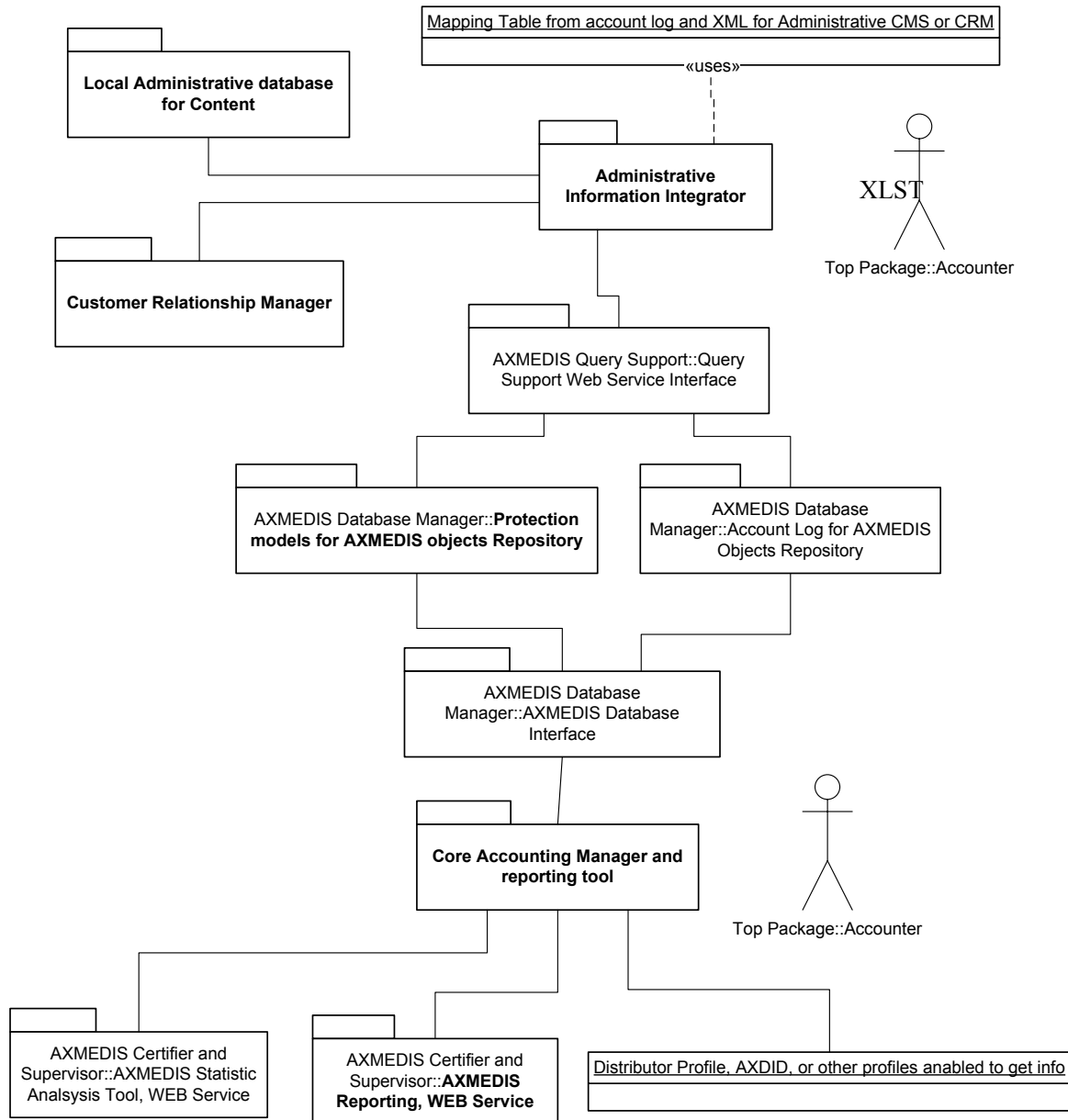
| Module Profile | | |
|---|---|--|
| Core Accounting Manager and Reporting Tools | | |
| Executable or Library(Support) | Executable (Web Service) | |
| Single Thread or Multithread | Multithread | |
| Language of Development | JAVA and PHP for CAMART web Interface | |
| Responsible Name | Fioravanti | |
| Responsible Partner | EXITECH | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | All | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| AXMEDIS Database Interface | | |
| Formal model for Licenses as DRM rules | | |
| AXMEDIS Reporting, WEB Service | | |
| Statistic Analysis Tool, WEB Service | | |
| File Formats Used | Shared with | File format name or reference to a section |
| XML, SOAP, WSDL | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| None | Developed as a Web service for the part to the end user query, | |
| | User interface, if necessary, will be developed in web oriented way | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |
| | | |
| | | |

AXMEDIS accounting area mainly includes:

- Core Accounting Manager and reporting tool, the so called CAMART
- Administrative Information Integrator

These tools are the milestone for extracting useful information from Action-Logs and translating them in the business environment for allowing immediate query (CAMART) for example for establishing program or for marketing purposes. On the other hand Administrative Information Integrator will be adopted to push in the user CMS and CRM business information that can be mapped from Action-Logs to the user system.

AXMEDIS Accounting Area



Core accounting manager and reporting tool (CAMART) has two main aspects: the interface toward the system (AXCS and AXDB) and the direct interface toward the user. Both aspects are discussed in the following:

6.1 Core Accounting Manager and Reporting Tool (EXITECH)

6.1.1 Core Accounting Manager and Reporting Tools interface toward the system

The role of Core Accounting manager and Reporting Tool (CAMART) is strictly bounded with database for logs since Core Accounting Manager and reporting tool has to collect:

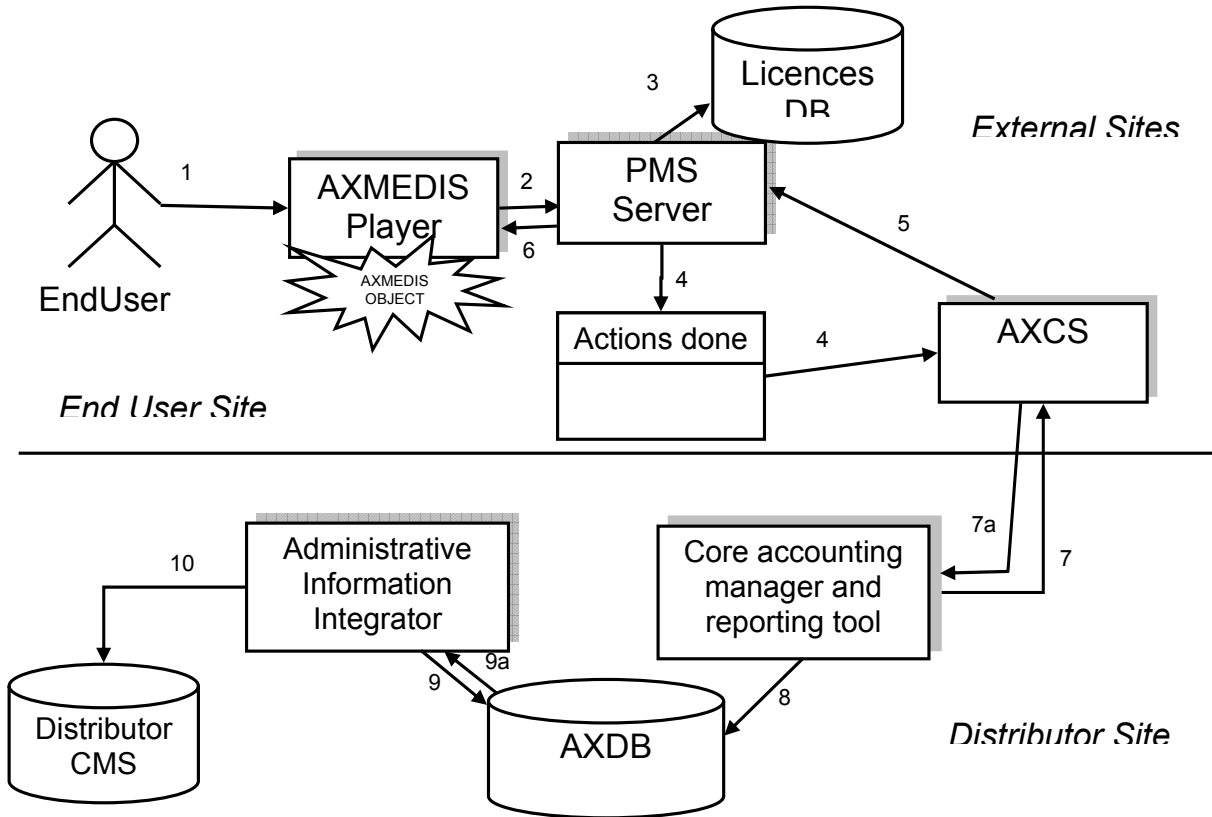
- information from AXMEDIS certifier and supervisor about Action Log and to store such info in the AXMEDIS database mapping fragmented information that can be sent by the different tool to the AXCS in a consistent manner in the database: this is very important since the AXCS or the SuperAXCS do not store forever info related to each user, but with some policies the detailed data will be removed after a certain amount of time or after some triggered operations to be defined. Only data for statistical purposes will be kept.
- Information regarding the B2B information. For example to have a table of Business counterparts that distribute, integrate, etc. AXMEDIS Objects. They should have a contract with the AXMEDIS Factory that host the AXMEDIS database (AXDB). This information is useful to interpreting the Action Log Information coming from the AXCS databases, when available in the AXCS.

