



Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE3.1.2E

Framework and Tools Specifications (Database and Gathering)

Version: 2.2

Date: 24/10/05

Responsible: EXITECH (DSI supervision)

Project Number: IST-2-511299

Project Title: AXMEDIS

Deliverable Type: PRIVATE

Visible to User Groups: No

Visible to Affiliated: No

Visible to the Public: No

Deliverable Number: DE3.1.2 Part E

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): EXITECH, DSI, FUPF, FHGIGD

Abstract:

Part E of the specification deals with the problems related to the Database Area and Data Gathering and therefore the specification of the database and the modalities to access data in the AXDB are reported together with the related tools such as crawling system for importing in AXMEDIS existing contents in user CMS; Query support for allowing user and tools to query the system; support for storing and querying licences; support for automatic generation of licenses and contracts; production on demand and its relationships with gathering and queries.

Keyword List: Database, query, crawler, production on demand, licences, PAR

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	7
2	CRAWLING AREA (DSI, EXITECH)	8
2.1	CRAWLER COLLECTOR INDEXER (WP4.1.1: DSI WITH SUBCONTRACT).....	9
2.1.1	Focusseek plug-in architecture.....	15
2.2	CRAWLER RESULTS INTEGRATED DATABASE(WP4.2.1: DSI WITH SUBCONTRACT).....	17
2.3	COLLECTOR ENGINE (WP4.2.1: DSI WITH SUBCONTRACT)	18
2.4	JAVAScript TO CRAWLER INTERFACE (WP4.2.1: DSI WITH SUBCONTRACT).....	18
2.5	DATABASE INTERFACE WITH LOBSTER/TAMINO	19
2.6	DATABASE INTERFACE WITH EXPORTED XML CONTENT	20
2.7	DATABASE INTERFACE WITH ODBC.....	21
3	AXMEDIS DATABASE AREA (EXITECH, DSI)	22
3.1	AXMEDIS OBJECTS REPOSITORY RELATION SCHEMA (EXITECH, DSI)	26
3.2	DESCRIPTORS METADATA MAPPING IN RELATIONAL SCHEMA (EXITECH, DSI, CRS4)	28
3.2.1	AXInfo table.....	30
3.2.2	DID.....	30
3.2.3	AccessMode	30
3.2.4	ObjectStatus	30
3.2.5	FingerPrint.....	31
3.2.6	PromorOf.....	31
3.2.7	Translations.....	31
3.2.8	MetadataAdditionalInfo	31
3.2.9	Descriptors	31
3.2.10	Creator	31
3.2.11	Distributor.....	31
3.2.12	Owner.....	31
3.2.13	OptionalField	31
3.2.14	DublinCore (DCMI).....	31
3.2.15	DCMIContributors	32
3.2.16	DCMICreators	32
3.2.17	DCMILanguages	32
3.2.18	DCMISources	32
3.2.19	DCMIPublishers	32
3.2.20	DCMIRelations.....	32
3.2.21	DCMIRights	32
3.2.22	DCMIMedium	32
3.3	INTEGRATION BETWEEN PAR DB AND DESCRIPTORS DB FOR MAKING QUERIES (EXITECH, FUPF).....	32
3.4	ACCOUNT LOG FOR AXMEDIS OBJECTS REPOSITORY (EXITECH, FUPF)	33
3.5	DATABASE SCHEMA FOR SUPPORTING AXMEDIS (EXITECH, FUPF)	34
3.5.1	User and groups	34
3.5.2	Query and selection archive.....	35
3.5.3	Version history and Protection Info.....	36
3.5.4	Query Distribution and Integration.....	37
3.5.5	Administrative Information Integrator	38
3.5.6	P2P Hub Node Support (EXITECH, CRS4).....	38
3.6	AXMEDIS DATABASE INTERFACE (EXITECH, DSI)	39
3.7	AXMEDIS DATABASE WEBSERVICE INTERFACE (EXITECH, DSI, CRS4).....	44
3.7.1	Descriptor_support (EXITECH, FUPF).....	46
3.7.2	Publication_support (EXITECH, CRS4)	70
3.7.3	User Support.....	73
3.7.4	CAMART Support.....	74
3.7.5	AXDBQuerySupport.....	78
3.7.6	P2P Hub Node Support.....	78
3.8	AXMEDIS ADMINISTRATIVE WEB DATABASE INTERFACE (EXITECH, DSI)	83
3.8.1	Add user	87

3.8.2	Delete user.....	88
3.8.3	Update user data.....	89
3.8.4	Update user rights	90
3.8.5	Assign user to additional groups	91
3.8.6	Add Group.....	92
3.8.7	Delete group.....	93
3.8.8	Update group data.....	94
3.8.9	Assign additional users to group	95
3.8.10	Delete Object.....	95
3.8.11	Mark an Object as unavailable.....	96
3.8.12	Delete one or more versions of an Object.....	96
3.8.13	Download Object.....	96
3.8.14	Insert/Update Object.....	97
3.8.15	Query Interface.....	97
3.8.16	Selection Interface.....	97
3.9	AXMEDIS OBJECT LOADER/SAVER (EXITECH, DSI, CRS4).....	97
3.9.1	Saver/Indexer (EXITECH, DSI, CRS4).....	98
3.9.2	Loader.....	106
3.10	PROTECTION MODELS FOR AXMEDIS OBJECTS REPOSITORY (FUPF, AFI).....	112
3.10.1	From REL licenses to ER model.....	113
3.10.1.1	UML diagram for Licenses	114
3.10.1.2	ER diagram for Licenses	115
3.10.1.3	Information inside the Conditions table.....	121
3.10.1.4	Example of information inside the License Database	123
3.10.2	Proposal of PAR description based on REL information.....	127
3.10.3	UML diagram for PAR.....	127
3.10.3.1	ER diagram for PAR	128
3.10.3.2	Pricing for Final users	132
3.10.4	Description of Web Services for accessing license and PAR database	132
3.10.4.1	Architecture of LicenseDBManager module.....	132
3.11	AUTOMATIC GENERATION OF CONTRACTS AND LICENSES AND FEATURE EXTRACTION FROM CONTRACTS	151
3.11.1	Module profile definition	151
3.11.2	Architecture of the module.....	152
3.12	HISTORY OF AXMEDIS OBJECTS EVOLUTION IN PRODUCTION REPOSITORY (EXITECH, FUPF)	157
3.13	ER DIAGRAM EXTENDED DESCRIPTION	158
4	AXMEDIS QUERY SUPPORT (EXITECH).....	158
4.1	QUERY SUPPORT GENERAL ARCHITECTURE.....	158
4.2	AXMEDIS QUERY SUPPORT SCENARIO	161
4.2.1	SCENARIO 6: Content accessibility, querying	162
4.2.2	SCENARIO 6.1: Making query on the several direction. Technical Querying, Collecting and integrating results. 163	
4.2.3	SCENARIO 6.2: Producing a selection.....	164
4.2.4	SCENARIO 6.5: Store a query/Selection and make it active later.....	165
4.2.5	SCENARIO 6.6: Make a query on a Kiosk (EXITECH, DSI, ILABS).....	165
4.3	QUERY FORMAT AND LANGUAGE (EXITECH)	167
4.3.1	General consideration on queries and on object search	167
4.3.2	Schema for an AXMEDIS query.....	167
4.3.2.1	Query sources.....	172
4.3.2.2	Query result	172
4.3.2.3	Query conditions	172
4.3.2.4	Similarities.....	172
4.3.2.5	XML Query example file (EXITECH, ILABS, COMVERSE)	173
4.3.3	Schema for an AXMEDIS query result.....	178
4.3.3.1	XML results example file.....	179
4.4	SELECTION FORMAT AND STORAGE (EXITECH).....	180
4.4.1	XML Selection Schema	181
4.4.2	XML Selection Samples	184
4.4.3	XML Selection Storage	185
4.5	QUERY USER INTERFACE (EXITECH, CRS4, DSI)	185
4.6	QUERY AND SELECTION ARCHIVE FOR EACH USER (EXITECH, DSI, HP, XIM, IRC)	185
4.6.1	Actualize Listener	196

4.7	QUERY SUPPORT WEB SERVICE INTERFACE (EXITECH).....	199
4.8	QUERY DISTRIBUTION (EXITECH).....	207
4.9	QUERY RESULTS INTEGRATION (EXITECH)	208
5	QUERY FOR PRODUCTION ON DEMAND.....	210
5.1	QUERY SUPPORT FOR DISTRIBUTION CHANNELS (FHGIGD)	211
5.2	QUERY SUPPORT FOR CLIENTS (FHGIGD).....	213
5.3	CLIENT PROFILE (FHGIGD)	214
5.4	DISTRIBUTION PROFILE (FHGIGD)	217

1 Executive Summary and Report Scope

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 E only and it is focused on producing a first version of the specification without having deep details to be adopted as a baseline in the specification tasks of the different work packages that will be developed in the first 18 months of the project.

The part E of the specification deals with the problems related to the Database Area and Data Gathering and therefore the specification of the database and the modalities to access data in the AXDB are reported.

Specification Part E is structured in 4 sections dealing with different aspects of data gathering:

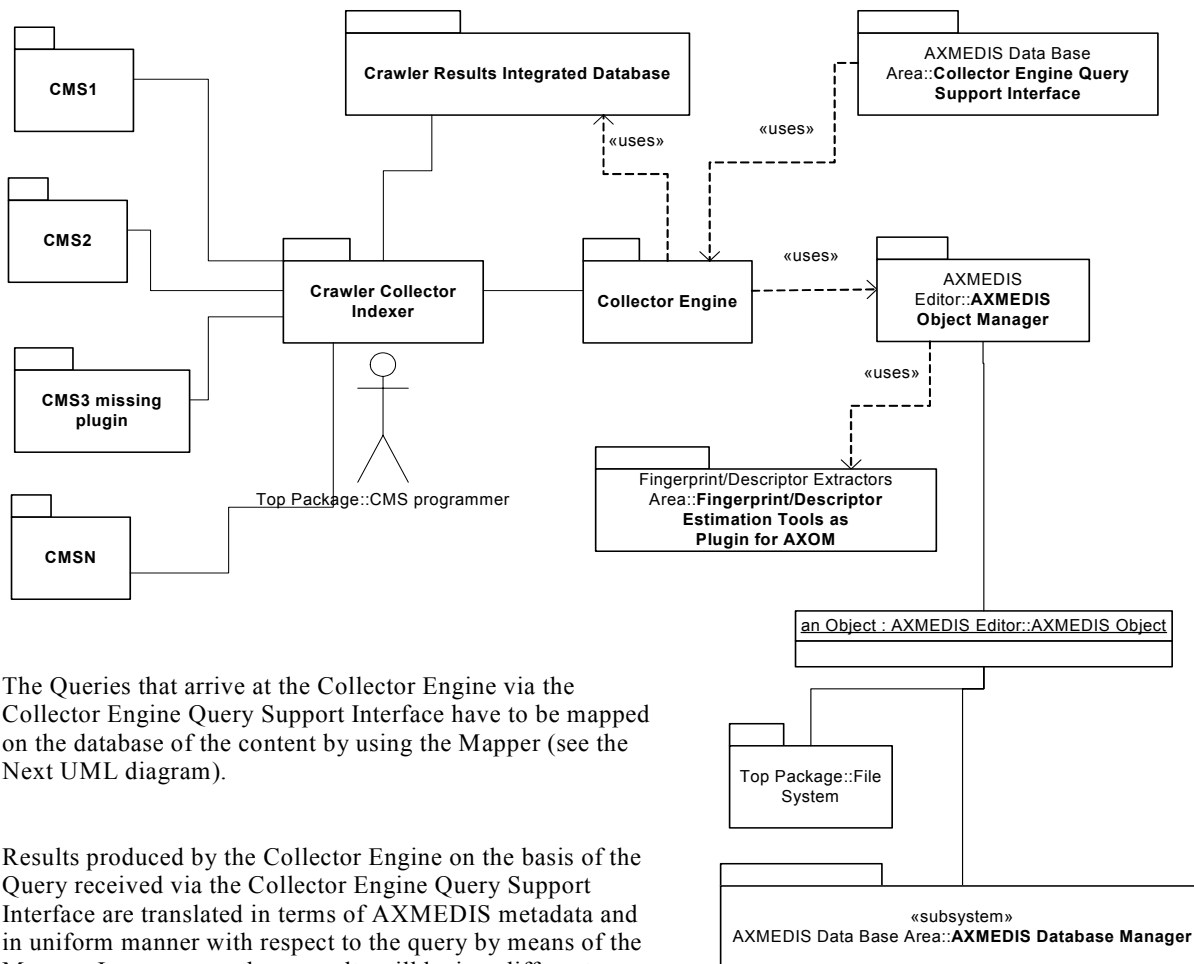
- Crawling area (under responsibility of DSI): this section describes how the AXMEDIS system interface the Crawling system in the content database of the user in order to find and automatically import in AXMEDIS the contents already present in digital form in the user CMS. Different interfaces and API will be implemented on the basis of the different CMS and interfaces provided by CMS;
- AXMEDIS database area (under responsibility of EXITECH, FUPF): in this section the mapping of MPEG21 standards in a relational DB has been specified in order to allow the storing of AXMEDIS objects that are MPEG21 objects, also if a simplified structure for allowing a more focused search of the parameters really needed by partners and experts has been provided. A large emphasis has been devoted to the specification of the metadata descriptors that are needed for categorize and search AXMEDIS objects with particular reference to AXInfo and DCMI. ER diagrams and textual description of such diagrams are reported. In this section also the aspects related to the storing of log, users, versions of objects and load/save of objects are reported. In general all the support tables needed by the different tools/engines of AXMEDIS have been specified. This section is also related to Protection Models for AXMEDIS Object repository where model for licences and PAR (both storing and querying) are presented together with the related webservice functionalities; and to Automatic generation of contracts and licenses and feature extraction from Contracts that have 4 main objectives (i) Generate license templates (or license models) from PARs (Possible Available Rights), (ii) Generate license templates from contracts (iii) Generate licenses from contracts, and (iv) Generate contracts from licenses.
- AXMEDIS Query support (under responsibility of EXITECH): in this section all the aspects related to querying (from formal model of queries and selection to description of related web services and mechanism for distributing and collecting results from different query channels are discussed. Some examples of real queries provided by partner have been inserted in order to allow a continuous verification of requirements against what has been specified.
- Query for production on demand (under responsibility of FGHIGD): in this section all the aspects related to the relationship between the production of contents on demand and the mechanisms for querying the system will be analyzed and discussed. Distribution channel can interact with AXMEDIS query support via the Query support Web service interface that is a standard technology for sharing information in Internet by using a standard protocol with a standard definition language for service interface. The Query Support for Distribution Channels gets a client queries from the Distribution Server which include a client and a distribution profile. The Query Support for Distribution Channels verifies and adjusts such a query according to client and distribution profile.

2 Crawling Area (DSI, EXITECH)

The Crawling area includes:

- **Crawler Collector Indexer:** implemented by the focuseek platform, is responsible to gather and index contents from CMSs
- **Crawler Results Integrated database:** a component of focuseek platform containing the local copy of original content.
- **Collector Engine:** component in charge of creating AXMEDIS Objects
- **CMS Missing Interface(s):** gathering plug-ins for CMSs
- **Collector Engine Query Support Interface:** interface to query Crawler database

Crawling Area



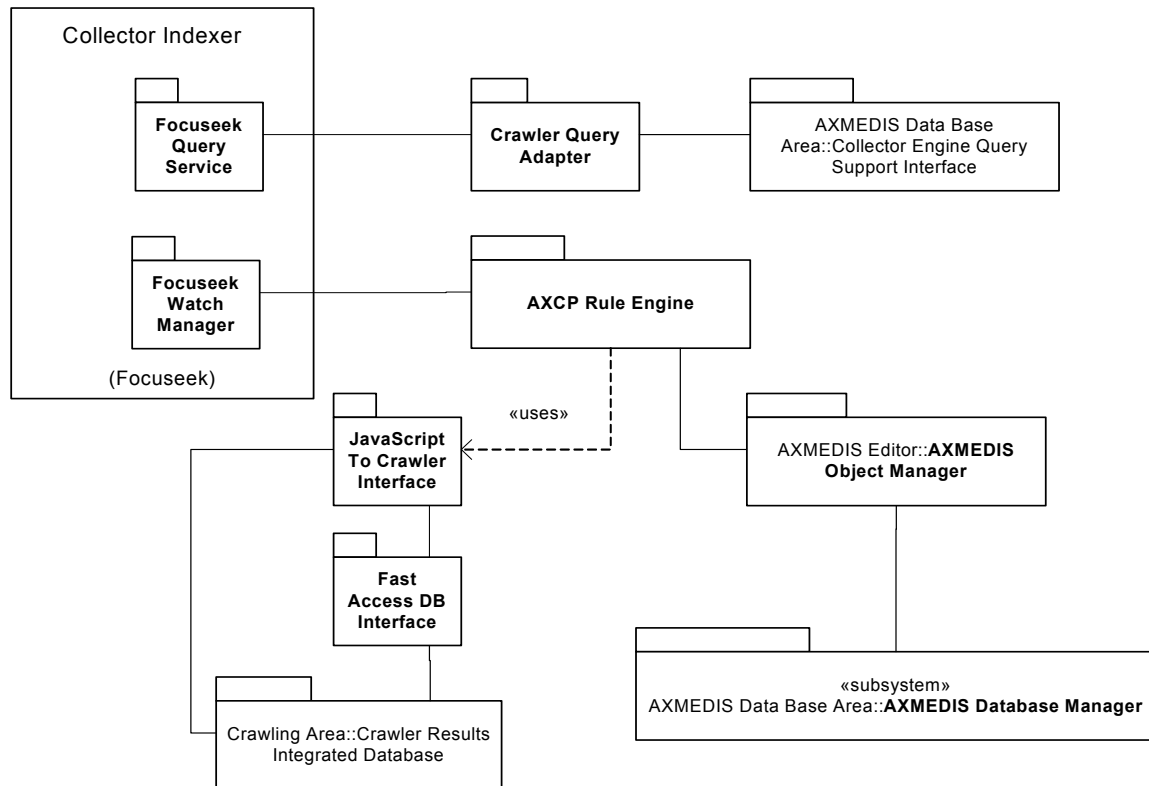
The Queries that arrive at the Collector Engine via the Collector Engine Query Support Interface have to be mapped on the database of the content by using the Mapper (see the Next UML diagram).