The collection of information will be made on a timed base and CAMART will act a a client of the AXCS Reporting Web Service.

AXMEDIS system has to be scalable and therefore we will have to deal with the problem that some installation can have AXDB, AXCS and other supporting tools on different machines, while other installation can be less distributed due to a minor need for speed or storage.

The core accounting manager is a sort of Client side of the bridge between the AXDB and the AXCS databases in order to allow AXCS to be independent by the database. The server side in the AXCS is the Web Service: AXMEDIS Reporting WEB Service. The CAMART can be interpreted as a part of the AXMEDIS Database Interface, since is the part of the system that allows to write data related to Action-Logs on the DB.

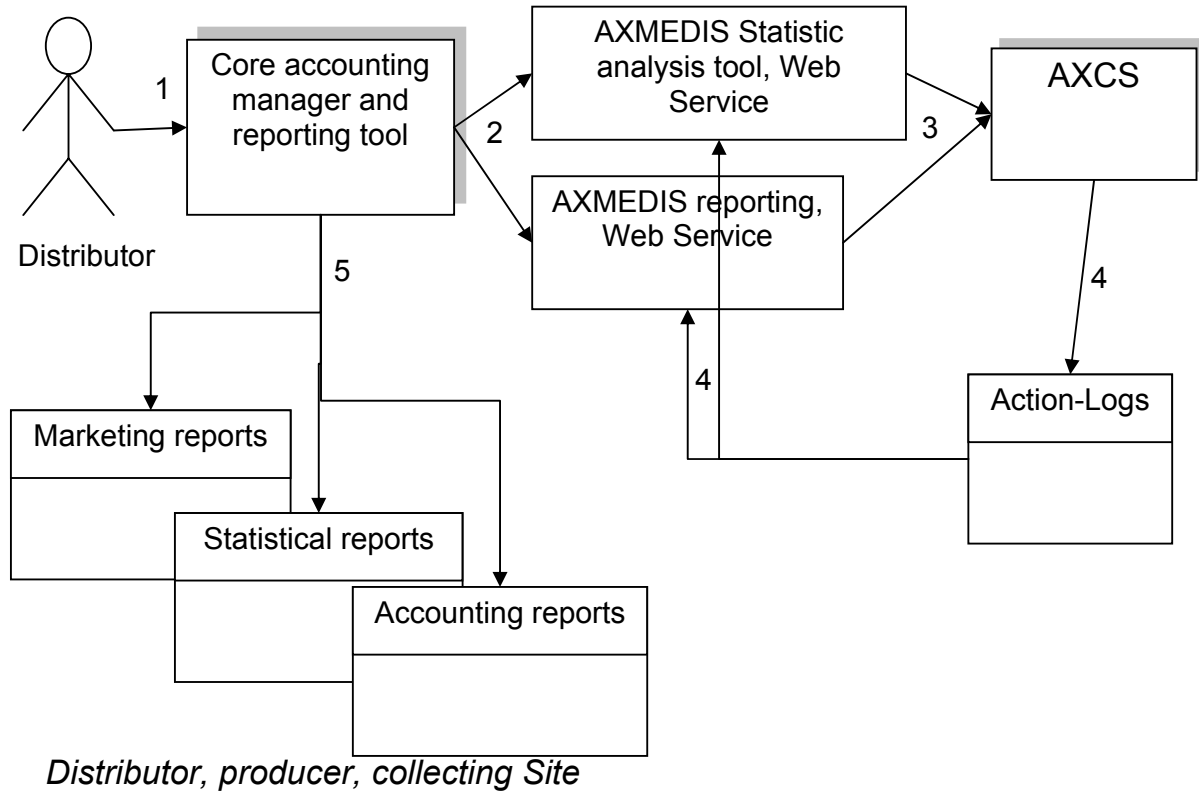
This aspect is clarified by the following scenario:



1. End User requests to perform an action on an AXMEDIS Protected Object
2. AXMEDIS Player asks PMS to perform an Action (assuming client has been already certified)
3. PMS checks in the Licence DB if the Action is allowed (assuming OK)
4. PMS sends AXCS the action performed
5. AXCS gives back the key to access the content (if necessary)
6. PMS gives the grant to access the content and possibly the key to the AXMEDIS Player
7. Accounting & Rep. Tool retrieves from AXCS the actions performed by all the End Users on objects distributed by the distributor
- 7a. Accounting & Rep. Tool sends the key to AXCS
8. A&R Tool stores the transactions in the AXDB
9. Adm. Integrator gets transactions performed from the DB
- 9a. Adm. Integrator sends the key to AXDB
10. Administrative information are mapped into the Distributor CMS

6.1.2 Core Accounting Manager and Reporting Tools interface toward the user

For detailing a draft specification for this second aspect of this tool, it is necessary to start from some practical scenarios that show interaction between the user and the tool with the involvement of other AXMEDIS parts.

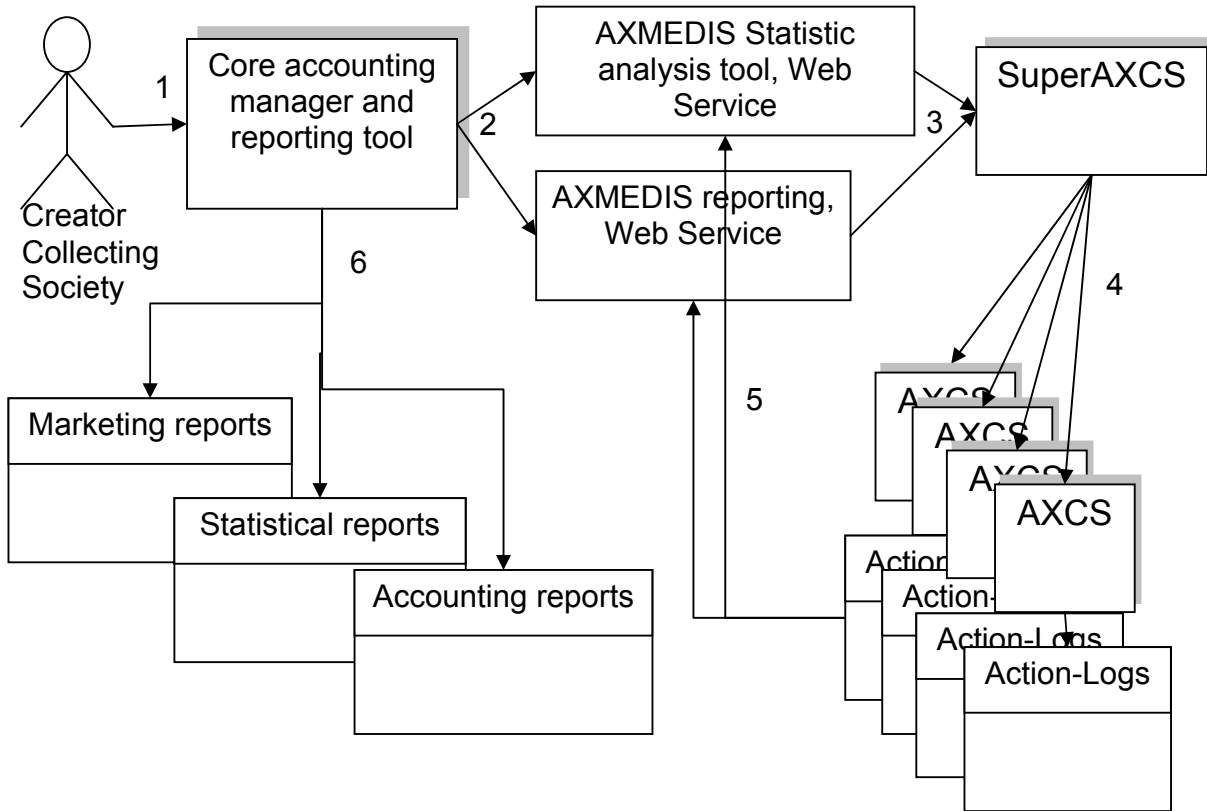


1. A Distributor wants to recover information on actions performed on the objects he has rights.
2. Core accounting manager and reporting tool query the correct tool for obtaining the Action-Logs in the correct form (anonymous or not, aggregated or not, etc)
3. AXMEDIS Statistic or reporting tools query AXCS
4. AXCS extracts the required Action-Logs and communicate them to the tools that perform actions to return results in the desired form
5. Different reports are generated on the basis of the information collected.

The actor working on the CAMART user interface can be any administrative and management user that has interest in making queries and browsing the information related to the usage of AXMEDIS objects. For example:

- A Distributor could be interested in seeing the list of Action Logs related to a given second distributor, integrator, etc.
- An Integrator could be interested in seeing the list of Action Logs related to a given AXMEDIS Object, etc.
- A Distributor could be interested in seeing the list of second Distributors that have exploited some specific AXMEDIS object, etc.
- A Distributor could be interested to see how many transactions have been registered on its AXMEDIS objects in the last two months.

These examples can be used by the Actor (account manager) to extract the information and move it into the Administrative Database of the CMS by means of the AXMEDIS tool called Administrative Information Integrator.



1. An Actor, that is collecting society or creator, wants to recover information on actions performed on the objects he has rights.
2. Core accounting manager and reporting tool query the correct tool for obtaining the Action-Logs in the correct form (anonymous or not, aggregated or not, etc)
3. AXMEDIS Statistic or reporting tools query the SuperAXCS
4. SuperAXCS recover information from the different AXCSs
5. The different AXCSs extract the required Action-Logs and communicate them to the tools that perform actions to return results in the desired form
6. Different reports are generated on the basis of the information collected.

Statistics information recovered by the AXCS are only general information and trends. This means that are not contextualised in terms of specific Object ID or Distributor ID, etc.

The general role of CAMART with respect to the end user that uses it, is to provide a web interface for making reporting and statistical queries directly on the system.

We have to consider in a different way Reporting and statistical queries since:

- Reporting queries are always executed on the AXMEDIS Database Interface for collecting information that are contained in the local database that in turn had been gathered from the AXCS. In this way the actor sees only the data for which is authorized;
- Statistical queries will always be resolved by the SuperAXCS by the means of Statistic Analysis Tool, Web Service. CAMART will act as the client side of the Web Service and will display results to the user in the web interface.

The user interface, if needed, will be web base in order to cover the needs of interoperability, usability and maintainability that are expected from AXMEDIS in the large sense.

6.1.3 CAMART interface with AXCS

CAMART needs to get fresh logs from AXCS and therefore it is a client of the webservice interface that AXCS will expose to have access to log information.

In this section a sample interface will be sketched in order to specify in terms of WSDL this communication protocol that will be a synchronous polling from CAMART to AXCS.

The format for the data transmitted will not be complete but will reflect in its final version the ER model provided for AXCS Logs.

AXCS will have to implement this WebService and CAMART will be a client for it, reading with predefined time interval the logs from AXCS and storing them in the local AXDB related to Logs.

This process is completely automatic and is managed by a CAMART daemon that will periodically extract from AXCS all the logs for which the user is entitled and store them in the AXDB.

| AXCS Log | |
|----------|---|
| Method | Description |
| GetLog | This method, exposed by AXCS, will be invoked by CAMART to get all the logs that have a registration timestamp after a certain timestamp given as input |

| AXCS Log | |
|----------|---|
| WSDL | <pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="AXCS_log" targetNamespace="http://www.axmedis.org/axcs_log.wsdl" xmlns:tns="http://www.axmedis.org/axcs_log.wsdl" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax" xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:MIME="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:DIME="http://schemas.xmlsoap.org/ws/2002/04/dime/wsdl/" xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"> <types> <schema targetNamespace="urn:ax" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="LogDetail"> <sequence> <element name="LogId" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="AXOID" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="AXUID" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="AXDID" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="AXCID" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="AXWID" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="RegistrationTimestamp" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="ExecutionTimestamp" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="Detail1" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> </pre> |

| | |
|--|--|
| | <pre> <element name="Detail2" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="Detail3" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="Detail4" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> </sequence> </complexType> <complexType name="LL"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="NumberOfLogs" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="Log" type="ax:LogDetail" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType> <!-- operation request element --> <element name="user" type="xsd:string"/> <!-- operation request element --> <element name="pwd" type="xsd:string"/> <!-- operation request element --> <element name="registrationTimestamp" type="xsd:string"/> <!-- operation response element --> <element name="logs" type="ax:LL"/> </schema> </types> <message name="getLogRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> <part name="registrationTimestamp" element="ax:registrationTimestamp"/> </message> <message name="ResultLog"> <part name="logs" element="ax:logs"/> </message> <portType name="AXCS_logPortType"> <operation name="getLog"> <documentation>Service definition of function ax__getLog</documentation> <input message="tns:getLogRequest"/> <output message="tns:ResultLog"/> </operation> </portType> <binding name="AXCS_log" type="tns:AXCS_logPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="getLog"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> </binding> <service name="AXCS_log"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="AXCS_log" binding="tns:AXCS_log"> <SOAP:address location="http://www.axmedis.org/axcs_log.cgi"/> </pre> |
|--|--|

| | |
|-------------------------|---|
| | <pre> </port> </service> </definitions> </pre> |
| Method | getLog |
| Description | This method, exposed by AXCS, will be invoked by CAMART to get all the logs that have a registration timestamp after a certain timestamp given as input |
| Input parameters | <p>xsd:string user: user name for authentication and authorization in AXCS</p> <p>xsd:string pwd: password for authentication and authorization in AXCS</p> <p>xsd:string timestamp: timestamp after which logs have to be extracted</p> |
| Output parameters | <p>ax:ResultLog, that is a complex type formed by:</p> <ul style="list-style-type: none"> xsd:int RetCode: return code for the operation (0 means OK) xsd:int NumberOfLogs: number of Logs returned sequence of Log that is a complex type: <ul style="list-style-type: none"> xsd:string AXOID: AXOID to which the Log refers xsd:string AXUID: User that performed the action xsd:string AXDID: ID of the distributor (optional) xsd:string AXCID: ID of the Creator (optional) xsd:string AXWID: ID of Work xsd:string RegistrationTimestamp: timestamp of event registration xsd:string ExecutionTimestamp: timestamp of event execution xsd:string Detail1: detail 1 of the operation xsd:string Detail2: detail 2 of the operation xsd:string Detail3: detail 3 of the operation xsd:string Detail4: detail 4 of the operation |
| Request Sample Message | <pre> <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:getLog> <user>myuser</user> <pwd>mypwd</pwd> <registrationTimestamp>20050225T10:05:45</registrationTimestamp> </ax:getLog> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
| Response Sample Message | <pre> <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:ResultLog> <logs> <RetCode>0</RetCode> <NumberOfLogs>2</NumberOfLogs> <Log> <LogId>1</LogId> <AXOID>45243fcr3</AXOID> <AXUID>3f34fecqr3</AXUID> <RegistrationTimestamp>20050221T4:09:23</RegistrationTimestamp> </pre> |

| | |
|--|--|
| | <pre> <ExecutionTimestamp>20050220T4:09:23</ExecutionTimestamp> </Log> <Log> <LogId>2</LogId> <AXOID>3255g24vf</AXOID> <AXUID>542tg43vfe4t</AXUID> <AXDID>5g2552gf43</AXDID> <RegistrationTimestamp>20050221T4:09:50</RegistrationTimestamp> <ExecutionTimestamp>20050221T4:09:45</ExecutionTimestamp> <Detail1>my details</Detail1> </Log> </logs> </ax:ResultLog> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
|--|--|

6.1.4 CAMART as a public service

As described in the scenario CAMART will have also a public part to allow external module to ask for Log according to predefined criteria. The criteria identified in this phase of the specification are:

- Get all the log
- Get all the log after a certain registration/execution timestamp
- Get all the log after a certain registration/execution timestamp that are related to:
 - an user
 - an object
 - a creator
 - a distributor
 - a worked
 - a combination in AND of the previous

The first cases are particular instances of the most general one, considering that if a field is left blank it is not considered in the query

Once the records have been extracted they can be processed by a statistical/administrative tool for obtaining the necessary business information.