Results produced by the Collector Engine on the basis of the Query received via the Collector Engine Query Support Interface are translated in terms of AXMEDIS metadata and in uniform manner with respect to the query by means of the Mapper. In any cases these results will be in a different format with respect to those coming from the AXDB and/or AXEPTool

The Collector Engine includes:

- **AXCP Rule Engine:** to execute rules
- **Crawler Query Adapter:** to adapt queries for the crawler
- **JavaScript to Crawler Interface:** interface to access Crawler database inside JS scripts
- **Fast Access DB Interface:** a socket based interface to the Crawler Results Integrated Database

Collector Engine



2.1 Crawler Collector Indexer (WP4.1.1: DSI with Subcontract)

Module Profile		
Crawler Collector Indexer		
Executable or Library(Support)	Executable	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Responsible Name	Nicola Baldini	
Responsible Partner	Focuseek	
Status (proposed/approved)		
Platforms supported	Linux, Windows 2000/XP, Mac OS X	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
Crawler Integrated DB		Web Service
File Formats Used	Shared with	File format name or reference to a section
FFF (Focuseek Flexible Format)		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Crawler User Interface	C++,	wxWidget 2.5

Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
CURL	Curl 7.12.3	BSD

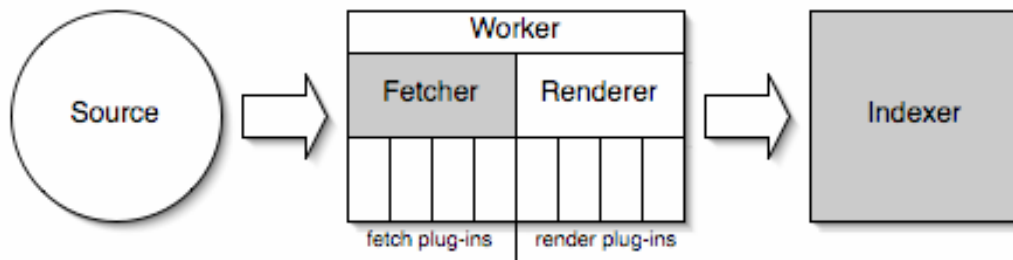
Focuseek will be used to crawl into the CMSs and the databases of the content providers, integrators, and distributors to access content and indexing it. Specific interfaces for accessing to these databases will be defined and developed. We will design and develop a framework (based on Focuseek) which allows collecting and indexing different media sources.

The idea is to access already available information and metadata into CMSs and transfer it into a cache of the content. Those may have information related to the content, their metadata, their technical details, DRM aspects, licensing aspects, possible products that use the content, multilingual information, etc...

This tool is called in the general architecture as Crawler Collector Indexer. It may navigate on any kind of source by means of special plug-ins that will be developed.

This task will be performed by acquiring the license of a product called Focuseek and integrating it in the AXMEDIS framework. The license will come with editable manual to customize the installation to the AXMEDIS solution.

This component is already integrated into the focuseek platform and it is represented by the union of the fetching agent (fetcher) and of the indexing service (indexer).



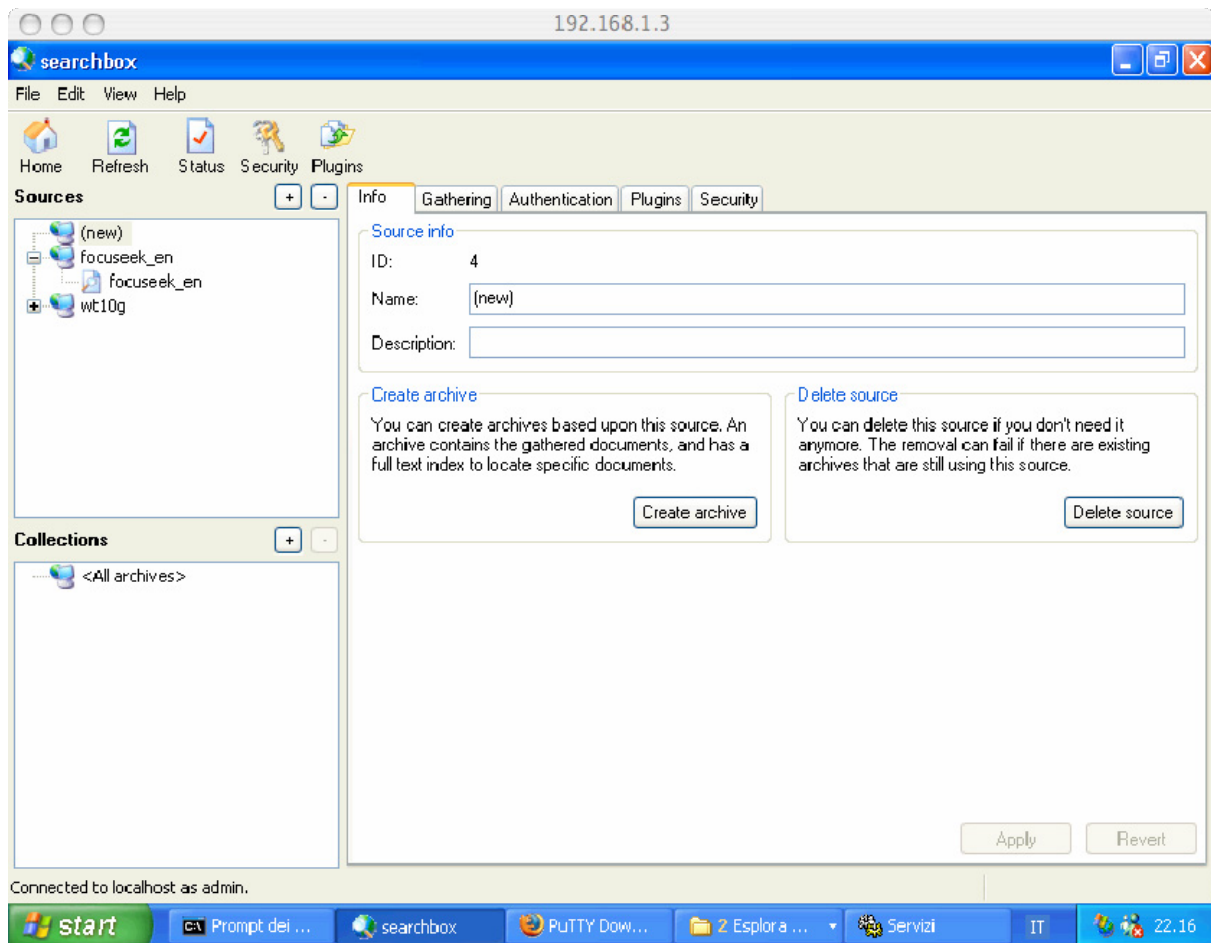
In the previous picture with light grey color the subcomponent of the focuseek architecture involved in this portion of AXMEDIS system.

Note that if the Renderer component of Worker has no active plug-ins configured it is excluded from the overall process.

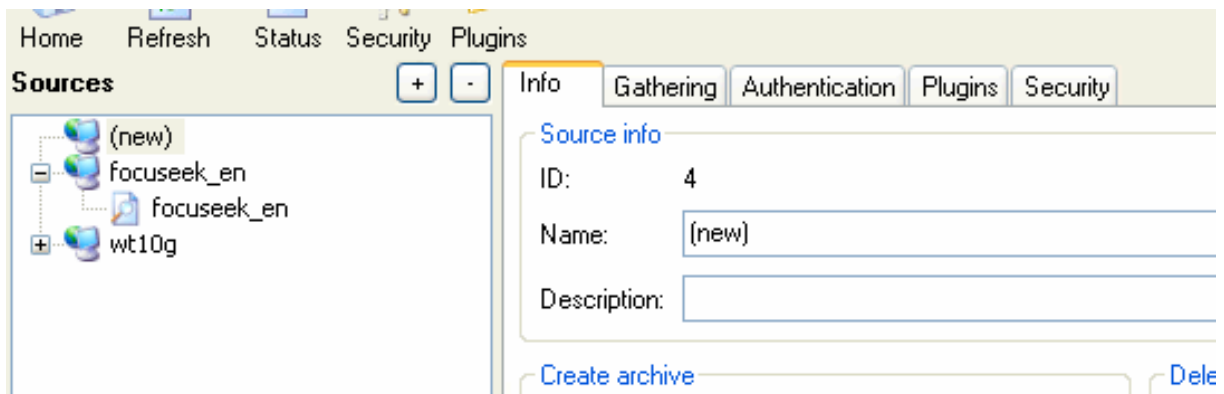
The other type of plug-in (fetch plug-in in the picture) are those used to let the fetcher to gather directly contents from sources. Custom plug-ins for specific CMS will be developed, focuseek supports by defaults HTTP, NNTP, SMB, FTP, local FS and ODBC protocols.

The Crawler User Interface will be derived from the standard focuseek GUI administration tool (focuseek Control Panel) that will be modified for this specific purpose.

The current focuseek administration tool can be used to setup sources and the connected archives together with the crawling policies. The following picture shows the general aspect of the administration tool.

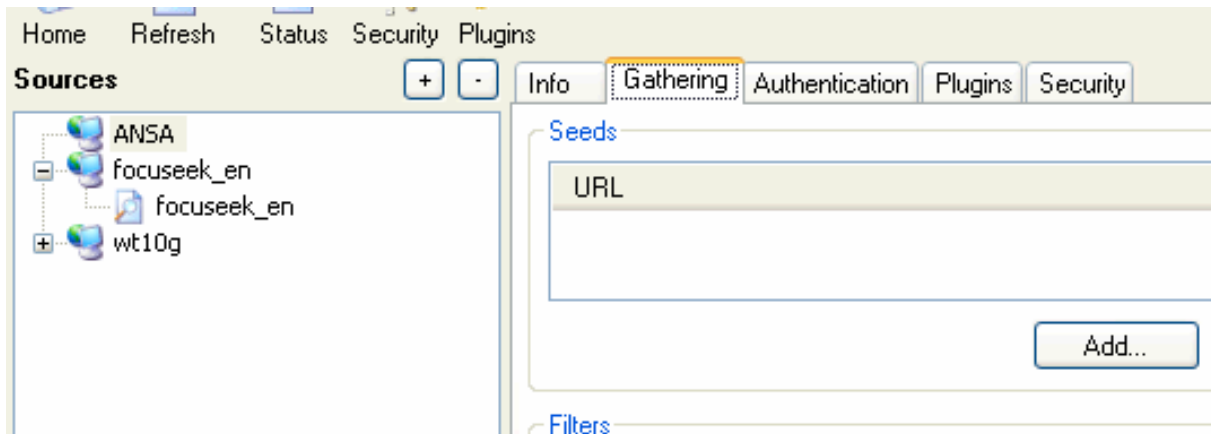


The process to add a new source to the system is very straightforward¹:



Using the button + a new entry in the left pane is added. Using the tabs of the central pane the new source can be fully configured. In the above figure the default name (new) must be substituted with the real one.

¹ This is only an example based on the feature currently available in the focuseek platform, specific feature will be developed during the project.



The next tab is used to insert the seeds of the source we are going to create. A seed is the physical place where contents reside (usually a URL). Many seeds can be added for a single source.

Url

Resource

☒ Web site

Address: http://

☐ Secure

☐ FTP site

Address: ftp://

☐ Gopher site

Address: gopher://

☐ Usenet news

Server:

Newsgroup:

☐ Filesystem

Server:

Path:

☐ Search engine

Search: on

☒ Add default inclusion filter

OK Cancel

Choosing to add a seed the system ask which type o seed must be added (we cha choose between all type of sources managed by the platform). Let's suppose to choose an HTTP source like <http://www.ansa.it>

The screenshot shows the 'Plugins' window with the 'Gathering' tab selected. It contains three main sections:

- Seeds:** A text box labeled 'URL' contains 'http://www.ansa.it/'. Below it are buttons for 'Add...', 'Edit...', 'Remove', and 'Browse'.
- Filters:** A text box labeled 'Match' contains the regular expression '^http://www\.\ansa\.it/.*\$'. Below it are buttons for 'Add...', 'Edit...', and 'Remove'.
- Crawl limits:** A section with two checkboxes: 'Depth limit:' (checked) with a value of '2', and 'Check robots.txt' (checked). There is also an unchecked checkbox for 'Has side-by-side metadata'.
- Text exclusion markers:** A section with 'Begin:' and 'End:' text boxes, and 'Apply' and 'Revert' buttons.

Crawling filters used to configure the action of the crawler agent can be added in the same tab as regular expressions (In the above picture filters used to crawl ANSA web site).

The screenshot shows the 'Sources' window with the 'Info' tab selected. It displays a list of sources on the left and configuration details for the selected 'ANSA' source on the right.

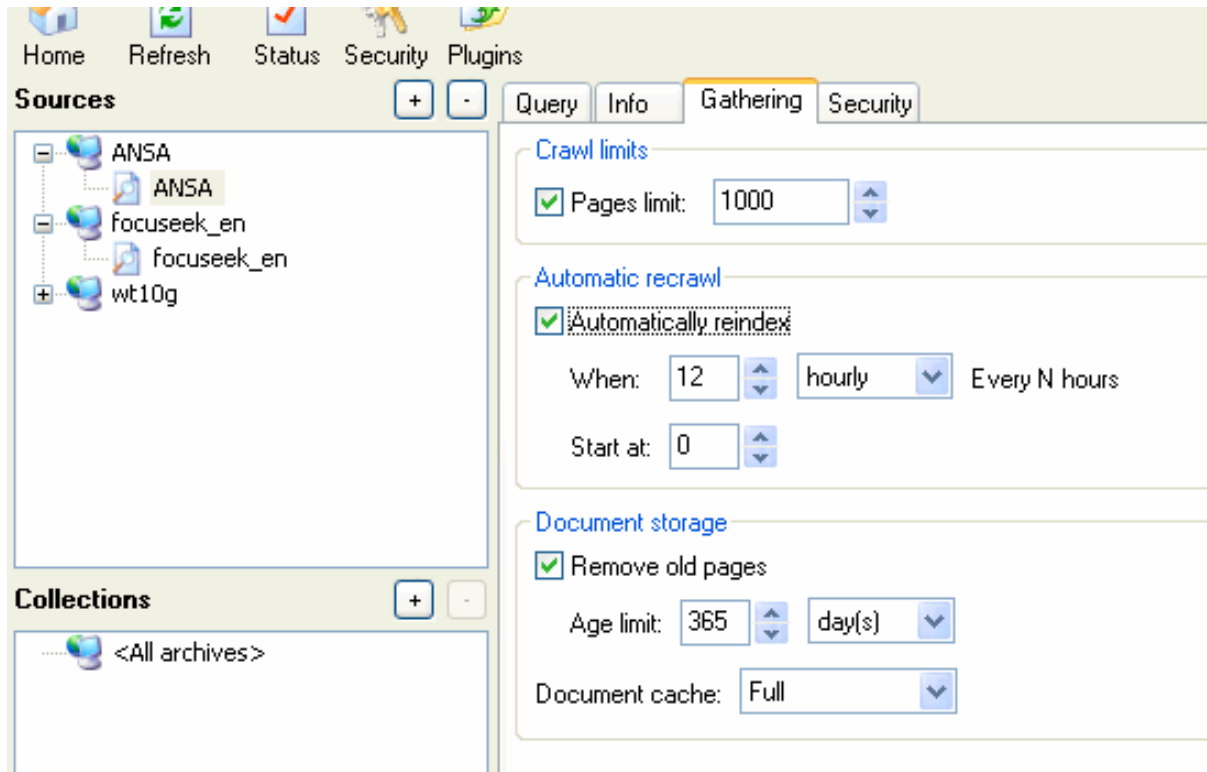
Sources List:

- ANSA (selected)
- ANSA
- focuseek_en
- focuseek_en
- wt10g

Archive info:

- ID: 4
- Name: ANSA
- Contact: (empty text box)
- Historicize: Yes

Once configured a source must have associated one or more archives. By default archives have the same name of the related source.



The third tab of archive configuration is devoted to schedule the action of the crawler. We can setup an upper limit on the number of pages, the time when the system have to poll the source and the deletion policy of garbage collector.

The same configuration philosophy will be implemented for the new kind of sources involved in the project.

2.1.1 Focuseek plug-in architecture

All plug-ins in the focuseek architecture are managed by a single ocmponent called Worker. Such component expose a standard C API and all Plug-Ins can be implemented as standard DLL.

focuseek searchbox supports three plugin types:

- Protocol plugin. Plugins implementing a standard protocol (e.g. http) or a custom one.
- Parser plugin. Plugins that parse documents (e.g. HTML documents).
- Mission plugin. Plugins that coordinate the document gathering/parsing process.

Protocol plugins

Protocol plugins basically accept an URL specifying a document and retrieve it. Their output comprises an error code (e.g. "Document not found"), the document contents as a binary blob of data and optionally the MIME type for the document if the protocol supports it [1].

Parser plugins

Parser plugins receive the original document contents, i.e. a blob of binary data, and extract textual and typographical information from it. A parser plugin input contains info about the document, such as the document URL, MIME type and contents. It also contains a list of info about “related documents”: documents that are required to correctly process the original document. For example a plugin for the HTML format could require a CSS style sheet, referred in the HTML document itself.