The CAMART will have a public web service with the following characteristics:

| CAMART_Log | |
|------------|---|
| Method | Description |
| extractLog | This method, extracts from the CAMART the logs that are fresher with respect to a certain timestamp and that satisfies some query criteria in AND over AXOID, AXUID, AXCID, AXDID and AXWID |

| CAMART_Log | |
|------------|---|
| WSDL | <pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="CAMART_log" targetNamespace="http://www.axmedis.org/camart_log.wsdl" xmlns:tns="http://www.axmedis.org/camart_log.wsdl" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax" xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/" </pre> |

| | |
|--|--|
| | <pre> xmlns:MIME="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:DIME="http://schemas.xmlsoap.org/ws/2002/04/dime/wsdl/" xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"> <types> <schema targetNamespace="urn:ax" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/"> <complexType name="LogDetail"> <sequence> <element name="LogId" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="AXOID" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="AXUID" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="AXDID" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="AXCID" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="AXWID" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="RegistrationTimestamp" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="ExecutionTimestamp" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="Detail1" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="Detail2" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="Detail3" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="Detail4" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> </sequence> </complexType> <complexType name="LL"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="NumberOfLogs" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="Log" type="ax:LogDetail" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType> <!-- operation request element --> <element name="user" type="xsd:string"/> <!-- operation request element --> <element name="pwd" type="xsd:string"/> <!-- operation request element --> <element name="timestamp" type="xsd:string"/> <!-- operation request element --> <element name="registration" type="xsd:boolean"/> <!-- operation request element --> <element name="AXOID" type="xsd:string"/> <!-- operation request element --> <element name="AXUID" type="xsd:string"/> <!-- operation request element --> <element name="AXDID" type="xsd:string"/> <!-- operation request element --> <element name="AXCID" type="xsd:string"/> <!-- operation request element --> <element name="AXWID" type="xsd:string"/> <!-- operation response element --> </pre> |
|--|--|

| | |
|------------------|---|
| | <pre> <element name="logs" type="ax:LL"/> </schema> </types> <message name="extractLogRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> <part name="timestamp" element="ax:timestamp"/> <part name="registration" element="ax:registration"/> <part name="AXOID" element="ax:AXOID"/> <part name="AXUID" element="ax:AXUID"/> <part name="AXDID" element="ax:AXDID"/> <part name="AXCID" element="ax:AXCID"/> <part name="AXWID" element="ax:AXWID"/> </message> <message name="ResultLog"> <part name="logs" element="ax:logs"/> </message> <portType name="CAMART_logPortType"> <operation name="extractLog"> <documentation>Service definition of function ax__extractLog</documentation> <input message="tns:extractLogRequest"/> <output message="tns:ResultLog"/> </operation> </portType> <binding name="CAMART_log" type="tns:CAMART_logPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="extractLog"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> </binding> <service name="CAMART_log"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="CAMART_log" binding="tns:CAMART_log"> <SOAP:address location="http://www.axmedis.org/camart_log.cgi"/> </port> </service> </definitions> </pre> |
| Method | extractLog |
| Description | Extract the Log from CAMART fresher with respect to a given timestamp |
| Input parameters | <p>xsd:string user: user name for authentication and authorization in AXCS</p> <p>xsd:string pwd: password for authentication and authorization in AXCS</p> <p>xsd:string timestamp: timestamp after which logs have to be extracted</p> <p>xsd:boolean registration: true if the previous timestamp is related to the registration, false if it is related to the execution,</p> |

| | |
|-------------------------|--|
| | <p>xsd:string AXOID: limit to a certain AXOID</p> <p>xsd:string AXUID: limit to a certain AXUID</p> <p>xsd:string AXDID: limit to a certain AXDID</p> <p>xsd:string AXCID: limit to a certain AXCID</p> <p>xsd:string AXWID: limit to a certain AXWID</p> |
| Output parameters | <p>ax:ResultLog, that is a complex type formed by:</p> <p> xsd:int RetCode: return code for the operation (0 means OK)</p> <p> xsd:int NumberOfLogs: number of Logs returned</p> <p> sequence of Log that is a complex type:</p> <p> xsd:string AXOID: AXOID to which the Log refers</p> <p> xsd:string AXUID: User that performed the action</p> <p> xsd:string AXDID: ID of the distributor (optional)</p> <p> xsd:string AXCID: ID of the Creator (optional)</p> <p> xsd:string AXWID: ID of Work</p> <p> xsd:string RegistrationTimestamp: timestamp of event registration</p> <p> xsd:string ExecutionTimestamp: timestamp of event execution</p> <p> xsd:string Detail1: detail 1 of the operation</p> <p> xsd:string Detail2: detail 2 of the operation</p> <p> xsd:string Detail3: detail 3 of the operation</p> <p> xsd:string Detail4: detail 4 of the operation</p> |
| Request Sample Message | <pre><?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:extractLog> <user>myuser</user> <pwd>mypassword</pwd> <timestamp>20050220T5:00:00</timestamp> <registration>false</registration> <AXOID>4352rfr434</AXOID> <AXUID></AXUID> <AXDID></AXDID> <AXCID></AXCID> <AXWID></AXWID> </ax:extractLog> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre> |
| Response Sample Message | <pre><?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:ResultLog> <logs> <RetCode>0</RetCode> <NumberOfLogs>2</NumberOfLogs> <Log> <LogId>1</LogId> <AXOID>45243fcr3</AXOID> <AXUID>3f34fecqr3</AXUID></pre> |

| | |
|--|--|
| | <pre> <RegistrationTimestamp>20050221T4:09:23</RegistrationTimestamp> <ExecutionTimestamp>20050220T4:09:23</ExecutionTimestamp> <Log> <Log> <LogId>2</LogId> <AXOID>3255g24vf</AXOID> <AXUID>542tg43vfe4t</AXUID> <AXDID>5g2552gf43</AXDID> <RegistrationTimestamp>20050221T4:09:50</RegistrationTimestamp> <ExecutionTimestamp>20050221T4:09:45</ExecutionTimestamp> <Detail1>my details</Detail1> </Log> </logs> </ax:ResultLog> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre> |
|--|--|

6.1.5 CAMART user interface

CAMART is a service and not a tool with an user interface, and therefore the user interface will be a simple web interface that will interact with the webservice specified above.

A raw sketch of the user interface can be found in the following.

The screenshot shows a web browser window titled "Your Title Here - Mozilla Firefox". The address bar shows a file path: "file:///D:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/Current-PART-E/CAMART/alley77564536.html". The browser's toolbar includes various icons and a search bar. Below the browser window, the "CAMART Log Extractor" interface is displayed. It features a form with the following elements:

- User** and **Password** input fields.
- Timestamp** input field, followed by radio buttons for **Registration** and **Execution**.
- AXOID**, **AXUID**, **AXCID**, **AXDID**, and **AXWID** input fields.
- Extract** and **Clear** buttons at the bottom.

This interface allow to extract toward tools that are not the CMS (for which the operation is performed by Administrative Information Integrator) for statistical or reporting purposes. This operation is not bouded to that of log extraction from AXCS that is performed always in an automatic manner. The data that are extracted here are already present in the AXDB and no communication will be established with AXCS.

In any case, the actions logs extracted are those related to:

- Distributor: for which AXDID is implicit, optionally asking for logs related to a certain combination of AXCID, AXOID, AXUID, AXWID
- Creator: for which AXCID is implicit, asking for logs related to a certain combination of AXDID, AXOID, AXUID, AXWID
- Collecting societies: for which Nationality is implicit with user name, asking for Logs related to a certain combination of AXWID, AXDID, AXCID, AXOID, AXUID

In a second phase we can have a tool to help Distributor, Creator, Collecting to identify the IDs of the tools from the names, the titles of getting them from the result of a query performed on QS.

The information provided/returned to the Distributor, Creator, Collecting, are different as evidenced by the optional fields that can be present or not in the result.

6.2 Administrative Information Integrator (WP9.1: EXITECH)

| Module Profile | | |
|---|-----------------------------------|--|
| Administrative Information Integrator | | |
| Executable or Library(Support) | Executable | |
| Single Thread or Multithread | Multithread | |
| Language of Development | PHP | |
| Responsible Name | Fioravanti | |
| Responsible Partner | EXITECH | |
| Status (proposed/approved) | Proposed | |
| Platforms supported | All | |
| | | |
| Interfaces with other tools: | Name of the communicating tools | Communication model and format (protected or not, etc.) |
| Local Administrative database for Content | | |
| Customer Relationship Manager | | |
| Formal model for Licenses as DRM rules | | |
| Query Support Web Service Interface | | |
| File Formats Used | Shared with | File format name or reference to a section |
| XML, XSL | | |
| | | |
| | | |
| User Interface | Development model, language, etc. | Library used for the development, platform, etc. |
| Simple Web interface for polling | Web Service client of CAMART | |
| No Interface for pushing | | |
| | | |
| Used Libraries | Name of the library and version | License status: GPL. LGPL. PEK, proprietary, authorized or not |
| | | |

| | | |
|--|--|--|
| | | |
| | | |

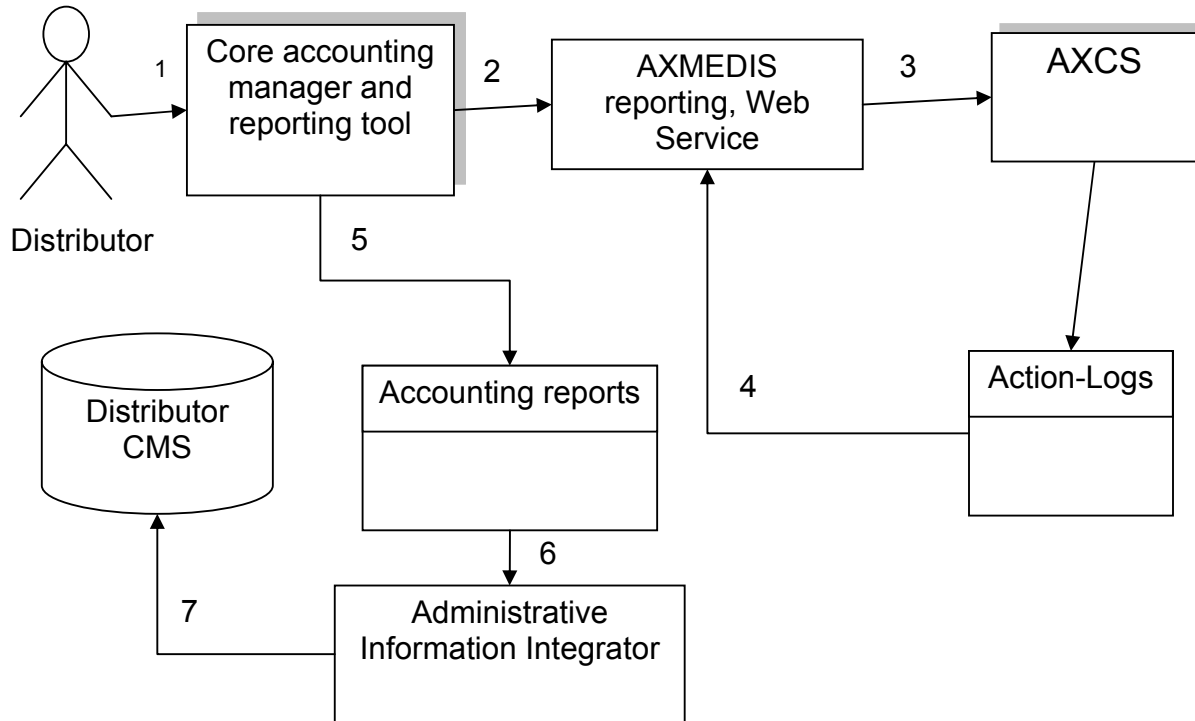
Administrative Information integrator is a critical part of the AXMEDIS system since it is the real bridge between the AXMEDIS world and the world of company CMS and CRM for taking in account administrative and legal aspects (such reclaim for payment not done and so on).

This component has also a double face since it can operate in a dual manner: used for polling information from AXMEDIS system when needed by distributor for example, or user for pushing information in the CMS as soon as they are available for example in the case of collecting societies.

The operation mode is determined by accounting people at the installation/configuration of the system when it will be established whose fields have to be exported from the DB to the CMS and the frequency of exporting. When a frequency is set, the Administrative Information Integrator will work in push mode, pushing information in the CMS import area, while when this frequency is not set the system operates in polling mode by starting the update in the CMS by a link to a web page.

The principal specification arises, as always, from requirements. Administrative Information Integrator has to:

- ***interface with different CMS technologies***: this means that administrative information integrator has to export its reports in a portable format such as an XML that will be defined in the detailed specification. The generated XML can then be parsed and transformed by the way of standard mechanisms such as XSL transformation;
- ***store administrative information into the Content Provider database***: this operation will be possible in two ways, or by preparing a file to be put in the import area on the content provider database or by interfacing with the services offered by the CMS of the Content provider by standard mechanisms such as web services or other remote interface available to the Administrative Information Integrator;
- ***communicate with the AXDB to get administrative information related to a specific Content Provider***: this is the minimum security requirement that have to be established in order to be sure to distribute information to entitled persons only.
- ***guarantee privacy of sensitive data via protection mechanisms***: apart from what has been already stated, if a network connection is necessary to transfer data and in the CMS of the Content Provider permit a connection on a secure channel the data will be sent encrypted.



1. A Distributor wants to recover information on actions performed on the objects he has rights.
2. Core accounting manager and reporting tool query the reporting webservice for obtaining the Action-Logs
3. AXMEDIS Statistic or reporting tools query AXCS
4. AXCS extracts the required Action-Logs and communicate them to the reporting tool
5. Accounting report is generated.
6. Accounting report is passed to the Administrative Information Integrator
7. Data are loaded in the Distributor CMS

The following three scenarios represent the relationship between AXMEDIS and the Collecting Societies providing a service that support Collecting societies in gathering reporting information on the use of the protected objects so to enhance the management, administration and enforcement of the rights they are vested in or represent.

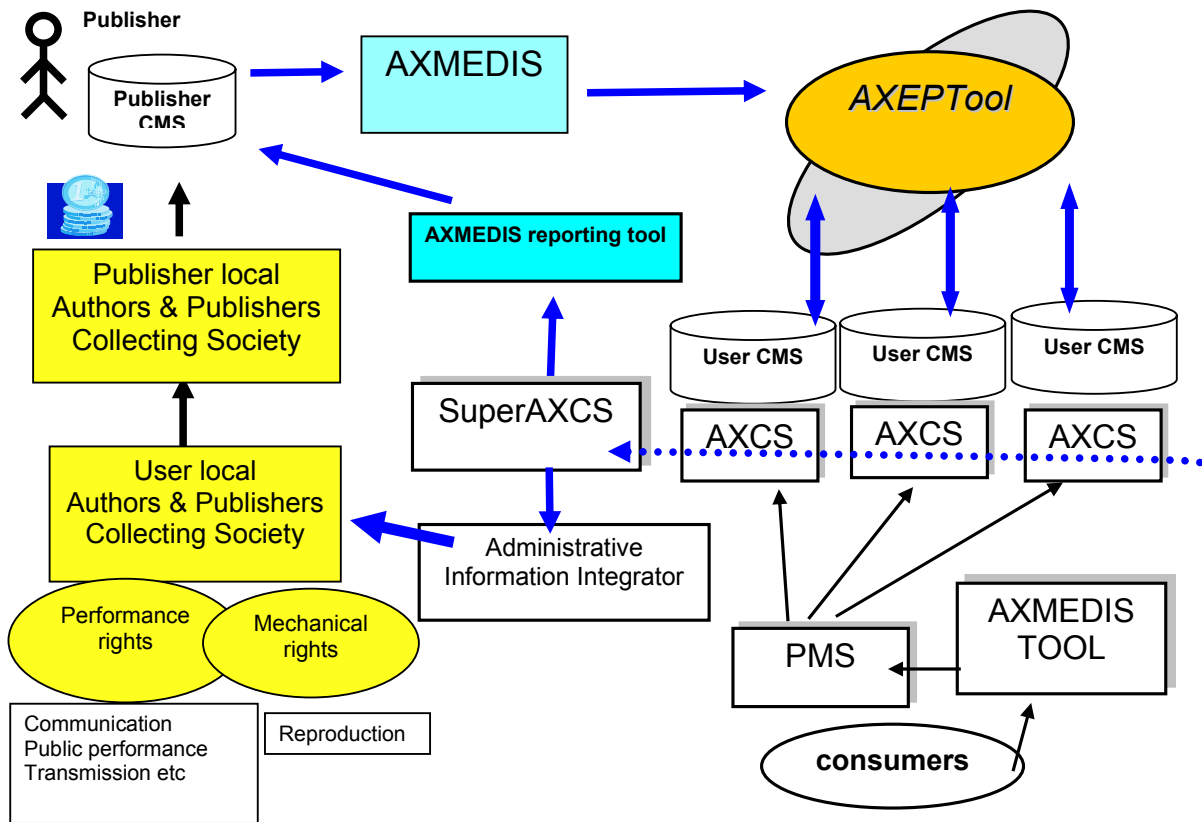
Collecting societies administer a wide range of rights on behalf of copyright owners for a wide range of uses and users and they collect and distribute to right owners royalty income and equitable remuneration in relation to the exercise of these rights. In addition to these core functions, there are many other functions carried out by all or many Collecting societies such as enforcement, monitoring and auditing activities, particularly important in view of the increasingly uses of copyrighted content in the AXMEDIS context.