The plugin outputs a list of requests for related documents and a list of parsed documents. Parsed documents are expressed in the FFF format and will be further processed by other plugins and by searchbox itself.

The plugin is invoked repeatedly until no more related documents are required or no more related documents can be successfully retrieved. At this point the plugin must emit the processed documents or signal an error condition.

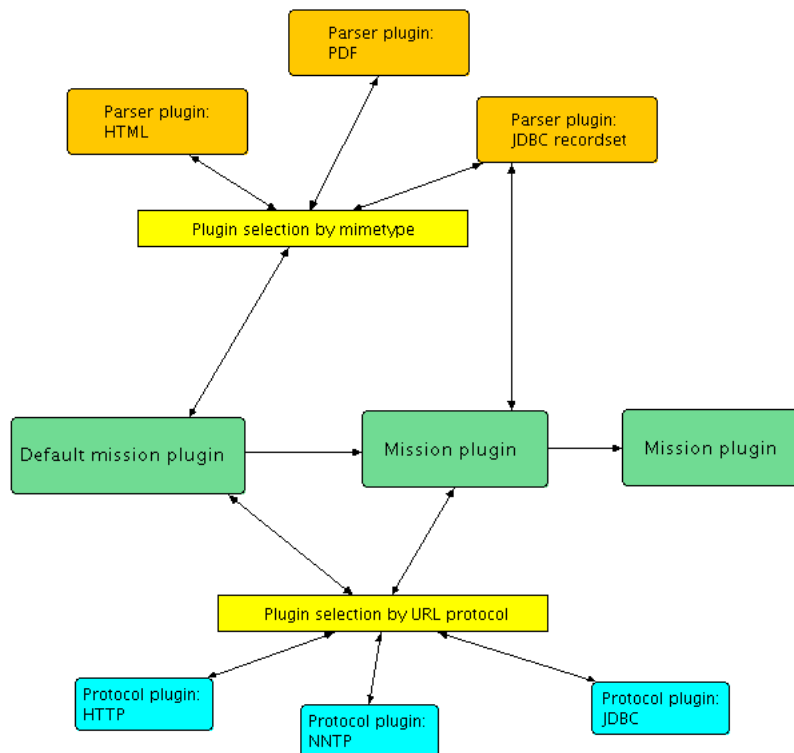
The plugin is also passed a *userdata* field that can be used to store temporary parser states.

Mission plugins

Mission plugins coordinate the retrieval process. They interact with protocol and parser plugins and may alter the latter requests. Each searchbox source has a *pipeline* of mission plugins associated. Searchbox runs the mission plugins in pipeline order, from the first to the last one until all the plugins agree there is no more work to be done.

Another important task for mission plugins is requiring *other* documents to be processed in addition of the original ones so that the *crawling* process can continue. searchbox lets the mission plugins assign a numerical priority for each link, enabling the support for multiple document retrieval strategies.

The round trip



Everything starts when searchbox needs to process a document from a source. It retrieves the mission plugin pipeline for the source and passes the document URL as input to the first mission plugin. Each mission plugin performs its computations, maybe requiring retrieval and/or parsing of other documents and then passes the results on to the next mission plugin in the pipeline. After all the mission plugins ran searchbox checks if some of them required further processing. If it did then the pipeline is fed *the whole output* of the last plugin and a new pipeline processing cycle starts. If no work needs to be done then searchbox retrieves the results and stores them in its index.

Although mission plugins usually schedule work for protocol and parser plugins this is not required; a trivial mission plugin might add a fixed metadata to all the documents it process just to mark them.

2.2 Crawler Results Integrated Database(WP4.2.1: DSI with subcontract)

This component is part of focuseek architecture and it is called DocumentStore. It organized as a multilevel document cache which is able to store the original documents fetched from source in their original format. Such cache can be used in cases where the original source is non always online. Such DocumentStore can be set-up in three different ways:

- to contain only the CRC of the original content
- to contain CRC and all metadata and a pointer to the original content
- to contain CRC, metadata and the content in its original format

The documentstore is accessible through the SOAP interface of the focuseek platform.

The specific data structure to be filled is:

```
struct ArchiveConfiguration
{
    string Name;
    boolean Historicize;
    string AdministrativeContact;
    integer Source;
    integer Auth;
    Frequency SchedulingFrequency;
    integer SchedulingAttribute;
    integer AccessTime;
    integer PagesLimit;
    integer GarbageLimit;
    Cache DocumentCache;
    AccessSpec[] ACL;
}
```

Used to define parameters of an archive, it has the following fields:

- **string Name** - Archive name, used by client.
- **boolean Historicize** - Enable historicization. The value that is initially set with the AddArchive call is retained throughout all the lifecycle of the archive, i.e. you cannot change its value with the SetArchiveConfiguration call.
- **string AdministrativeContact** - Email address of administrative contact.
- **integer Source** - ID of the source configuration used to crawl documents for this archive.
- **integer Auth** - ID of the authentication configuration used to crawl documents for this archive when an authentication type other than AUTH_NONE is specified in the source configuration. Use 0 if AUTH_NONE is used in the source configuration.
- **Frequency SchedulingFrequency** - Base archive automatic refresh frequency.
- **integer SchedulingAttribute** - Base refresh frequency multiplication factor. If the base frequency is HOURLY or DAILY, the multiplication stands for the interval in hours or day between two crawls.
- **integer AccessTime** - Start of crawl. If the base frequency is HOURLY it can have a value between 0 and SchedulingAttribute and sets the time of the first daily crawl, expressed as an offset from 00:00 GMT. If the base frequency is DAILY it can have a value between 0 and 23 and sets the GMT time of crawl.
- **integer PagesLimit** - Maximum number of documents to fetch during a crawl session. "0" mean no limit. When during a crawl this limit is reached the crawling session is terminated.
- **integer GarbageLimit** - Minimum age of a document before garbage collection. "0" mean no garbage collection. The age is expressed in seconds and is measured from the last time the document was fetched.

- **Cache DocumentCache** - Caching level to be used for this archive.
- **AccessSpec[] ACL** - Access control list for this configuration.

Where DocumentCache can be:

- **FULLCACHE** - Retain full document cache
- **CONTEXTCACHE** - Retain minimal cache needed for context extraction
- **NOCACHE** - Don't retain any cache

For a full specification of current focuseek SOAP API see *searchbox 2.0*: Reference Manual

2.3 Collector Engine (WP4.2.1: DSI with subcontract)

Collector/Transcoder Engine will migrate the content from the Crawled Results Integrated Database (created by the Crawler Collector Indexer) to the AXMEDIS format and database. These tools will be capable of processing data automatically updating the content into the AXMEDIS database when these are updated into the CMSs.

The AXCP Rule Engine will be used to execute specific scripted rules on demand by the Focuseek Watch Manager.

These set of scripts implement the Metadata Mapper. They will use the JavaScript to Crawler Interface, the AXOM, and the original content obtained from CMSs to create AXMEDIS Objects. Scripts may use the AXOM Plugins to apply fingerprints and adaptation algorithms on the content. In the case we have to access to the original content stored in the Crawler DB the Fast Access DB Interface can be used instead of the standard SOAP interface.

The Collector Engine will be endowed of a specific interface (Crawler Query Adapter) to receive, from the Query Support, queries for searching into the Crawler Results Integrated Database and sending back the information to the AXMEDIS database manager. In fact, queries submitted to the AXMEDIS database have to be also forwarded (when explicitly requested) into the database of information and content of the CMS which is present in the content factory (content provider, content integrator, etc...). The matched information has to be integrated into the results of the AXMEDIS database general query. The results of the query will not have to include equivalent content that is already available on the AXMEDIS database since it is automatically pushed in by the Collector Engine tool.

This component will be realized using some services already offered by the focuseek platform.

This component will be notified of new contents to be imported in the AXMEDIS DB by a special object called Watch and available from focuseek. A Watch is defined by a set of rules (persistent queries on the index of crawled contents) so that every time a new content match this set of rules the Watch can notify to another component that there is a new content to be imported. This is a native feature of focuseek so that only a configuration task must be performed using the standard Crawler User Interface. The component in charge of this activity is the Watch Manager. It notifies to AXCP Rule Engine through a specific SOAP method used to send the following information:

- the rule id to be applied to create AXMEDIS Objects
- the persistent query contained in the watch as a string, which can be used to retrieve all the objects satisfying the watch query and not only the newest.
- the list of new document ids affected by the watch, that can be used to retrieve the original content from the Crawler Integrated DB.

2.4 JavaScript to Crawler Interface (WP4.2.1: DSI with subcontract)

Module Profile	
Javascript to Crawler Interface	
Executable or Library(Support)	Library
Single Thread or Multithread	Multithread

Language of Development	C++	
Responsible Name	Nicola Baldini	
Responsible Partner	Focuseek	
Status (proposed/approved)	proposed	
Platforms supported	Linux, Windows 2000/XP, Mac OS X	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
Crawler Collector Index		SOAP
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.
N/A		
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

see section 3.9 of Part E

2.5 Database interface with LOBSTER/TAMINO

Module Profile		
Database interface with LOBSTER/TAMINO		
Executable or Library(Support)	Library	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Responsible Name	Nicola Baldini	
Responsible Partner	Focuseek	
Status (proposed/approved)		
Platforms supported	Windows 2000/XP	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
Plug-In Manager		C API
File Formats Used	Shared with	File format name or reference to a section
FFF (Focuseek Flexible Format)		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.

Crawler User Interface		wxWidget 2.5
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
easysoup		LGPL

This is a fetching and rendering Plug-In used to gather and parse contents from Lobster/Tamino XML Database so that it will be interfaced to AXMEDIS like any other Relational database. Lobster system will be accessed using a standard WebService interface by the gathering plug-in. A graphical user interface will be available to configure the gatherer and the XML queries to be used for content extraction from the database.

2.6 Database interface with exported XML content

Module Profile		
Database interface with exported XML content		
Executable or Library(Support)	Library	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Responsible Name	Nicola Baldini	
Responsible Partner	Focuseek	
Status (proposed/approved)		
Platforms supported	Linux, Windows 2000/XP, Mac OS X	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
Plug-In Manager		C API
File Formats Used	Shared with	File format name or reference to a section
FFF (Focuseek Flexible Format)		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Crawler User Interface		WxWidget 2.5
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
Expat		LGPL
Libxml2		LGPL

A rendering Plug-In to gather XML contents exported from an XML database. In this case the gathering protocol is HTTP so only the rendering portion must be implemented. The configuration tool is the same used for the Tamino interface.

2.7 Database interface with ODBC

Module Profile		
Database interface with ODBC		
Executable or Library(Support)	Library	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Responsible Name	Nicola Baldini	
Responsible Partner	Focuseek	
Status (proposed/approved)		
Platforms supported	Linux, Windows 2000/XP, Mac OS X	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
Plug-In Manager		C API
File Formats Used	Shared with	File format name or reference to a section
FFF (Focuseek Flexible Format)		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Crawler User Interface		WxWidget 2.5
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
iODBC		LGPL

Gathering and rendering Plug-In. The gathering section uses the ODBC protocol implemented into almost all Relational Databases in the market. The rendering section will use specific SQL queries to extract required contents from database tables. The configuration will be possible through a graphical user interface.

3 AXMEDIS Database Area (EXITECH, DSI)

AXMEDIS relational DB is a general purpose relational database that can be selected among: Postgres, Mysql, Oracle, DB2 and MSSQL and therefore the system is scalable to fit the need of small end user and of corporate AXMEDIS user.

The AXMEDIS architecture must be independent by such component and must work with different databases.

Since in the database can be stored protected and unprotected objects, it is evident that the security of the unprotected object is not demanded to the database but must be guaranteed by external network equipments such as firewall and other anti intrusion tools that can limit the accesses to the object repository only to privileged network identities.

The requirements collected in the first phase of the project has defined some key feature of the general AXMEDIS Database architecture. The main requirements that have been collected in the analysis phase and that have a strong impact on the technical specification are reported and commented.

1. AXDB have to ***be independent from changes to the AXMEDIS Model structure schema***: this is a strong requirement since it impose that the AXDB schema must be flexible enough to cover changes in the AXMEDIS data model. At the implementation level, as discussed in detail in this chapter, it means that AXDB must match all the requirements of MPEG21 objects, that is all the data that can be stored in a MPEG21 object have to be mapped in the AXDB schema, and that additional metadata have to mapped with a flexible mechanism of couple (fields, value) in order follow the changes in the AXMEDIS model. This point also allow to access to the structure of the AXMEDIS object allowing the possibility of locating, extracting and searching items inside the object, if the object is not encrypted. This is true also for the part regarding the PAR and DRM.
2. AXDB has to ***manage AXMEDIS objects both in binary and xml formats***; this is implemented considering that objects can be external references (binary objects) or BLOB fields in the database capable of fitting objects embedded in XML metadata. For each object a table for taking in account if the object is locked and who owns the lock is created.
3. AXDB has to ***support versioning of AXMEDIS objects***: the versioning mechanism is intended in its simplest form, by tracking the new versions of an AXMEDIS object (given its AXOID) and storing an automatic version number when the object is uploaded to the AXDB. In order to verify if an object is a new version or a copy of an existing version, together with the object must be stored an hash value or a unique fingerprint in order to compare the object to be added with the set of versions of the same object that are already present in the AXDB. This means also that operation such as version recovering and deleting will be possible and that the elimination of an object will erase all its versions in the AXDB. When a new version of an object is uploaded AXEPTOOL has to be notified.
4. AXDB has to ***store administrative information like the transactions performed on the clients as well as the accounting information like licences, contracts etc.***: the database schema has tables that store simple administrative information such as the so called Action-Logs that can be adopted to rebuild accounting information. This means that an Action-Log is a collection of information, detailed in the following, such the object on which the operation is performed (OID), the operation performed with its parameters (if not a simple information, as play, print and so on is stored; for example for streaming also the starting point of the streaming and the duration must be taken in account), who performed the operation (UID), the tool that has been adopted (TID). These information will be available for querying also.
5. AXDB has to ***store the licences sold to the customers***: this means that AXDB has a table with three fields, the user who received the licence, the licence issuer and the licence itself. The possibility to query the licence table will be given in terms of statistic information and not by querying inside the licenses.
6. AXDB has to ***support user management, to set what the user can do in the DB***: the user and group management can be done at different levels. By adopting the user and group capability that each DB has or by implementing an autonomous system for users and group, leaving to the DB the management of low level user access fro administrative purpose or API accessing. Since not all DBs support the same mechanism for user, groups, grant and authorization, it is better to establish API for managing user in a separate manner with respect to the Database users. A special set of tables for

users, for groups and for grants will be created. Since users can be useful also for other parts of the system (such as for Workflow), it is better to create a simplified provisioning system that manages users and groups and authentication for the whole AXMEDIS system, while grants that are specific for the tool or engines will be managed internally. This provisioning system will interface all AXMEDIS tools and engines that need to interact with user and will authenticate users, by using internal authentication or authentication by operating system procedures (Active directory for Windows, Samba or UNIX authentication for Unix like systems, etc). User activity on AXDB have to be monitored in order to track operations.