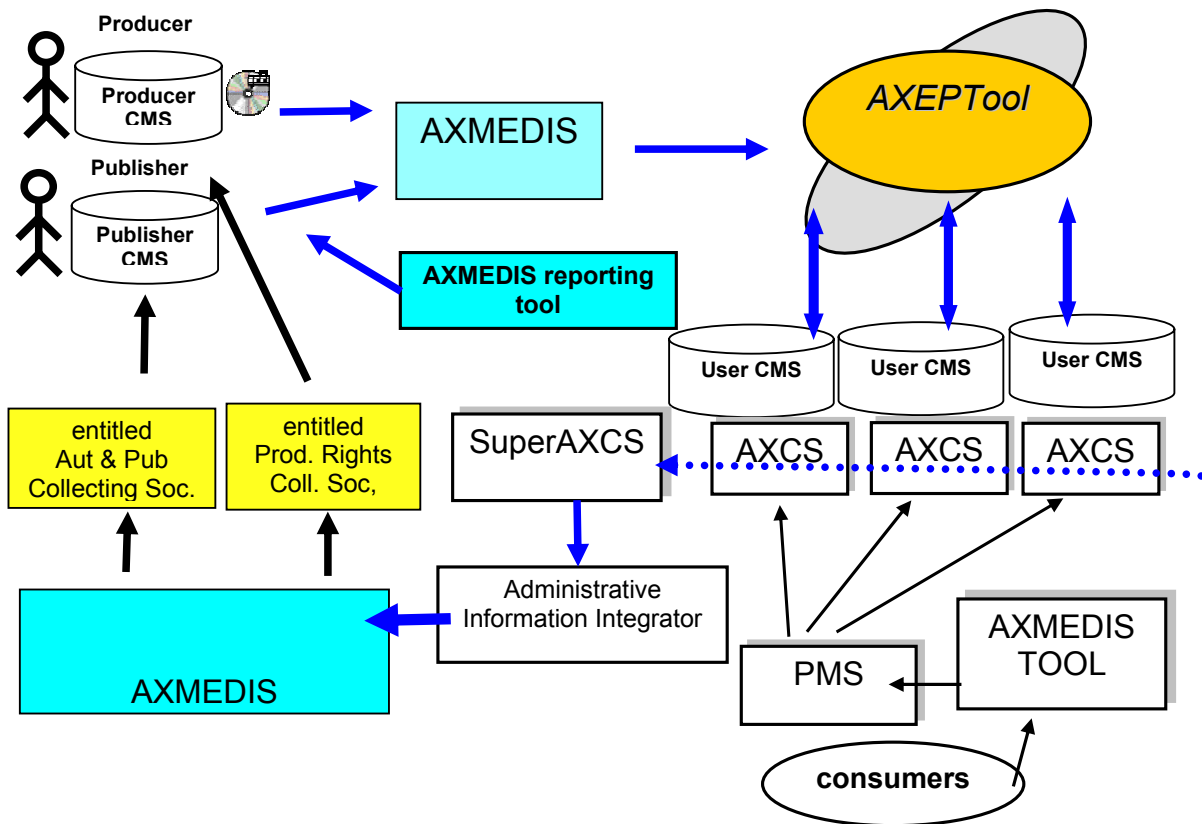
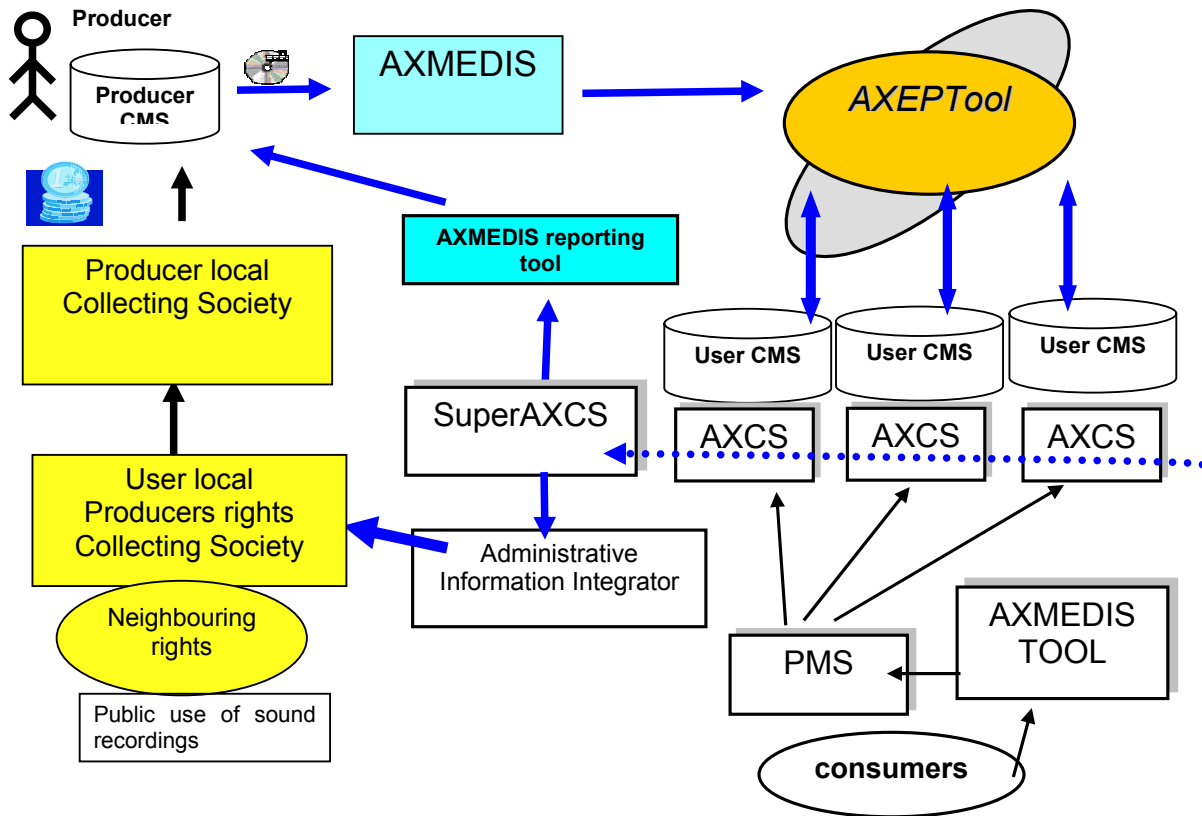
It has to be underline that these scenarios only refer to the use of music whose exploitation rights are granted to the original publisher and the producer. Their content, once governed as an AXMEDIS object, are ready to be exploited within the AXEPTOOL and distributed accordingly to the DRM and license terms provided. All actions (uses) or events performed on an that AXMEDIS objects and recorded in the AXCS of each user are then reported, along with other relevant data, to the super AXCS. The super AXCS tool will then interact with the AXMEDIS reporting tool and with the Administrative Information Integrator that will respectively report relevant information into the right owners database and into the database of the entitled collecting societies. It has to be underline that the link between AXMEDIS tools and the collecting societies should be implemented by taking into account new tools and network developed by collecting societies themselves such as the FastTrack project aiming at realizing a global interconnected network of databases on musical and audiovisual works, rights owners, contracts and data on sound recording to support diary operations of the societies involved such as identification of works and distribution of royalties.

Independently from the ways and methodologies rights are granted to users (e.g. compulsory license, individual license) the Administrative Information integrator tool should provide Collecting societies with data needed to check, verify and monitor the use of the AXMEDIS objects in conformity with the rights granted by the relevant license and with information necessary to identify right owners including identification standard codes already developed (such as the ISRC) as well as those under development.

An AXMEDIS object will involve a multiplicity of rights owners (there could be many just in one musical work) and many different collecting societies (such as public performance rights societies, mechanical rights societies, producers rights societies, performers rights societies etc) and its multi distribution channels will allow multiple reproductions, transmissions and retransmissions until it reaches the end user/consumer.

Due to this issue and to the complexity of different rules that govern the collection and distribution of royalties for the exploitation of multimedia contents and compounded objects in the digital environment, the control of the correct use of the rights granted and the consequent collection of the royalties due become a complex task and one of the main concern of right holders community.

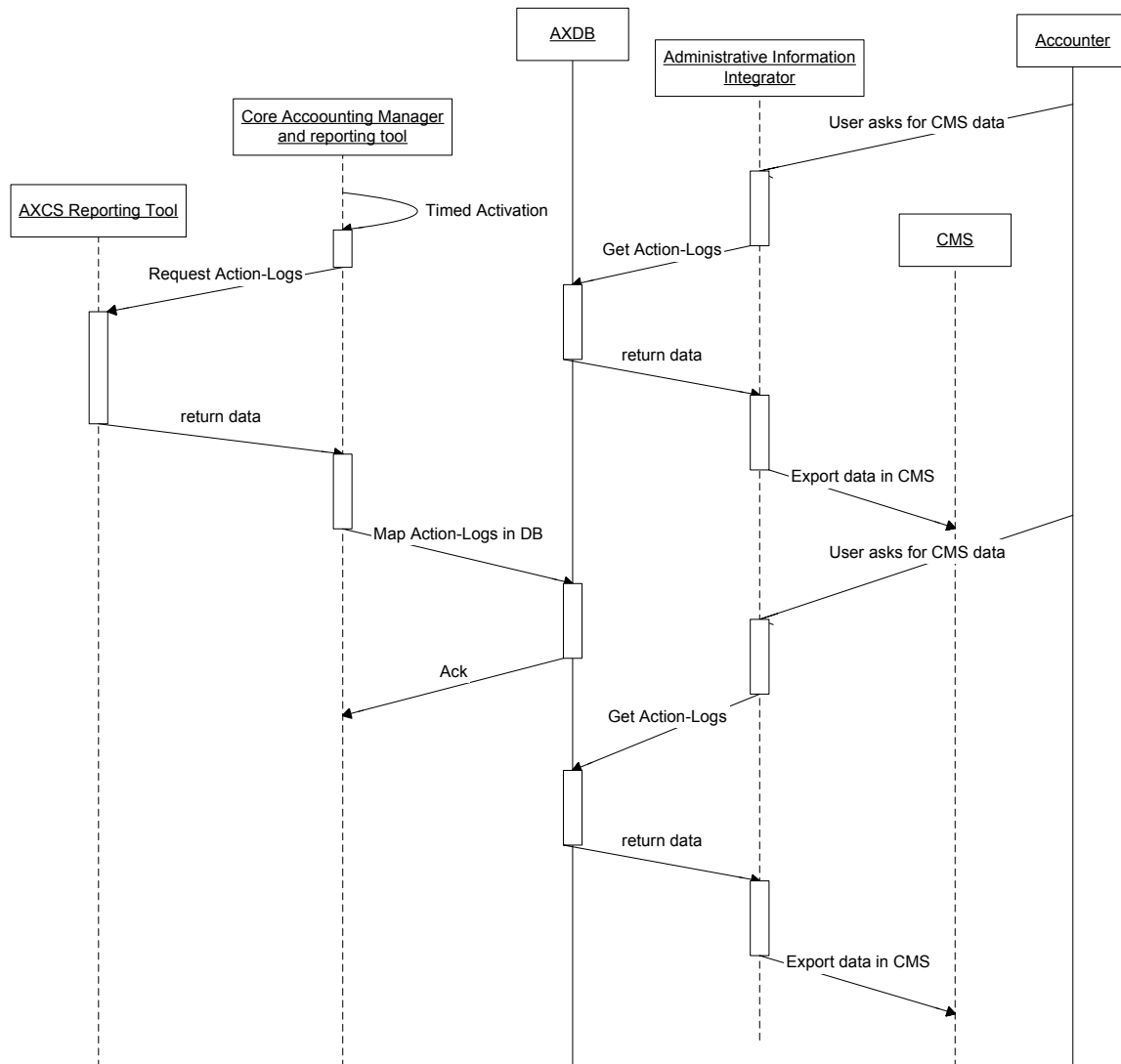




6.2.1 Administrative Information Integrator in polling mode

This is the operational mode when an automatic update of the data in the CMS is not set. In this mode the Accounter connects to a web page and issue a simple command to the Administrative Information Integrator that export in the CMS Import area the information that have not been already exported according to the format defined during the installation and configuration process.

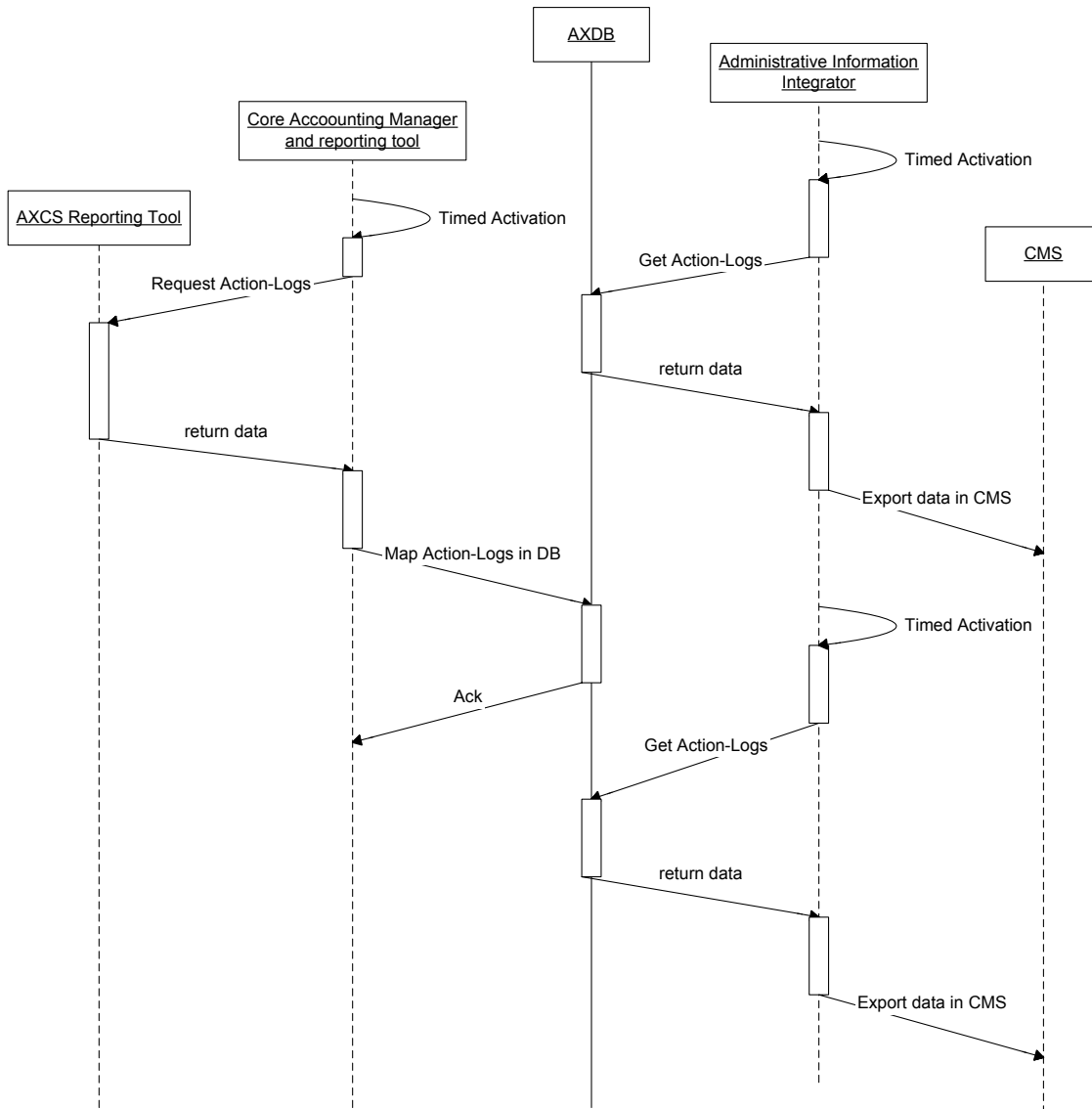
The process that happens in the Accounting Area when the Administrative Information Integrator is in polling mode can be modelled by the following diagram.