Grants that each user can have are selected among:

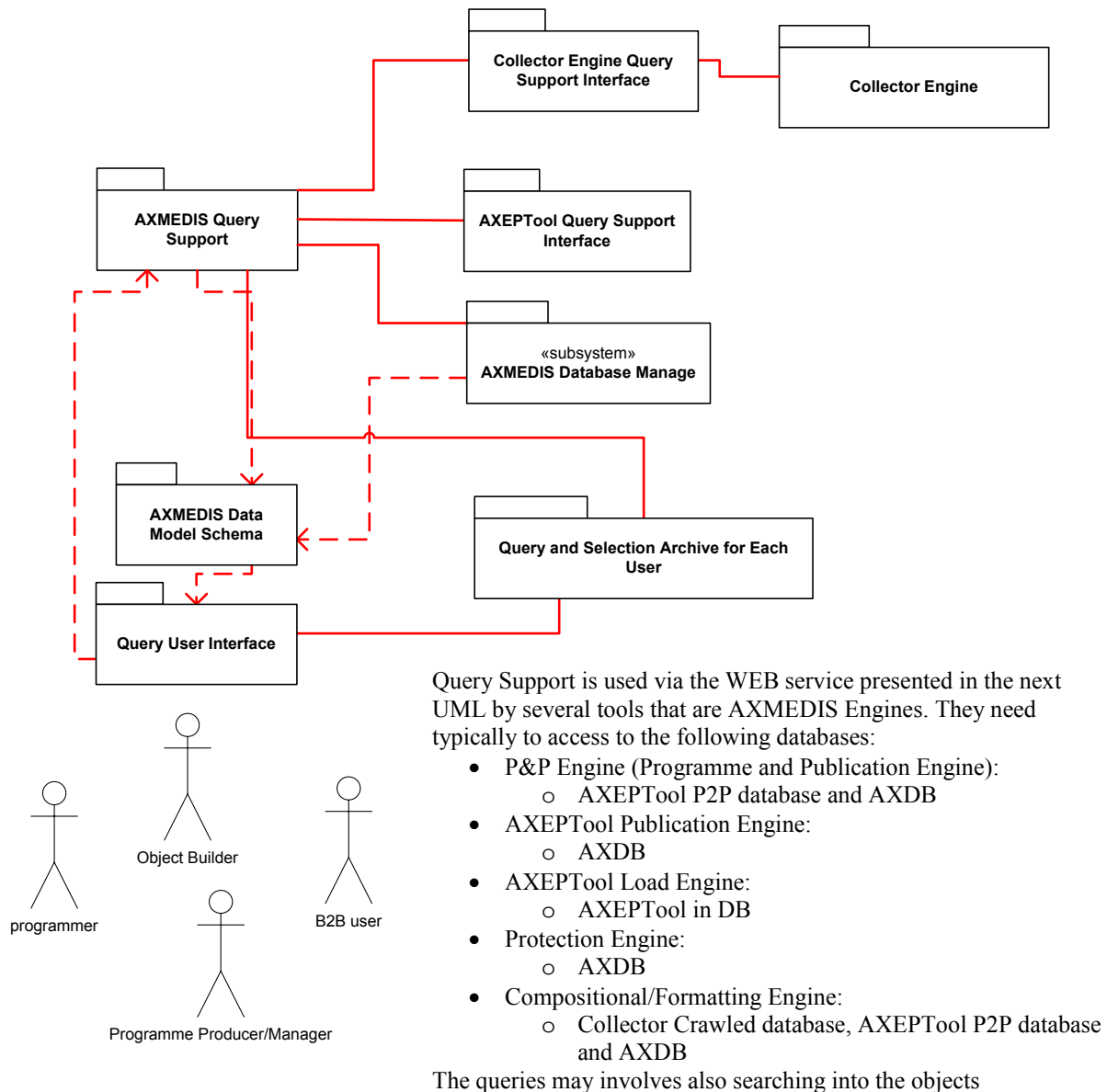
- read object
 - upload new object
 - lock/unlock object
 - upload new version of object
 - remove object
 - change profile of user
 - manage other users
 - create users or group (this functionality is in the provisioning system)
7. ***The database has to provide efficient ways of searching inside the stored objects:*** this means that DB has to provide indexing on the field that are present in MPEG21 format (the core or the system) and full text search in all the other fields. In the first prototype of the system the full text engine of a database will be employed, and in a second step it will be evaluated if a full text search engine must be created over the database structure since not all the DB support full text indexing and the full text engine that non commercial database can provide are limited in the functionalities. Customized full text indexing have to support the following characteristics: (i) list of words to be excluded from indexing, (ii) on-line indexing when the object is uploaded and not off-line indexing in batch, (iii) table for associating words with worded, in order to avoid duplication as much as possible, (iv) table for associating AXOID with the word ID in order to allow section of AXOID that match a word or a set of words. This means that a more efficient indexing will be provided for AXInfo metadata and a less efficient for other metadata not considered in the standard set.
8. ***AXMEDIS Database Administration Tools has to allow to ...:*** this means that a database administration tool shall exist. This tool is a web based tool for searching the AXDB, download for opening all the versions of an AXMEDIS object, manage object and manage user by linking to the provisioning system.

Several data that are contained in the AXDB are needed also in the WFDB, and therefore it is necessary to establish a way to sync the content of AXDB with that of WFDB. Some data are in charge to WFDB and are also present in AXDB: in that case the WFDB must communicate to AXDB any changes in such data; on the contrary, the data that are in charge to AXDB and are used also in the WFDB must be communicated to WFDB as soon as they change.

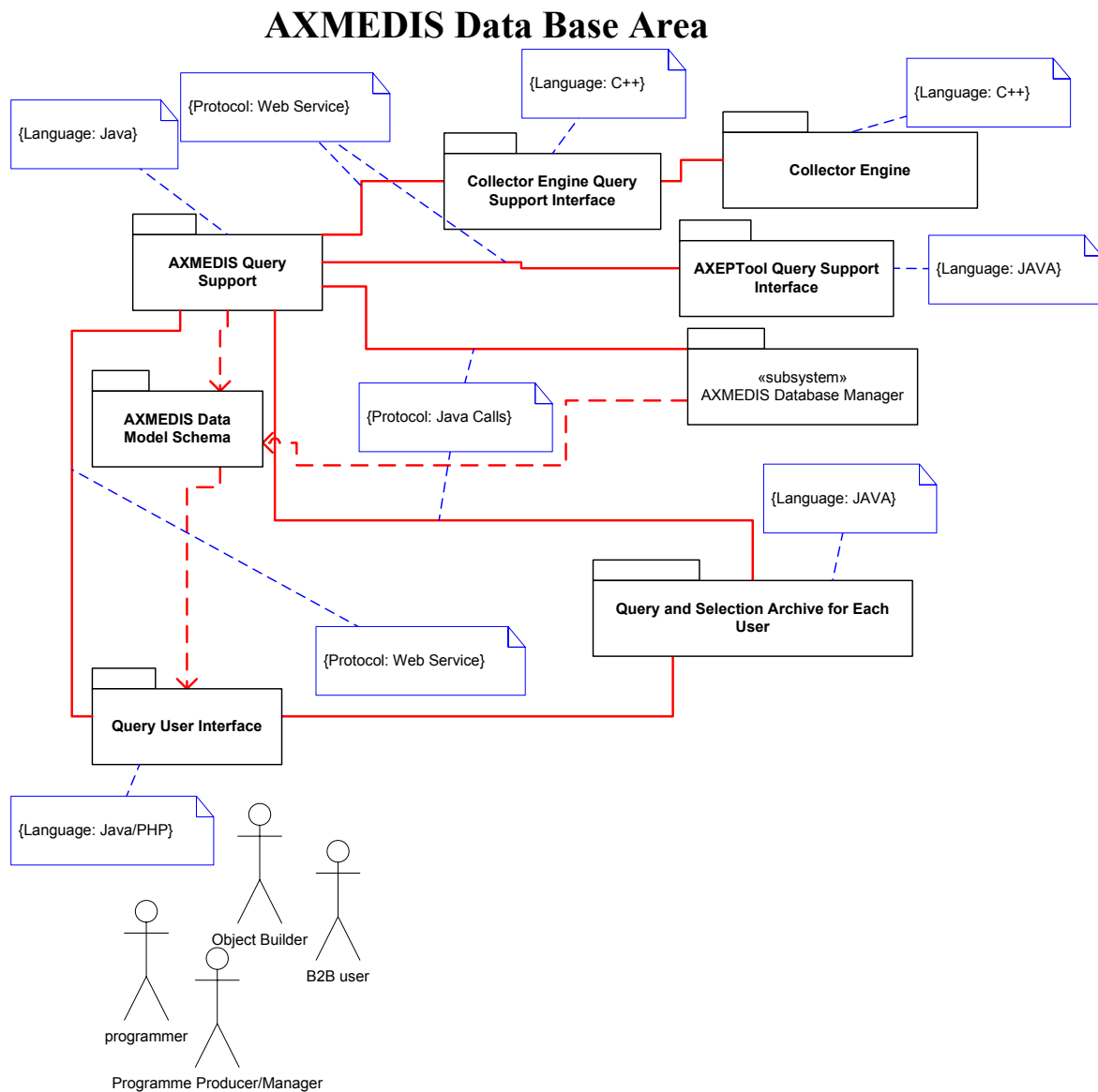
The previous list of commented requirements are the first milestones for the AXDB system specification and will be detailed in the paragraph of the different AXMEDIS components that are present in the AXMEDIS database area.

The AXMEDIS database area includes:

- **AXMEDIS Query Support**
- **AXMEDIS Query User Interface**
- **Collector Engine Query Support Interface**
- **AXEPTool Query Support Interface**
- **AXMEDIS Database Manager**
- **Query Archive for Each User**



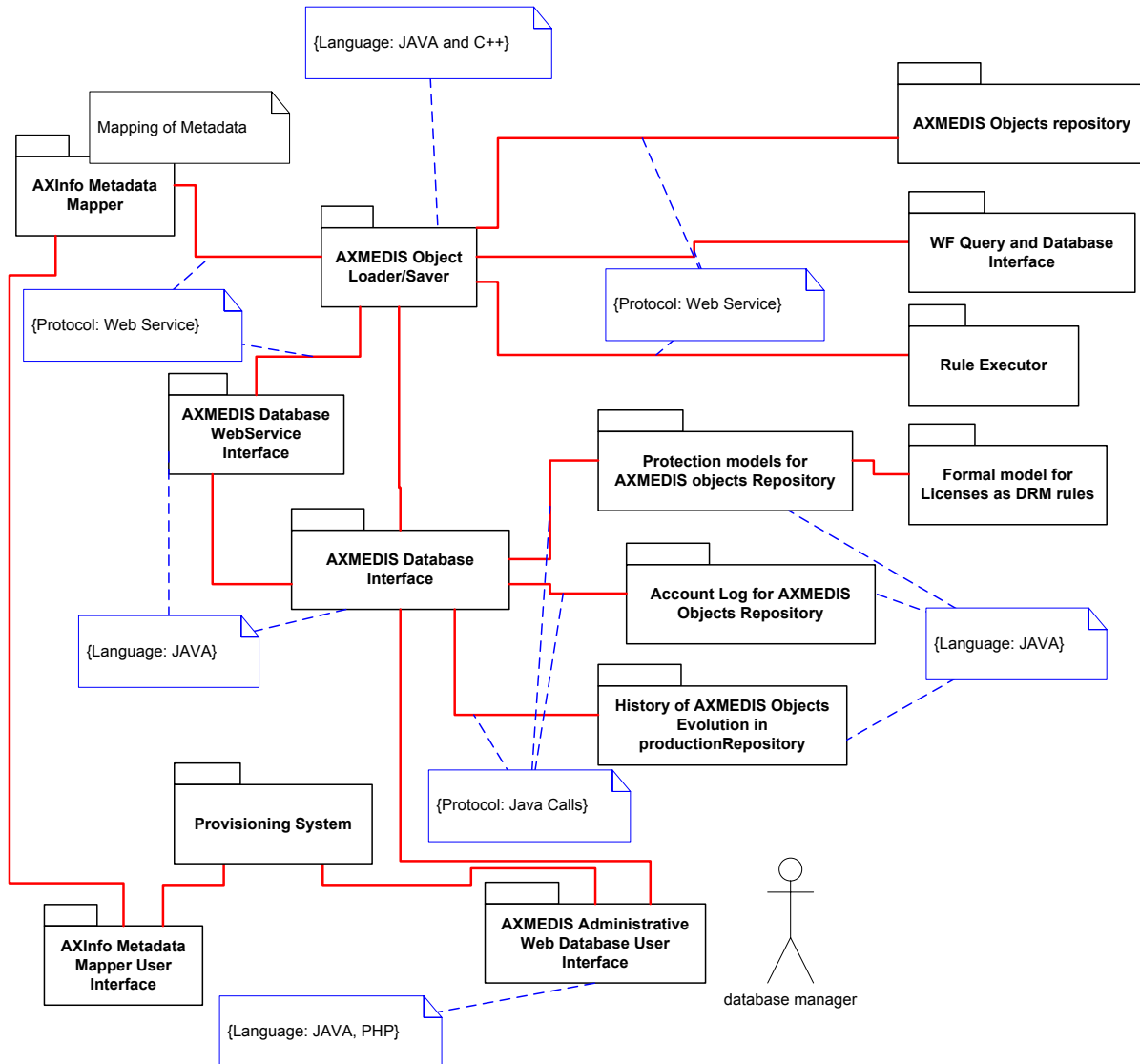
In the following the same diagram with constraints on languages and protocols is reported:



And Internally to the AXMEDIS Database Manager:

- **AXMEDIS Database Interface**
- **AXMEDIS Objects repository**
- **Protection Models for AXMEDIS Objects repository**
- **Account Log for AXMEDIS Objects repository**
- **History of AXMEDIS Objects Evolution in Production Repository**
- **Formal model for Licenses as DRM rules**
- **Provisioning system for managing users and groups**

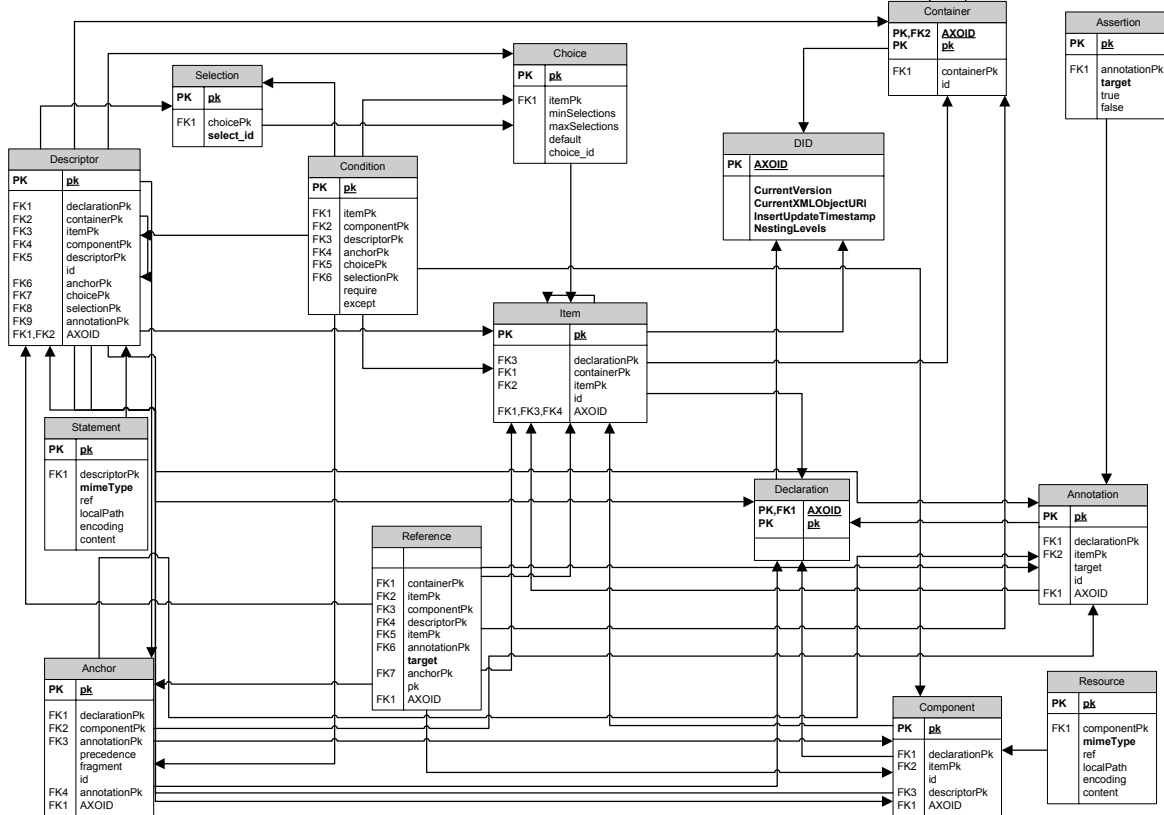
AXMEDIS Database Manager



3.1 AXMEDIS Objects repository Relation Schema (EXITECH, DSI)

This section describes the relation schema for the database related to the digital item description of the MPEG21 standard (PART 2). In the following the Entity Relationship extracted from the MPEG21 PART 2 is reported. This means that an MPEG21 object, relating to the PART2 can be mapped in this ER, without having the possibility to enforce all the bounds of the mpeg21 such as for Declarations for which MPEG21 asserts that at least one among Item, Descriptor, Component, Annotation or Anchor must exist; the ER diagram allow to have one or more Item, Descriptor, Component, Annotation or Anchor referred to the same declarations by the means of a foreign key. The DID table can be enlarged with additional information such as current version and other information useful for recovering the objects of information about the objects.

DID MPEG21 PART2 ER model

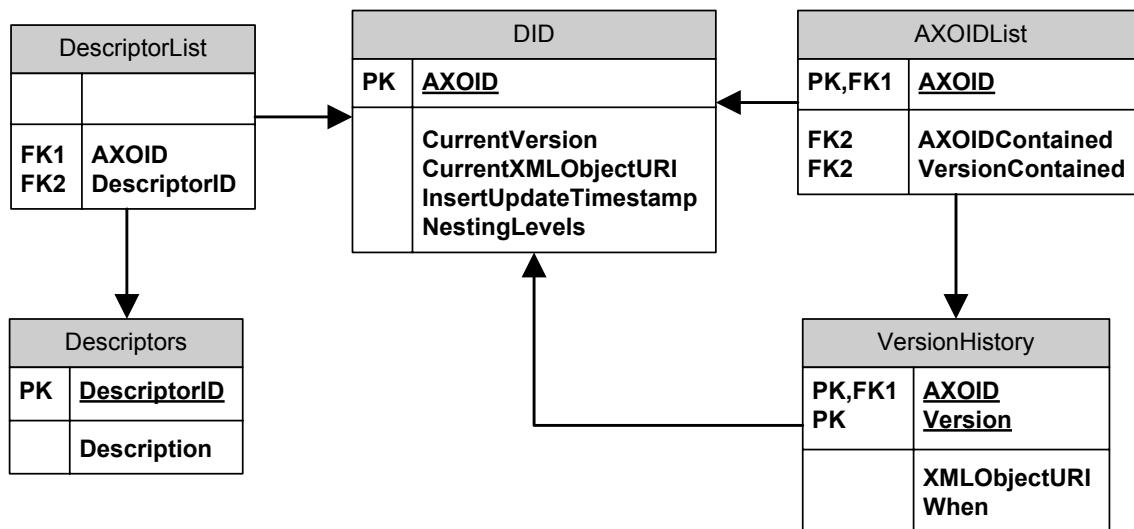


A textual description of the tables with types and relationship among keys is reported in section 3.13.

During the interview of the different partner involved in the project, it has emerged that in this first part of the project it has no sense to have this full description of the object structure according to MPEG21 model, but only some high level information will be needed. To this end, in this first part of the project a very reduced set of information will be implemented. This reduced set is mainly comprised of the list and counting of objects contained in the object under evaluation, the list and number of descriptors present in the object (apart from AXINFO and DCMI that are mandatory and the number of nesting level of the object.