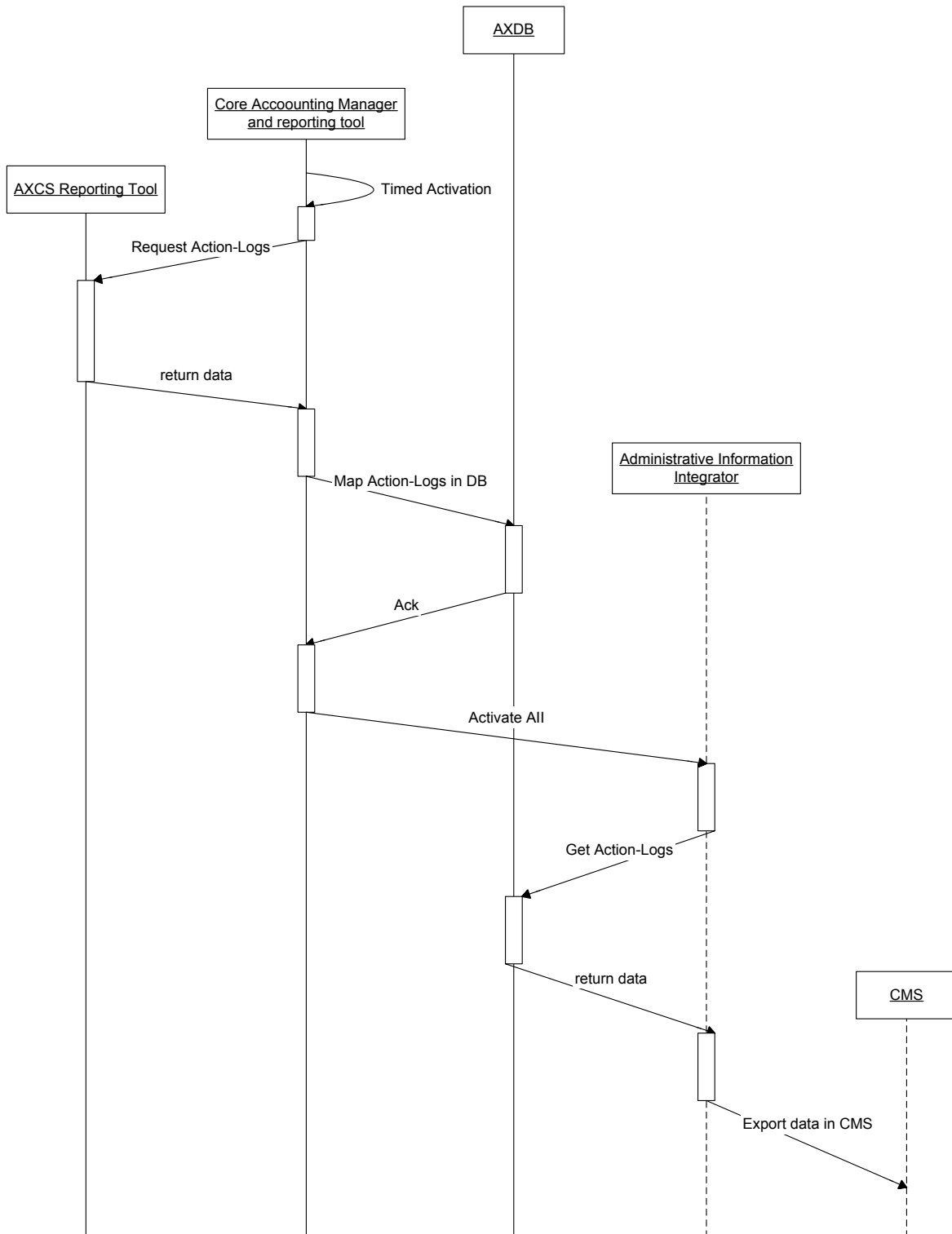
6.2.2 Administrative Information Integrator in push mode

All the times that during the configuration of the Administrative Information Integrator the timed insertion of administrative information has been selected, the Administrative Information Integrator is enabled to put in the import area of the CMS all the information that the CMS accounter has selected. The updating of the CMS should be also synchronized by a trigger sent by Core Accounting Manager and reporting tool.

In the following diagram the timed activation is reported, by which the Administrative Information Integrator and the CAMART operates asynchronously on a timed basis.



In the following diagram the timed activation is reported, by which the Administrative Information Integrator is triggered by the CAMART.



6.2.3 Administrative Information Integrator User Interface

The user interface for the Administrative Information Integrator is able to configure the tool in the different modes (pushing or polling mode). In the case of pushing mode the frequency at which the push will be made can be set, while in polling mode, the file can be displayed or downloaded. In any case the formatting style can be selected among those present in the database. Refer to the Specification Part E for the support database needed by the Administrative Information Integrator.

The screenshot shows a web browser window titled "Your Title Here - Mozilla Firefox". The address bar displays a file path: `file:///d:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/Current-PART-E/AII`. The browser's toolbar includes various icons for navigation and search. Below the browser window, the "Administrative Information Integrator Manager" interface is visible. It contains several input fields and buttons:

- User** and **Password** text input fields.
- Exporting Style** dropdown menu set to "CMS1 style".
- Exporting Path for Pushing** text input field.
- Save Configuration** button.
- Pushing Mode:**
 - Enable Pushing Mode** checkbox (unchecked).
 - Push Data every** dropdown menu set to "10 minutes".
 - Ok** and **Clear** buttons.
- Polling Mode:**
 - Download** and **View** radio buttons (both unchecked).
 - Start** and **Clear** buttons.

In the case of pushing mode, a daemon will be created, that for each user enabled to get data from Administrative Information Integrator will automatically download in the configured path the fresh logs. System will take care of keeping in the database the timestamp of last download (both for polling and pushing) in order to have for each user a stored profile with the last logs downloaded.

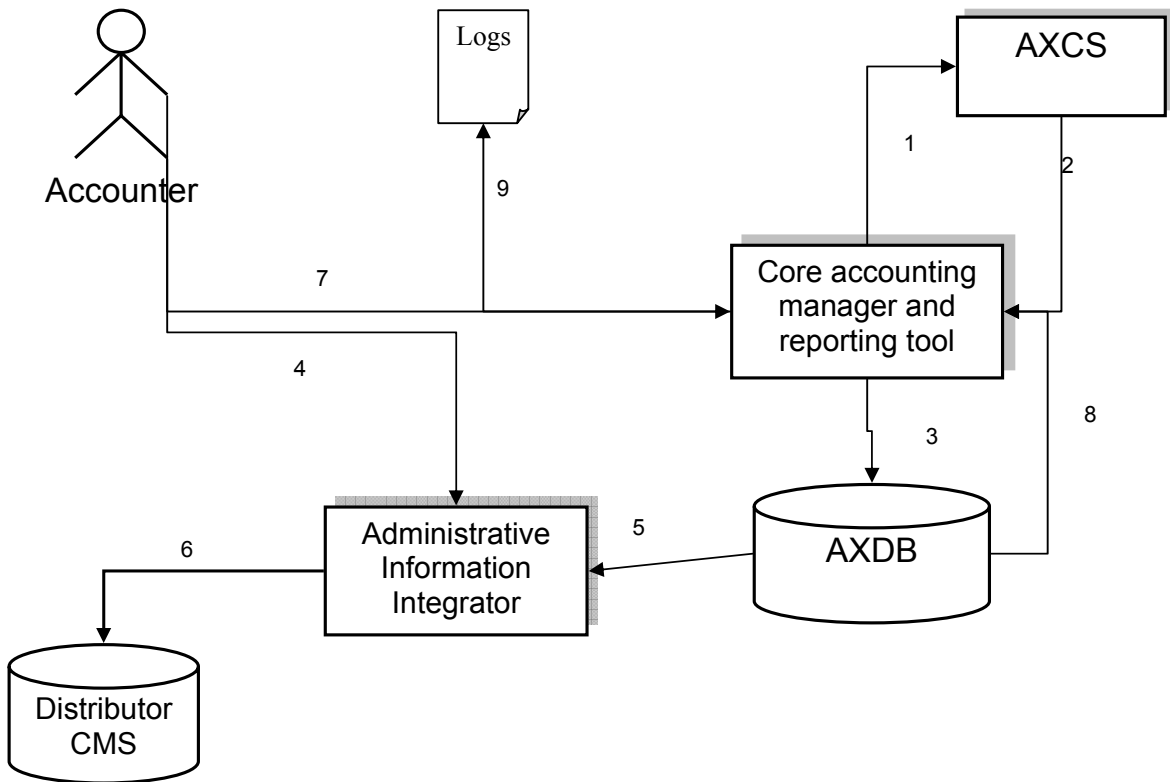
This user interface is for configuring how the AII will interact with CMS in terms of periodic update of the file to be imported by the CMS, the style that have to be applied in order to have multiplatform capability for different CMS. The timestamp of the extraction will be registered automatically from the system and there is no need for the user to take care of that. This information can be displayed in the previous web page.

6.2.4 Administrative Information Integrator and CAMART integration

It is necessary to point out how the CAMART and AII are related in order to have the right tool doing the right work.

CAMART is the only interface toward the AXCS, while AII will read the Logs that are already in the AXDB; CAMART has the duty to put log on AXDB with a predefined scheduled period. CAMART will offer to the user an interface for getting logs that are on AXDB independently of the CMS exporting.

All the activities can be summarized in the following scenario:



1. CAMART contact AXCS web service for having fresh logs
2. AXCS give back to CAMART the Logs
3. CAMART store logs on AXDB
4. Accounter configure AII to put export logs toward CMS
5. AII get logs from AXDB
6. AII export logs to CMS
7. Accounter can optionally query directly the CAMART for logs filtered by AXOID, AXUID, timestamp etc
8. In that case CAMART will get the log from AXDB
9. CAMART will generate a raw XML report of Logs