The ER of this database is reported below:

AXMEDIS Object Structure Database



Clear text objects are stored together with all the information that you have in the MPEG21 structure, while protected object for which this structure is not available will be stored only as referenced external objects in the DID table in the CurrentXMLObject field and in the version table that will be described after.

3.2 Descriptors Metadata mapping in Relational Schema (EXITECH, DSI, CRS4)

This section describe the relation schema for the mapping of a fixed number of metadata present in the AXINFO, in a relational schema that have to be flexible to accommodate different metadata for different user and have to take care of multilingual indexing.

The main concept that have to be expressed regarding AXINFO mapping in a relational schema is that it is impossible to reproduce all information that are contained in the AXINFO in the database of the company that is hosting AXMEDIS. This is true for several reasons among whose:

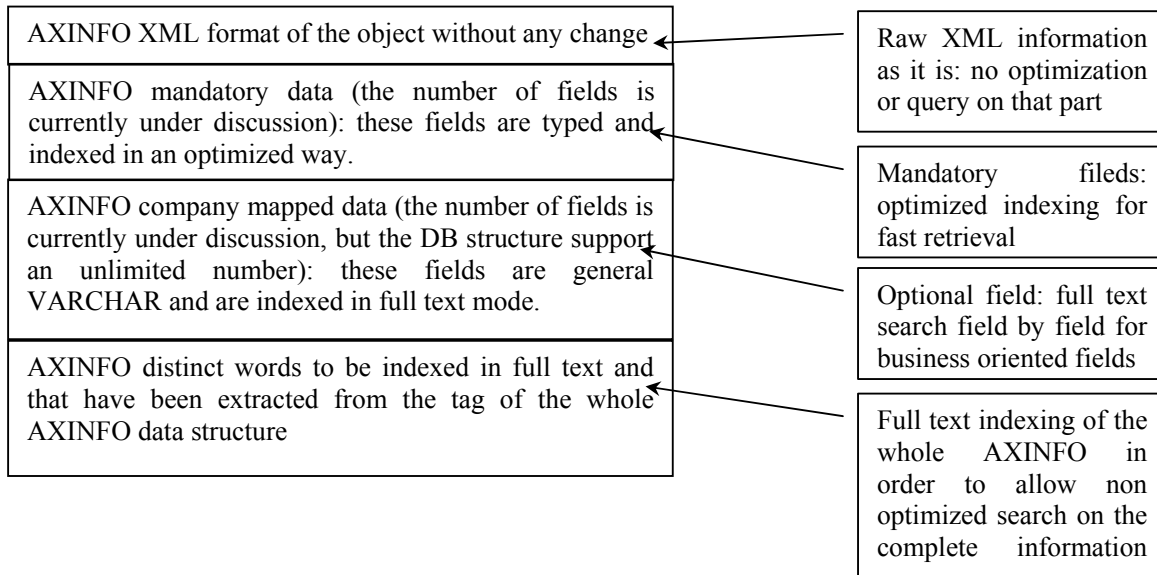
- Only a subset of the whole AXINFO data are in the hand of the company relating to the business activity, business model and activities to be performed on objects;
- The detailed indexing of each fields of the AXINFO is a costly operation to be done for the database and for the previous statement if often unnecessary
- The possibility for the company to have a flexible mapping of AXINFO metadata in an efficient DB is an added value

On the other hand it is not possible to fix in advance the fields that are relevant for each partner apart from a minimum subset of fields (extracted from DUBLIN CORE plus other data relevant). This has been evaluated in the data model, reported in the section A of this specification.

It exists the need to create an additional set of metadata, the model supports any number of fields, that can be mapped by the company at the AXMEDIS initial configuration, so that all items put in the AXMEDIS will have an automatic indexing with respect to the data that are interesting for the business of the company.

Another issue to be addresses is the possibility to index in FULL text the whole content of the AXINFO, that means a full text indexing of the content of all the tags in the AXINFO. This can be implemented by extracting all the distinct words that are present in the AXINFO tags and concatenating these words in order to create a unique fields to be indexed in full text mode.

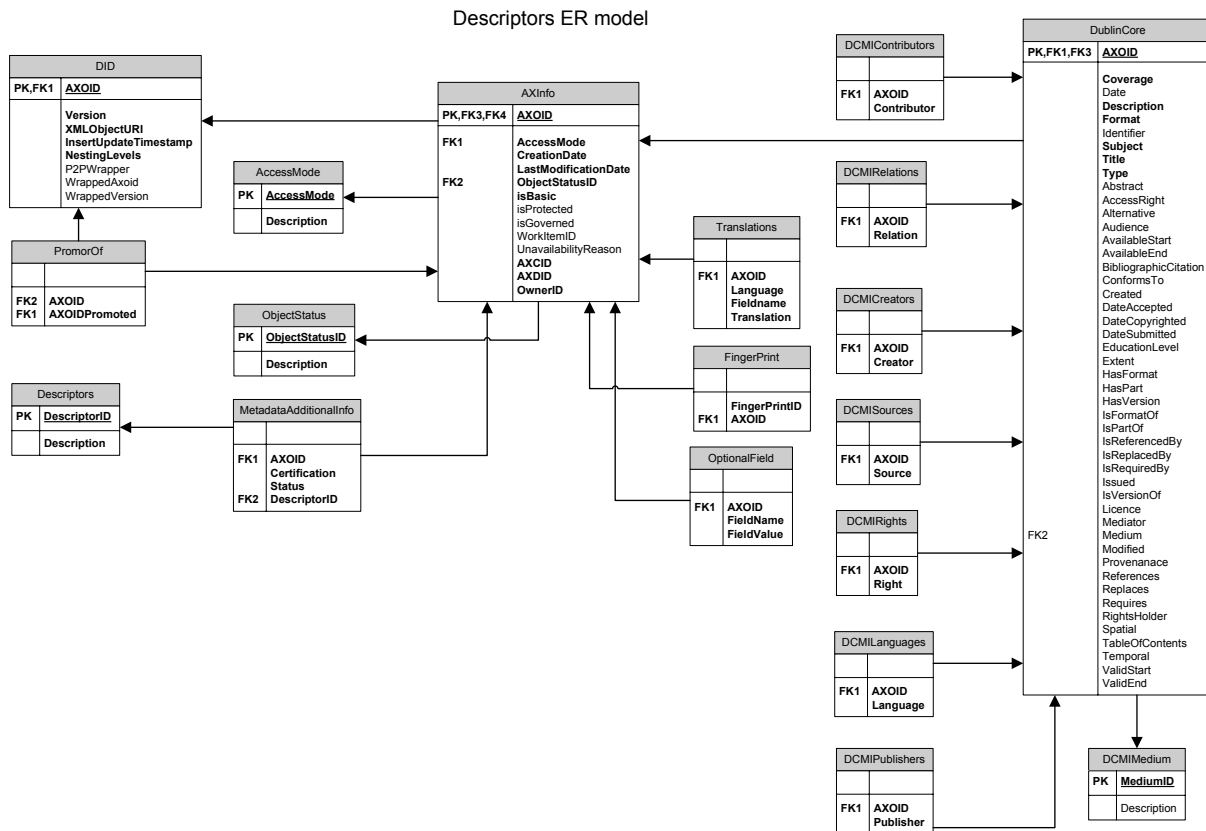
The following picture describes the data to be put in the AXINFO section of the AXDB:



The mandatory fields of the AXInfo contain informations that are managed by AXCS or SuperAXCS and that can be duplicated or not in the AXInfo. For the moment in the AXInfo table only the data necessary to collect other data are reported. Some external tables can be inserted in the model for describing Creator and other parameters or these tables can be not inserted in the database since the information can be extracted from AXCS or SuperAXCS. For the sake of specification the tables will be reported and it will be decided at the implementation phase if a sync mechanism will be established or if the data will be directly asked to other sources that are authoritative for such data.

Some differences can be identified about the organization of the AXInfo in the Data Model and of the AXInfo in the database; the main differences are:

- Version of the object AXInfo/Version in the data model is kept in the DID table and not in the AXinfo table;
- Revision of the object AXInfo/Revision is not stored in the DB since from its specification is evident that once it is uploaded in the DB it must be always 0;
- ObjectType has been implemented as a boolean field isBasic, that is true when the value is BASIC, or false when the object is COMPOSITE;
- PromorOf is implemented with an additional table with a relation with the AXInfo table and another to the DID table in order to identify the Promoted objects;
- AXInfo Workflow information are referenced by the WorkItemID
- InternalPotentialAvailableRights and PotentialAvailableRights are collected together with DRM rules in PMS database. AXDB will ask for information to this db each time it is required to have such informations.
- History is not managed at AXDB level but at the Workflow level.



Apart from the textual description of the tables with types and relationship among keys that is reported in section 3.13, in the next subsections a conceptual description of the Tables meaning is reported.

3.2.1 AXInfo table

This table allows to store all the fields of the AXInfo of the Object in order to have a fast indexing of such mandatory fields. The model is compliant to that reported in Part A of the Specification.

3.2.2 DID

This table is present only because it is referenced by AXInfo and because it contains the information on the version of the object.

3.2.3 AccessMode

In order to reduce duplication and in order to improve search capabilities, a table for standardizing the description of AccessMode has been created. All the different AccessMode will be put there and in the AXInfo table a foreign key links to the correct Access Mode.

List of possible access mode are:

- ReadOnly
- ReadWrite
-

3.2.4 ObjectStatus

This table has been created with the same purpose of AccessMode for standardizing the Status of the Objects.

List of possible Object Status are:

- Published
- In production
- Not Available (in order to manage the situation in which you have an object that is no more available for you, but for which you can also have some older versions that you can still retain).

-

3.2.5 FingerPrint

Since each Axmedis Object can have more than one fingerprint associated, this table is created as a bridge between the Fingerprints and the AXOID. Fingerprints values are not stored in the AXDB, but in the AXCS. They are generated on the fly on the basis of the content and checked against the values in the AXCS for verification.

3.2.6 PromorOf

An Axmedis object can be a Promor of other objects. This table implements this 1:n relationships between other object (DID table) and the current object on the AXINFO.

3.2.7 Translations

It is very difficult to find a suitable way of representing translations for mandatory and optional fields. The solution is general and flexible enough to obtain such objective but is not optimized for searches.

The table has the following fields:

- Languages will be inserted following the ISO 639-2 format standard.;
- Fieldname will be the fully qualified name of the field in dotted notation (DCMISchema.Creator, DublinCore.Mediator, and so on);
- Translation, that is the translation of the Field (all the fields in the main tables are assumed to be in english, while extra languages will be stored here);
- AXOID identify the object to which the translation is referred to.

3.2.8 MetadataAdditionalInfo

This table is useful to implement the 1:n relation between the AXInfo and the certification of multiple metadata descriptors set. For the moment the model implements a number of descriptors to be standardized, but that can be also added at run time. See Descriptor Table for more details.

3.2.9 Descriptors

This table has been created with the same purpose of AccessMode for standardizing the names of the Descriptors of the Object.

List of possible Descriptors are:

- AXINFO
- DCMI
- MPEG7
-

3.2.10 Creator

This table for the moment contains only, the ID of the creator, since it has to be decided if AXCS will keep all information of ID, and therefore this table will disappear, or if such information will be duplicated here and in some way synchronized with AXCS, and therefore this table will contain a lot of fields for describing the creator.

3.2.11 Distributor

Same purpose of Creator.

3.2.12 Owner

Same purpose of Creator.

3.2.13 OptionalField

This table allow the insertion of a variable number of optional fields that can be mapped from the metadata of the object to the database for allowing searches. The table mainly contains a set of couples (Name, Value) related to an AXOID.

3.2.14 DublinCore (DCMI)

This very large table contains the 15 mandatory fields of Dublin Core Metadata Initiative, most of them implemented in terms of lists, as evidenced by the next tables, as:

- Contributor
- Creator

- Language
- Source
- Publisher
- Relation
- Right

The other fields are implemented as single fields such as:

- Coverage
- Date
- Description
- Format (that must be a valid MIME type)
- Identifier
- Subject
- Title
- Type

The Additional or refinement of Dublin Core are also considered as single filed optional parameters, in the sense that for example all BibliographicCitation will be put in a single text field and indexed in the same manner with respect to OptionaFields.

3.2.15 DCMContributors

This table allows to create a list of Contributors for the element Contributor of Dublin Core.

3.2.16 DCMICreators

This table allows to create a list of Creators for the element Creator of Dublin Core.

3.2.17 DCMILanguages

This table allows to create a list of languages for the element language of Dublin Core.

3.2.18 DCMISources

This table allows to create a list of Sources for the element Source of Dublin Core.

3.2.19 DCMIPublishers

This table allows to create a list of Publisheress for the element Publisher of Dublin Core.

3.2.20 DCMIRelations

This table allows to create a list of Relations for the element Relation of Dublin Core.

3.2.21 DCMIRights

This table allows to create a list of Rights for the element Right of Dublin Core.

3.2.22 DCMIMedium

This table has been created with the same purpose of AccessMode for standardizing the names of the Mediums of the Dublin Core.

List of possible Medium are:

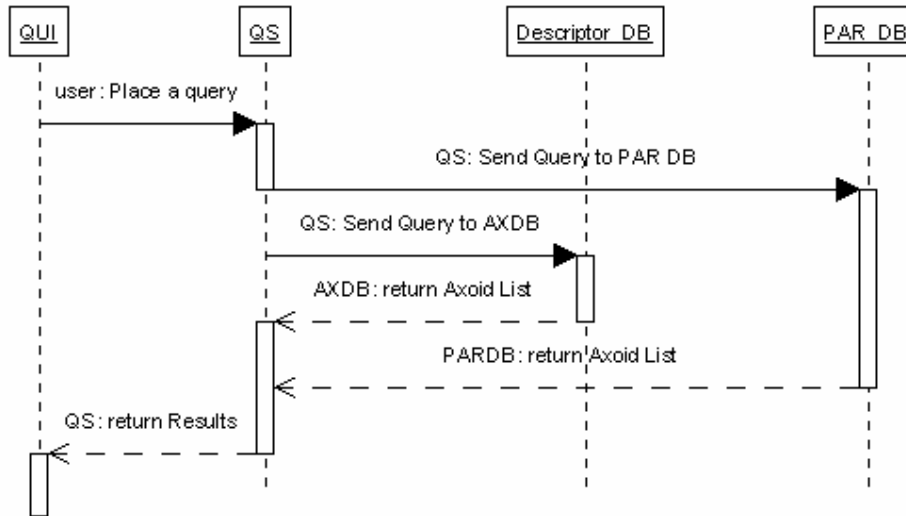
- CD
- DVD
- FILE
-

3.3 Integration between PAR DB and descriptors DB for making queries (EXITECH, FUPF)

The model for resolving the an AXMEDIS query in a global way (and therefore finding all AXOB that satisfy both descriptors and PAR criteria) requires that QS receive the query as is, distribute the query to both Descriptor DB (AXDB Query Support) and PAR DB that should implement a similar interface.

These two subsystems will return to the query support the list of AXOID that match their criteria and therefore and AND operation is performed on the two returned set in order to get only the common AXOID.

The model can be summarized in the following diagram.



The other operation for distribution of queries will be realized in parallel, so that the results of all the sources (P2P, Crawler, other QS) will be merged together with that of AXDB.

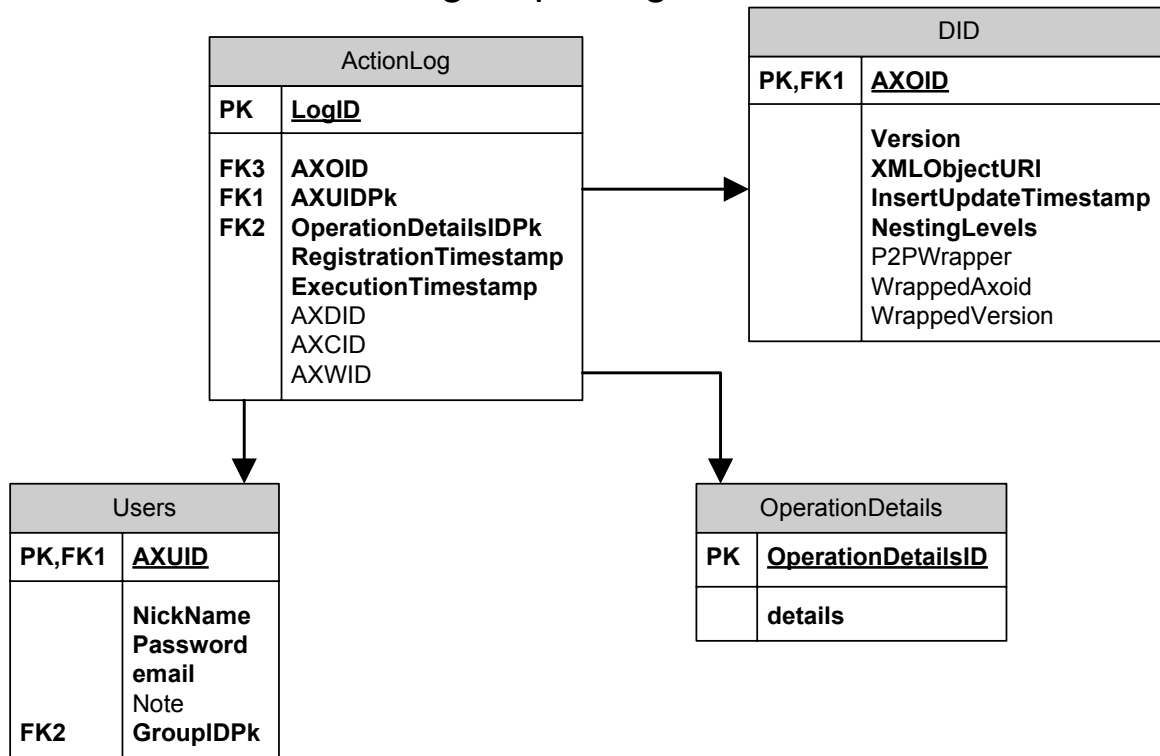
In order to have an easy integration it is necessary that PAR DB implements the same web service interface with respect to AXDBQuerySupport as a server in synchronous manner, so that results are immediately collected by QS.

3.4 Account Log for AXMEDIS Objects repository (EXITECH, FUPF)

The Action-Logs that are the basis for the accounting must be stored in the database by the Core manager and reporting Tool that read such data from AXCS.

The AXCS transfers only the results for which the user is eligible and therefore the information that are stored locally are safe from the point of view of the privacy.

Accounting/Reporting ER model



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.

By now, the structure that have to be defined in more details during the specification of the single work packages is divided (apart from users and DID table that have been discussed yet or will be discussed in the rest of the document and that are reported here only as a reminder for referential integrity) in three main tables:

- ActionLog Table: this table store the Action-Log as it is with some fixed information such as the AXOID on which the operation is performed, the ActorID that performed the operation, the registration timestamp and execution timestamp (that can be different because of off-line operations performed on the objects) also a reference to an external table that will contain the OperationDetails;
- OperationDetails Table: this table will contain the details of the operations such as channel, duration and other details that are to be specified in deep. These details will be taken from the Operation detail table of the AXCS.

A textual description of the tables with types and relationship among keys is reported in section 3.13.

3.5 Database Schema for supporting AXMEDIS (EXITECH, FUPF)

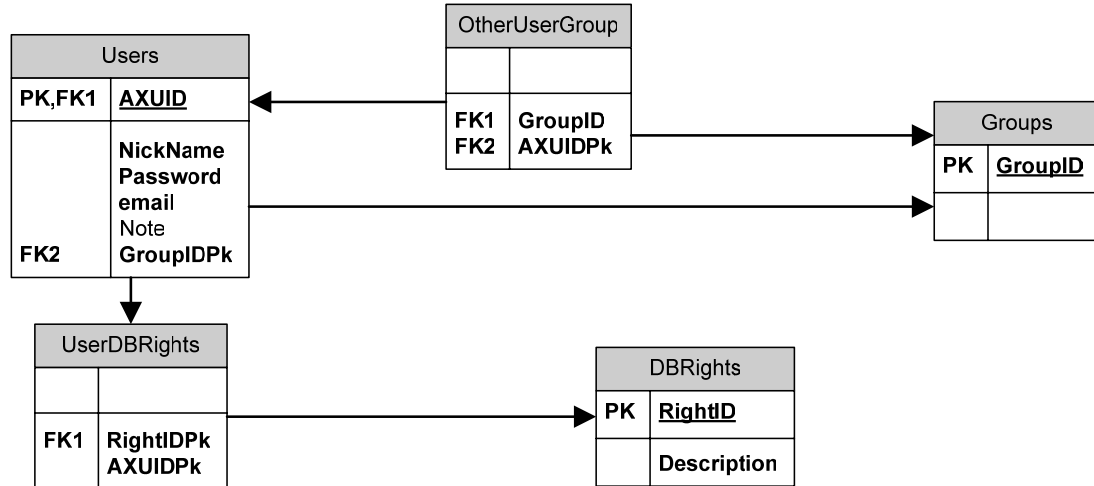
This section will cover all the aspects not covered in the previous sections such as user and group management with rights for users of the AXMEDIS system, Query and Selection Archive for Each User, etc

3.5.1 User and groups

Regarding users a flexible and scalable approach has been followed by implementing a main table with user details (only a few are reported at the moment), main group to which an user belong and additional groups to which the user can optionally be inserted. The group table apart from a description can contain also some other fields to be defined at the moment.

The rights table lists the operation that a user can perform and to each user a set of operations are associated by the means of UserRights table.

User/Groups ER model



For a more detailed and commented view of the ER, a textual representation is reported in section 3.13.

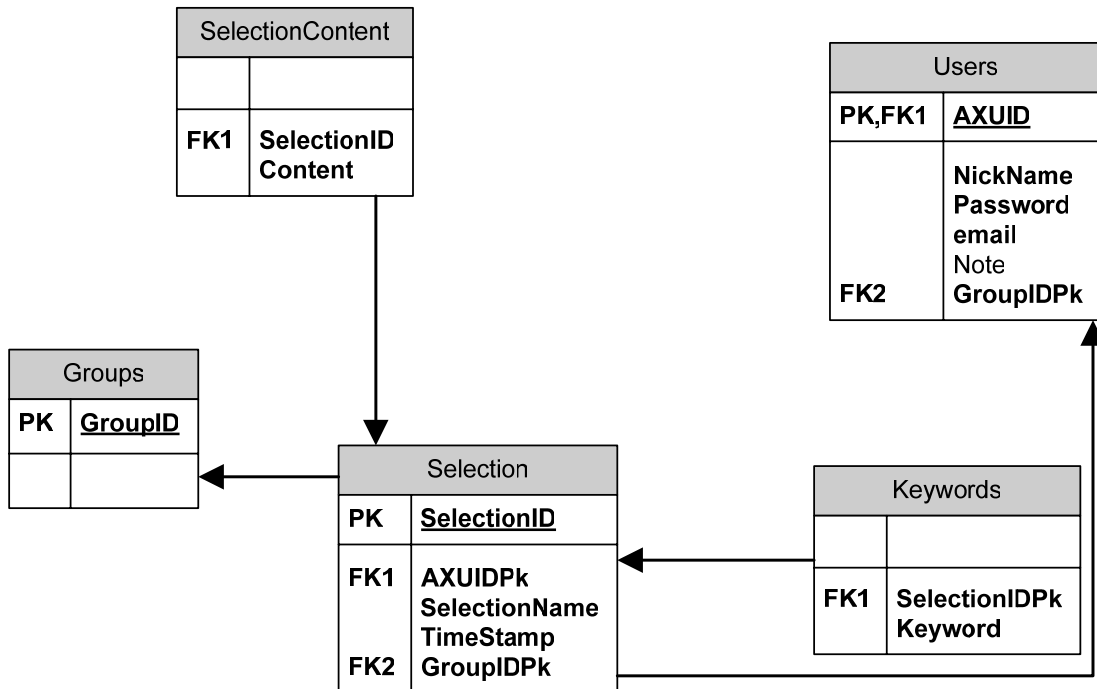
3.5.2 Query and selection archive

In order to support query and selection archive for each user the ER reported below has been created. Since a query can be considered a special case for a selection (a selection with a single symbolic query inside) only selection have been addressed.

The ER is mainly based on two table:

- The selection table contains a reference to the user that has generated it, the name of the selection and the timestamp. This will allow to have different selection with the same name actualized in different moment and therefore with different timestamps. The GroupIDPk field has been inserted with the aim of giving to the selection a visibility to a group of users according to the decision of the selection author
- The SelectionContent table contains the list of items (AXOID or queries) that are contained in the selection

Query/Selection ER model

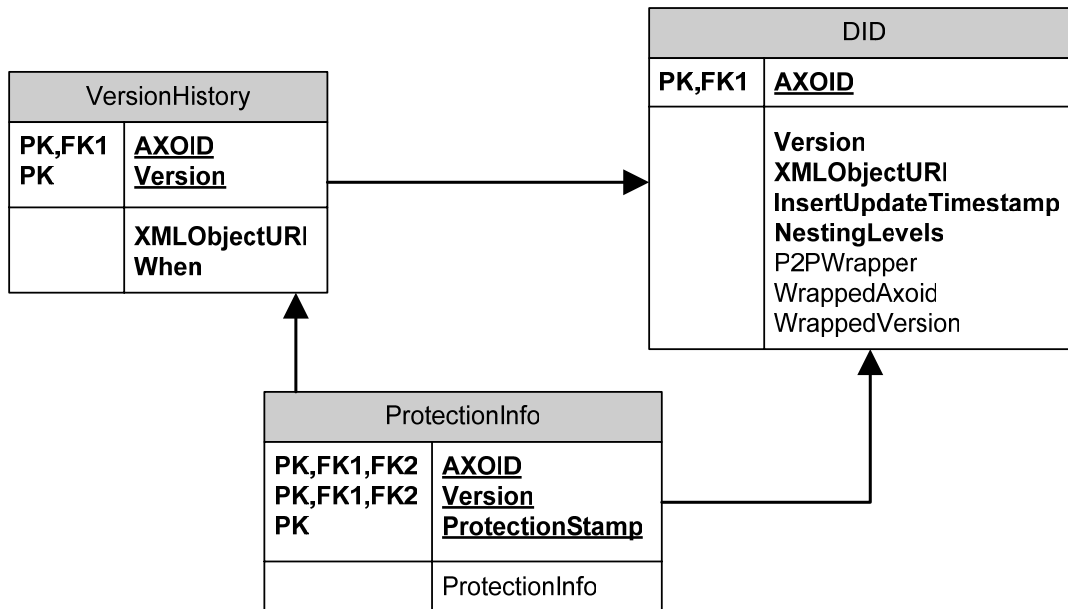


A textual description of the tables with types and relationship among keys is reported in section 3.13.

3.5.3 Version history and Protection Info

A table that contains all the versions of each object is necessary in order to have immediately available the needed object in its native format. If the contents are stored in the object or in the filesystem by the means of the mechanism described in the versioning section, it is enough to store in the VersionHistory table together with the AXOID, the version, the timestamp and the object related to that version that can be recovered from DID table once a check-in of a new version is performed.

Versioning ER model



This database, apart from the already discussed DID table, contains a VersionHistory table that store the URI of each version of the AXOID with the information related to the creation of the version.

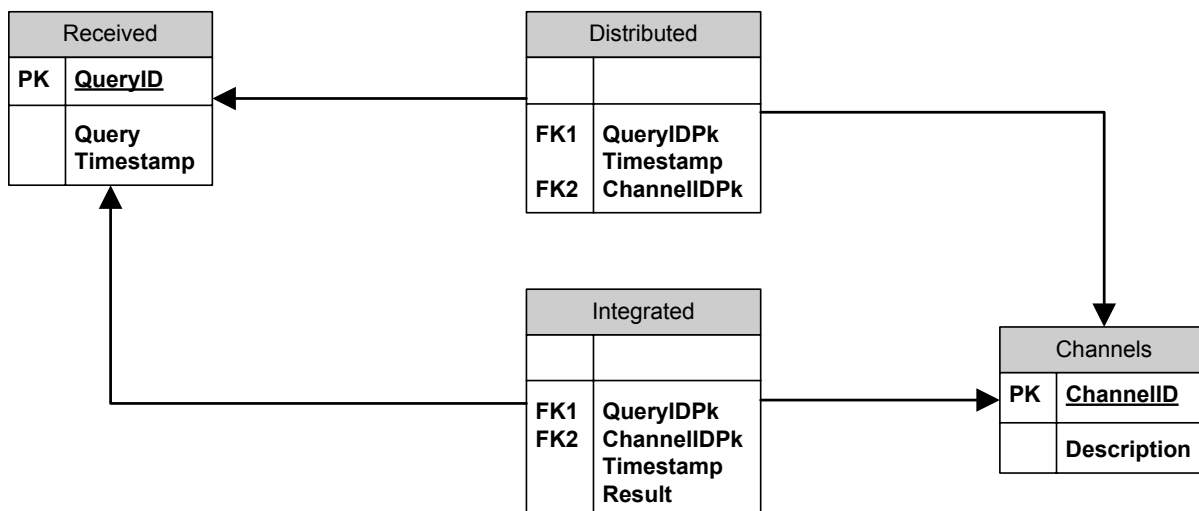
ProtectionInfo is a table that contains the ProtectionStamp that bound the objects to the different protection model that can be applied to the object; ProtectionInfo field is also reported.

A detailed textual description of the tables with types and relationship among keys is reported in section 3.13.

3.5.4 Query Distribution and Integration

In this section the ER diagram that is needed by the Query Support Web Interface for distributing and Integrating query and query results is reported. The description of the table and their meaning is reported in section 4.7.

Query Distribution/Intergration ER model



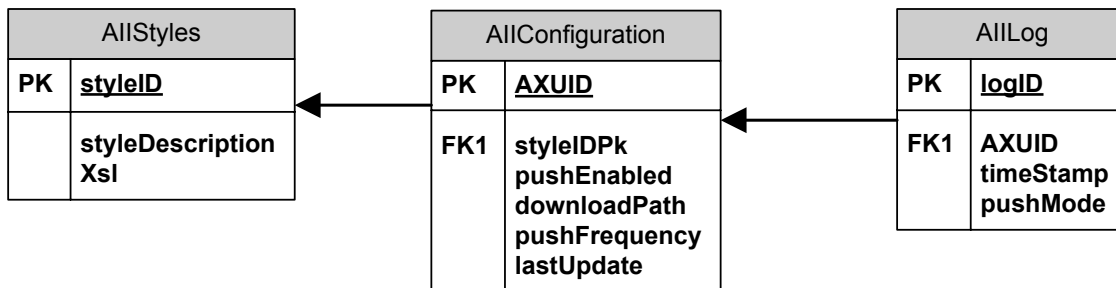
A textual description of the tables with types and relationship among keys is reported in section 3.13.

3.5.5 Administrative Information Integrator

In this section the ER for the simple database that have to support the Administrative Information Intergrator is drawn.

This database is quite simple since it needs only to store the XSLT styles for transforming the logs to the appropriate format for the CMS, in order to support multiple platforms, the configuration of the user preferences and a tagle of logs with all download made by users.

Administrative Information Integrator ER model



3.5.6 P2P Hub Node Support (EXITECH, CRS4)

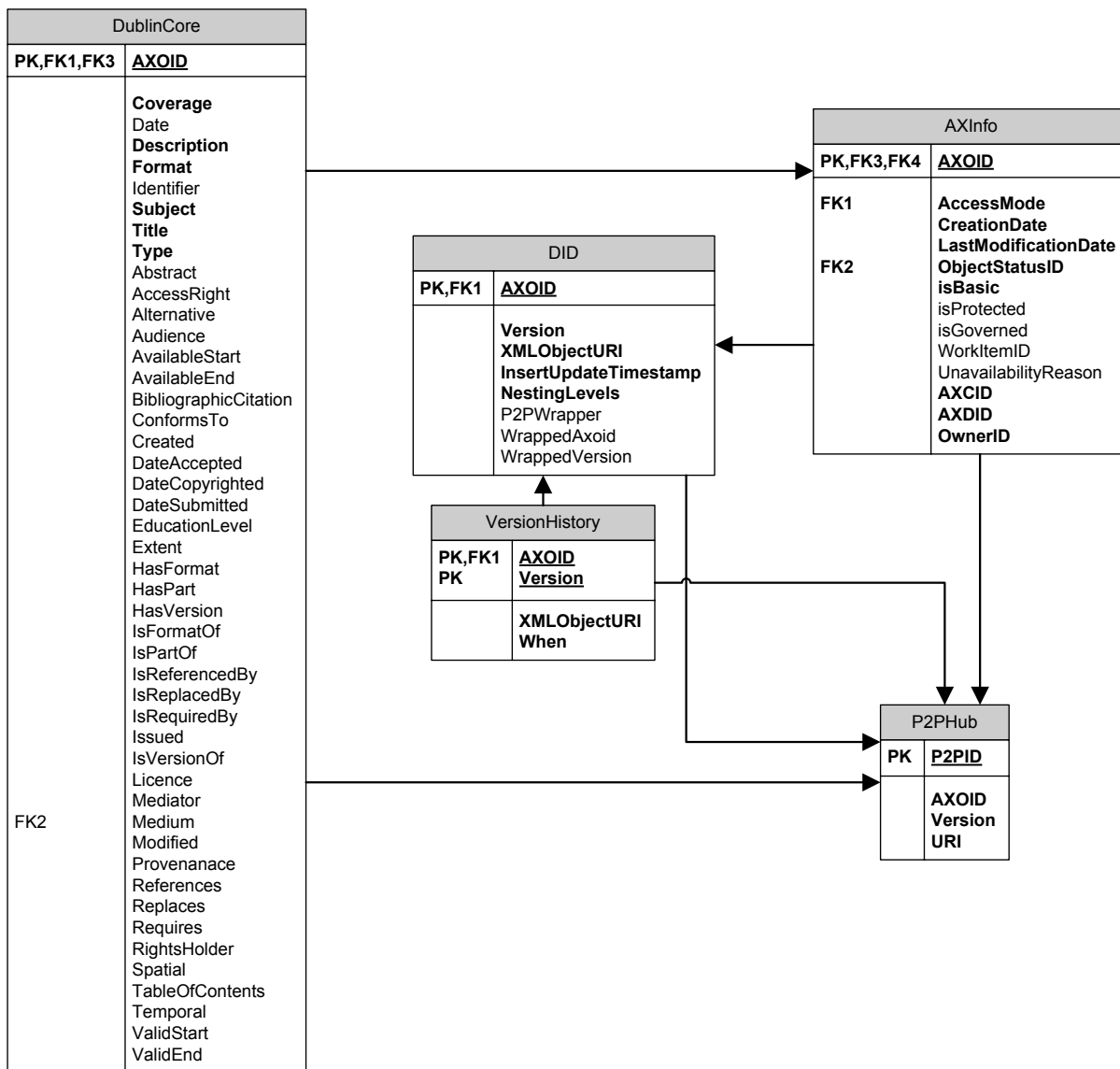
Two different node types exists in the P2P network: leaf nodes and hub nodes. Leaf nodes contains the real objects to be shared, while HUB nodes contains the idexes of the leaves under their specific control. It is unsuitable to pass objects from leaves to hubs, while it is necessary to pass idexes information for performing queries.

Each P2P node contains an instance of AXDB, but unfortunately the database schema is not suitable for addressing the problems of P2P Hub since it is not possible to identify an object in an HUB only with AXOID and version, since several leadf nodes can contain the same object with the same version. It is necessary to distinguish among the same copy of the objects in the P2P network by using also the XMLObjectURI field that can be suitably employed in HUB nodes for storing the peer node that contains the object.

The primary key for P2P hub nodes is the triple (AXOID, Version, URI) that cannot be employed as a general primary key since it propagates in too many tables in unuseful manner for all the other applications that relay on DB.

In order to solve this problem allowing at the same time to adopt QS for making queries also in P2P Hubs, a special table, named P2PHub, has been added for these nodes according to the ER defined below:

P2P Hub Node Support



This table introduces the concept of P2PID that is associated to the triple (AXOID, Version, URI) that represents the object, the version and the location of the object in the P2P network.

P2PID will be used in all the other tables used by Hubs as AXOID, so that when a search will be performed, a list of P2PID will be returned and therefore the HUB node has to perform a query of P2PHub table in order to get from P2PID the AXOID, Version and URI to be returned.

This solution allows, with a minimum overhead to:

- Use the same DB also for P2P Hub nodes
- User the same query engine for P2P Hub nodes
- Index objects in hub nodes without physically transfer the objects

3.6 AXMEDIS Database Interface (EXITECH, DSI)

Module Profile

AXMEDIS Database Interface		
Executable or Library(Support)	Java Package	
Single Thread or Multithread	Multithread	
Language of Development	JAVA	
Responsible Name	Fioravanti	
Responsible Partner	EXITECH	
Status (proposed/approved)	Proposed	
Platforms supported	Windows, Unix like	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
AXMEDIS Object Loader/Saver Core Accounting Manager and reporting tool Account Log for AXMEDIS Objects Repository Protection models for AXMEDIS objects Repository History of AXMEDIS Objects Evolution in production Repository AXMEDIS Administrative Web Database User Interface Protection Tools Inside Engine AXMEDIS Database Webservice interface		
Query Support Webservice Interface		
Query and Selection Archive for Each User		
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.
None	Java package	
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

This module is responsible for giving to the whole system a view of the database that is abstracted by the database that is really employed. This module can be decomposed in an abstraction layer with plugin that can have also different implementations on the basis of the database adopted. One of the plugin that can change in the full text engine that can be different from system to system.

The AXMEDIS Database Interface is comprised of two main modules that are the Database interface and the Web Service database Interface. While the first module can be a JAVA package, the second module is necessary to offer database service to modules that are not implemented in JAVA. To this end part of the functionalities or more advanced functionalities can be put in the WebService part of the module to allow a better integration at the system level.

In this section it is important to define the possible interactions between other parts of the system and the database on the basis of the requested functionalities in the user requirements. In any we have two choices:

- Limit the AXMEDIS Database Interface to a very simple interface capable of executing SQL commands, translating and optimizing them for the specified engine demanding to the external tools the real functionalities toward the other parts of AXMEDIS;
- Create a very complex interface with all the functionalities inside this module having in external modules only the stubs for calling Database Interface;

Since it is very difficult to organize from the beginning all the functionalities offered by the database it is preferable to have a simple interface of the DB that allow to execute simple commands that can be used by the other modules that are part of the real High level interface to the DB.

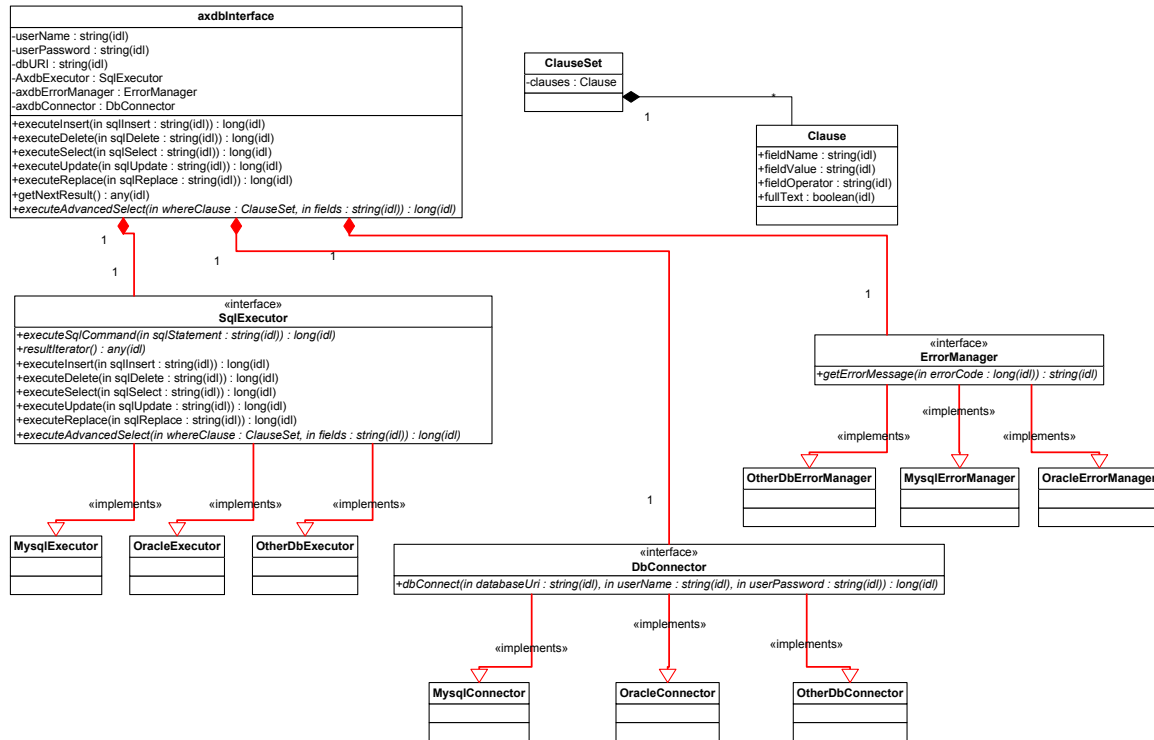
In particular we have:

- AXMEDIS Object Loader/Saver: this module is responsible for inserting, exporting and change versions of AXMEDIS objects;
- Core Accounting Manager and reporting tool: this module is responsible for the storing and retrieving of Action-Logs;
- Query Support Web Service Interface: this service is responsible for all kind of queries;
- Evolution in production Repository: this module is responsible for advanced management of versioning not covered by the Object Loader/Saver, such as analysis of changes in contents, different version of AXINFO and so on;
- Query and Selection Archive for Each User: for getting and saving Selections (and therefore also queries in the personal archive);

Starting from this point of view it can be understood easily that the Database Interface has to provide all back end services for interfacing optimally with the selected database engine and have to expose some simple methods for executing basic I/O and select operation of the DB.

Since this module is a Java package, it is necessary to draw the preliminary class diagram over which, it is possible to define the public methods exposed by the module at least in terms of interfaces.

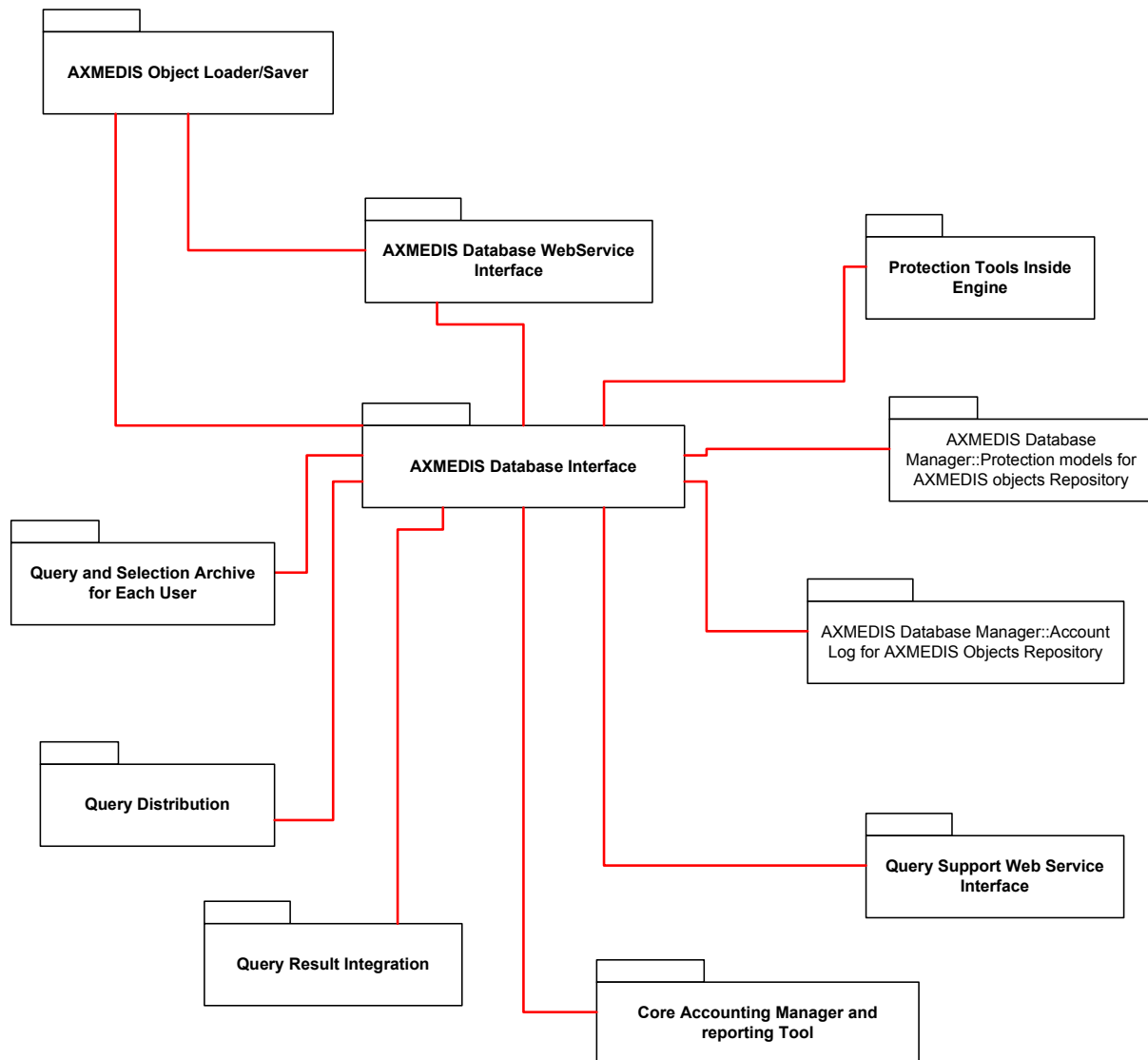
A simplified class diagram for this module is reported below. In the diagram you can easily identify the public methods of AxdbInterface class that are described according to the IDL language.



Fields and DB tables will be standard names that will be mapped on real DB tables, on join of table or on views depending on the organization of the physical database. This simplify the work of the tool since aggregated fields can be created and translated in real db fields. For example the generic field AXINFO can assume the meaning of all AXINFO fields mapped in the DB searched on full text basis.

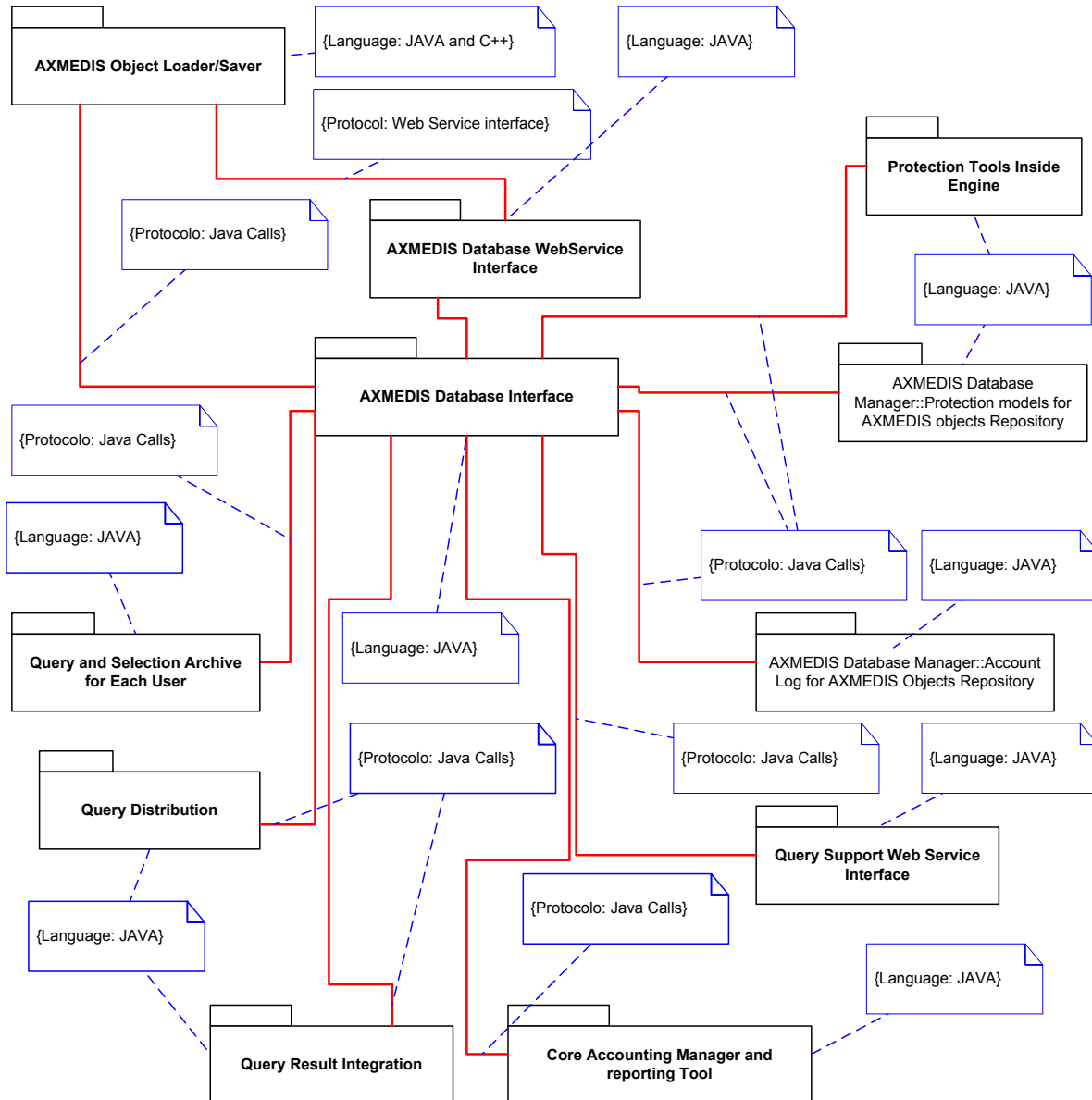
This more advanced way of interfacing the database will result useful especially for SELECT statements, while for insert and delete more direct methods will be provided once identified in the different tools.

The relationships identified by the database Interface and the other parts of the system are reported in the next picture, where on the basis of the development language of each module, the link between the different modules and the database Interface will be moved to the WebService part of the database interface. This will allow to define the level of complexity of the webService and the functionalities that have to provide.



In the following the same diagram with the constraints on languages and protocol is reported.

AXMEDIS Data Base Interface



3.7 AXMEDIS Database WebService Interface (EXITECH, DSI, CRS4)

Module Profile	
AXMEDIS Database WebService Interface	
Executable or Library(Support)	WebService
Single Thread or Multithread	Multithread
Language of Development	JAVA
Responsible Name	Fioravanti

Responsible Partner	EXITECH	
Status (proposed/approved)	Proposed	
Platforms supported	Windows, Unix like	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
AXMEDIS Object Loader/Saver Database Interface Engine and tools using webservices		
File Formats Used	Shared with	File format name or reference to a section
None apart WSDL defined in the text		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
None	WebService for offering services to non Java modules	
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
None		

AXMEDIS Database WebService Interface is a critical module since allow via different methods of several webservices to access to the functionalities that are provided by the database. The list of functionalities will be improved over the time as soon as other engine or tools will require the services of this set of webservices.

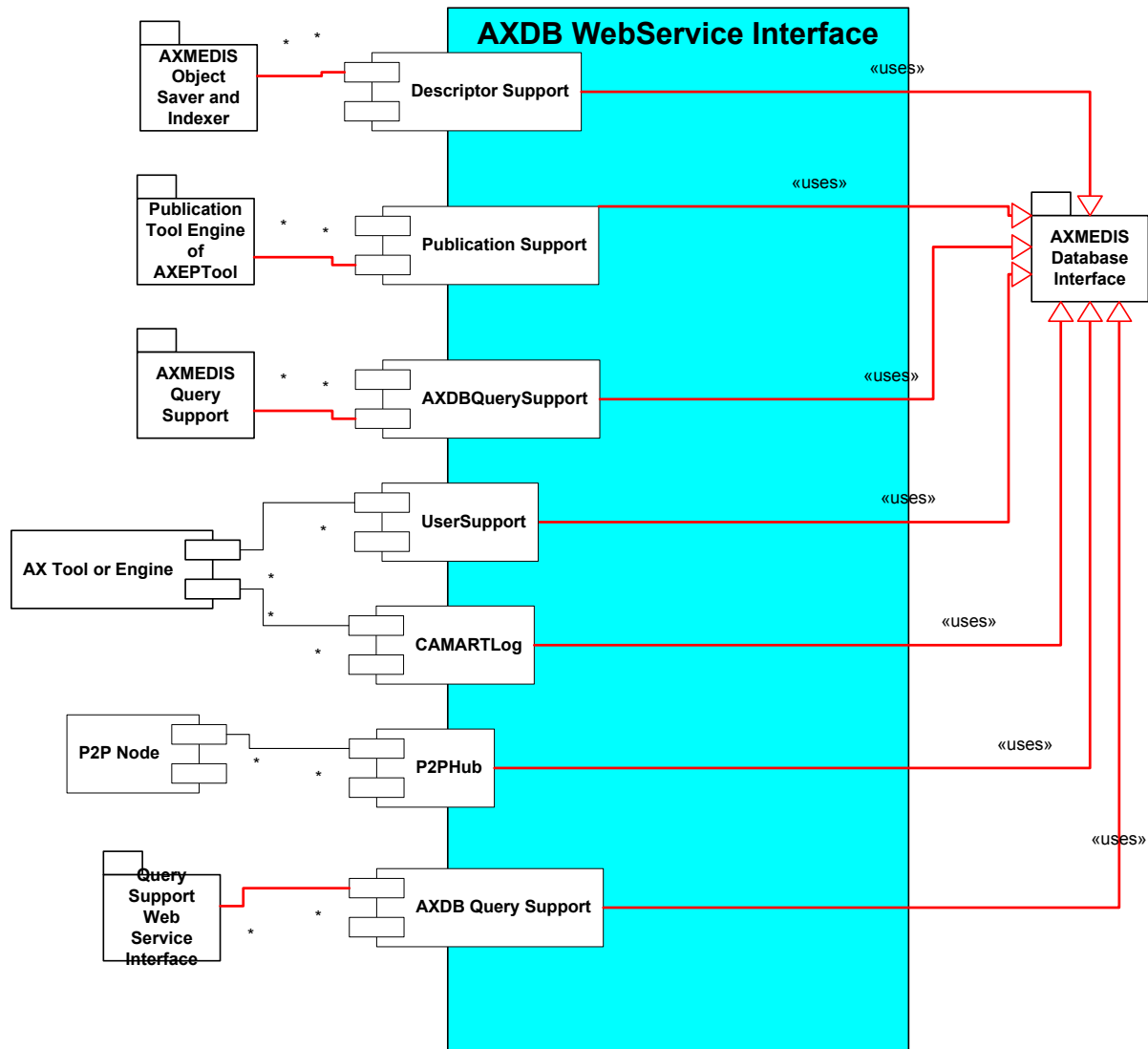
For the moment the following Webservices are defined:

- Descriptor_support webservice that will provide to the saver/indexer module all the functionalities to insert an object in the database or to update an existing one;
- Publication_support that will provide the services needed by the publication engine to recover the AXOID that have been inserted/update after a certain date/time;
- User_support will be provided to verify that if a user is authenticated for a specific operation to be performed;
- CAMART_Support will be provided by the means of a web service, also if the API behind that web service will be available for direct calling by the modules that will be capable of including the Java package;
- P2P_Hub_Support will be provided by the means of a web service capable of performing operations on the P2PHub table, as inserting, updating, deleting, selecting and a more general interface for performing a generic direct SQL query;

In the following the definition of the webservices with the related methods will be detailed, creating a subsection for each webservice and inside the subsection all the methods will be detailed.

Since all the functionality WebService functionalities of the AXDB are exported through this module, it is necessary to create a module that receives the query from the QuerySupport and sends back to it the results. This module is named AXDBQuerySupport and is described in the following.

For this module the following more detailed schema applies.



3.7.1 Descriptor_support (EXITECH, FUPF)

Descriptor_support	
Method	Description
set_Did	This method allows to set all fields of the DID table
set_Axinfo	This method allows to set all the fields of the AXINFO table
add_FingerPrintId	This method add a FingerPrintId record to the table that contains FingerPrintIds
add_PromorOf	This methos add a promor object to the current object
add_Translation	This method add the translation for a field of one of the descriptors for the object
add_OptionalField	This method add an optional couple (field, value) for the object
set_DublinCore	This method set all the DublinCore basic metadata
add_DCMICContributor	This method add a Contributor to the list of contributors of DCMI
add_DCMICreator	This method add a Creator to the list of creators of DCMI
add_DCMILanguage	This method add a Language to the list of languages of DCMI
add_DCMISource	This method add a Source to the list of sources of DCMI
add_DCMIPublisher	This method add a Publisher to the list of publishers of DCMI
add_DCMIRelation	This method add a Relation to the list of relations of DCMI

add_DCMIRight	This method add a Right to the list of rights of DCMI
add_ProtectionInfo	This method allows to add ProtectionInfo for an AXOID with a version and with a predefined TimeStamp
update_ProtectionInfo	This method allows to update a ProtectionInfo record for an AXOID with a version and with a predefined TimeStamp
get_ProtectionInfo	This method allows to get ProtectionInfo for an AXOID with a version and with a predefined TimeStamp
del_ProtectionInfo	This method allows to remove a ProtectionInfo record for an AXOID with a version and with a predefined TimeStamp
clear_AXOID	This methods clears all descriptors for a certain AXOID excluding ProtectionInfo that have to be managed manually

Descriptor_Support	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="Descriptor_support" targetNamespace="http://www.axmedis.org/descriptors.wsdl" xmlns:tns="http://www.axmedis.org/descriptors.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="pi"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="ProtectionInfo" type="xsd:string" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <!-- operation request element --> <element name="Axoid" type="xsd:string"/> <!-- operation request element --> <element name="Version" type="xsd:int"/> <!-- operation request element --> <element name="ObjectLocation" type="xsd:anyURI"/> <!-- operation request element --> <element name="NestingLevels" type="xsd:int"/> <!-- operation request element --> <element name="P2PWrapper" type="xsd:boolean"/> <!-- operation request element --> <element name="WrappedAXOID" type="xsd:string"/> <!-- operation request element --> <element name="WrappedVersion" type="xsd:int"/> <!-- operation response element --> <element name="Result" type="xsd:int"/> </pre>

	<pre> <!-- operation request element --> <element name="Axcid" type="xsd:string"/> <!-- operation request element --> <element name="AccessMode" type="xsd:string"/> <!-- operation request element --> <element name="CreationDate" type="xsd:date"/> <!-- operation request element --> <element name="LastModificationDate" type="xsd:date"/> <!-- operation request element --> <element name="ObjectStatus" type="xsd:string"/> <!-- operation request element --> <element name="isBasic" type="xsd:boolean"/> <!-- operation request element --> <element name="Ownerid" type="xsd:string"/> <!-- operation request element --> <element name="Axdid" type="xsd:string"/> <!-- operation request element --> <element name="isProtected" type="xsd:boolean"/> <!-- operation request element --> <element name="isGoverned" type="xsd:boolean"/> <!-- operation request element --> <element name="WorkItemId" type="xsd:string"/> <!-- operation request element --> <element name="ProtectionStamp" type="xsd:string"/> <!-- operation request element --> <element name="ProtectionInfo" type="xsd:string"/> <!-- operation response element --> <element name="result" type="ax:pi"/> <!-- operation request element --> <element name="FingerPrintId" type="xsd:string"/> <!-- operation request element --> <element name="AxoidPromoted" type="xsd:string"/> <!-- operation request element --> <element name="Language" type="xsd:string"/> <!-- operation request element --> <element name="Fieldname" type="xsd:string"/> <!-- operation request element --> <element name="Translation" type="xsd:string"/> <!-- operation request element --> <element name="FieldName" type="xsd:string"/> <!-- operation request element --> <element name="Fieldvalue" type="xsd:string"/> <!-- operation request element --> <element name="Coverage" type="xsd:string"/> <!-- operation request element --> <element name="Date" type="xsd:string"/> <!-- operation request element --> <element name="Description" type="xsd:string"/> <!-- operation request element --> <element name="Format" type="xsd:string"/> <!-- operation request element --> <element name="Identifier" type="xsd:string"/> <!-- operation request element --> <element name="Subject" type="xsd:string"/> <!-- operation request element --> <element name="Title" type="xsd:string"/> <!-- operation request element --> <element name="Type" type="xsd:string"/> <!-- operation request element --> <element name="Abstract" type="xsd:string"/> <!-- operation request element --> <element name="AccessRight" type="xsd:string"/> <!-- operation request element --> <element name="Alternative" type="xsd:string"/> <!-- operation request element --> <element name="AvailableStart" type="xsd:string"/> <!-- operation request element --> </pre>
--	--

	<pre> <element name="AvailableEnd" type="xsd:string"/> <!-- operation request element --> <element name="BibliographicCitation" type="xsd:string"/> <!-- operation request element --> <element name="ConformsTo" type="xsd:string"/> <!-- operation request element --> <element name="Created" type="xsd:string"/> <!-- operation request element --> <element name="DateAccepted" type="xsd:string"/> <!-- operation request element --> <element name="DateCopyrighted" type="xsd:string"/> <!-- operation request element --> <element name="DateSubmitted" type="xsd:string"/> <!-- operation request element --> <element name="EducationalLevel" type="xsd:string"/> <!-- operation request element --> <element name="Extent" type="xsd:float"/> <!-- operation request element --> <element name="HasFormat" type="xsd:string"/> <!-- operation request element --> <element name="HasPart" type="xsd:string"/> <!-- operation request element --> <element name="HasVersion" type="xsd:string"/> <!-- operation request element --> <element name="IsFormatOf" type="xsd:string"/> <!-- operation request element --> <element name="IsreferencedBy" type="xsd:string"/> <!-- operation request element --> <element name="IsReplacedBy" type="xsd:string"/> <!-- operation request element --> <element name="IsRequiredBy" type="xsd:string"/> <!-- operation request element --> <element name="Issued" type="xsd:string"/> <!-- operation request element --> <element name="IsVersionOf" type="xsd:string"/> <!-- operation request element --> <element name="License" type="xsd:string"/> <!-- operation request element --> <element name="Mediator" type="xsd:string"/> <!-- operation request element --> <element name="Medium" type="xsd:string"/> <!-- operation request element --> <element name="Modified" type="xsd:string"/> <!-- operation request element --> <element name="Provenance" type="xsd:string"/> <!-- operation request element --> <element name="References" type="xsd:string"/> <!-- operation request element --> <element name="replaces" type="xsd:string"/> <!-- operation request element --> <element name="requires" type="xsd:string"/> <!-- operation request element --> <element name="RightsHolder" type="xsd:string"/> <!-- operation request element --> <element name="Spatial" type="xsd:string"/> <!-- operation request element --> <element name="TableOfContents" type="xsd:string"/> <!-- operation request element --> <element name="Temporal" type="xsd:string"/> <!-- operation request element --> <element name="ValidStart" type="xsd:string"/> <!-- operation request element --> <element name="ValidEnd" type="xsd:string"/> <!-- operation request element --> <element name="Contributor" type="xsd:string"/> <!-- operation request element --> <element name="Creator" type="xsd:string"/> </pre>
--	--

```

<!-- operation request element -->
<element name="Source" type="xsd:string"/>
<!-- operation request element -->
<element name="Publisher" type="xsd:string"/>
<!-- operation request element -->
<element name="Relation" type="xsd:string"/>
<!-- operation request element -->
<element name="Right" type="xsd:string"/>
</schema>

</types>

<message name="set-DidRequest">
<part name="Axoid" element="ax:Axoid"/>
<part name="Version" element="ax:Version"/>
<part name="ObjectLocation" element="ax:ObjectLocation"/>
<part name="NestingLevels" element="ax:NestingLevels"/>
<part name="P2PWrapper" element="ax:P2PWrapper"/>
<part name="WrappedAXOID" element="ax:WrappedAXOID"/>
<part name="WrappedVersion" element="ax:WrappedVersion"/>
</message>

<message name="set-DidResponse">
<part name="Result" element="ax:Result"/>
</message>

<message name="set-AxinfoRequest">
<part name="Axoid" element="ax:Axoid"/>
<part name="Axcid" element="ax:Axcid"/>
<part name="AccessMode" element="ax:AccessMode"/>
<part name="CreationDate" element="ax:CreationDate"/>
<part name="LastModificationDate" element="ax:LastModificationDate"/>
<part name="ObjectStatus" element="ax:ObjectStatus"/>
<part name="isBasic" element="ax:isBasic"/>
<part name="Ownerid" element="ax:Ownerid"/>
<part name="Axdid" element="ax:Axdid"/>
<part name="isProtected" element="ax:isProtected"/>
<part name="isGoverned" element="ax:isGoverned"/>
<part name="WorkItemId" element="ax:WorkItemId"/>
</message>

<message name="set-AxinfoResponse">
<part name="Result" element="ax:Result"/>
</message>

<message name="add-ProtectionInfoRequest">
<part name="Axoid" element="ax:Axoid"/>
<part name="Version" element="ax:Version"/>
<part name="ProtectionStamp" element="ax:ProtectionStamp"/>
<part name="ProtectionInfo" element="ax:ProtectionInfo"/>
</message>

<message name="add-ProtectionInfoResponse">
<part name="Result" element="ax:Result"/>
</message>

<message name="update-ProtectionInfoRequest">
<part name="Axoid" element="ax:Axoid"/>
<part name="Version" element="ax:Version"/>
<part name="ProtectionStamp" element="ax:ProtectionStamp"/>
<part name="ProtectionInfo" element="ax:ProtectionInfo"/>
</message>

<message name="update-ProtectionInfoResponse">
<part name="Result" element="ax:Result"/>
</message>

```

	<pre> <message name="del-ProtectionInfoRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Version" element="ax:Version"/> <part name="ProtectionStamp" element="ax:ProtectionStamp"/> </message> <message name="del-ProtectionInfoResponse"> <part name="Result" element="ax:Result"/> </message> <message name="get-ProtectionInfoRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Version" element="ax:Version"/> <part name="ProtectionStamp" element="ax:ProtectionStamp"/> </message> <message name="protectioninfo"> <part name="result" element="ax:result"/> </message> <message name="clear-AXOIDRequest"> <part name="Axoid" element="ax:Axoid"/> </message> <message name="clear-AXOIDResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-FingerPrintIdRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="FingerPrintId" element="ax:FingerPrintId"/> </message> <message name="add-FingerPrintIdResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-PromorOfRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="AxoidPromoted" element="ax:AxoidPromoted"/> </message> <message name="add-PromorOfResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-TranslationRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Language" element="ax:Language"/> <part name="Fieldname" element="ax:Fieldname"/> <part name="Translation" element="ax:Translation"/> </message> <message name="add-TranslationResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-OptionalFieldRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="FieldName" element="ax:FieldName"/> <part name="Fieldvalue" element="ax:Fieldvalue"/> </message> <message name="add-OptionalFieldResponse"> <part name="Result" element="ax:Result"/> </message> </pre>
--	--

```

<message name="set-DublinCoreRequest">
  <part name="Axoid" element="ax:Axoid"/>
  <part name="Coverage" element="ax:Coverage"/>
  <part name="Date" element="ax:Date"/>
  <part name="Description" element="ax:Description"/>
  <part name="Format" element="ax:Format"/>
  <part name="Identifier" element="ax:Identifier"/>
  <part name="Subject" element="ax:Subject"/>
  <part name="Title" element="ax:Title"/>
  <part name="Type" element="ax:Type"/>
  <part name="Abstract" element="ax:Abstract"/>
  <part name="AccessRight" element="ax:AccessRight"/>
  <part name="Alternative" element="ax:Alternative"/>
  <part name="AvailableStart" element="ax:AvailableStart"/>
  <part name="AvailableEnd" element="ax:AvailableEnd"/>
  <part name="BibliographicCitation" element="ax:BibliographicCitation"/>
  <part name="ConformsTo" element="ax:ConformsTo"/>
  <part name="Created" element="ax:Created"/>
  <part name="DateAccepted" element="ax:DateAccepted"/>
  <part name="DateCopyrighted" element="ax:DateCopyrighted"/>
  <part name="DateSubmitted" element="ax:DateSubmitted"/>
  <part name="EducationalLevel" element="ax:EducationalLevel"/>
  <part name="Extent" element="ax:Extent"/>
  <part name="HasFormat" element="ax:HasFormat"/>
  <part name="HasPart" element="ax:HasPart"/>
  <part name="HasVersion" element="ax:HasVersion"/>
  <part name="IsFormatOf" element="ax:IsFormatOf"/>
  <part name="IsreferencedBy" element="ax:IsreferencedBy"/>
  <part name="IsReplacedBy" element="ax:IsReplacedBy"/>
  <part name="IsRequiredBy" element="ax:IsRequiredBy"/>
  <part name="Issued" element="ax:Issued"/>
  <part name="IsVersionOf" element="ax:IsVersionOf"/>
  <part name="License" element="ax:License"/>
  <part name="Mediator" element="ax:Mediator"/>
  <part name="Medium" element="ax:Medium"/>
  <part name="Modified" element="ax:Modified"/>
  <part name="Provenance" element="ax:Provenance"/>
  <part name="References" element="ax:References"/>
  <part name="replaces" element="ax:replaces"/>
  <part name="requires" element="ax:requires"/>
  <part name="RightsHolder" element="ax:RightsHolder"/>
  <part name="Spatial" element="ax:Spatial"/>
  <part name="TableOfContents" element="ax:TableOfContents"/>
  <part name="Temporal" element="ax:Temporal"/>
  <part name="ValidStart" element="ax:ValidStart"/>
  <part name="ValidEnd" element="ax:ValidEnd"/>
</message>

<message name="set-DublinCoreResponse">
  <part name="Result" element="ax:Result"/>
</message>

<message name="add-DCMICContributorRequest">
  <part name="Axoid" element="ax:Axoid"/>
  <part name="Contributor" element="ax:Contributor"/>
</message>

<message name="add-DCMICContributorResponse">
  <part name="Result" element="ax:Result"/>
</message>

<message name="add-DCMICreatorRequest">
  <part name="Axoid" element="ax:Axoid"/>
  <part name="Creator" element="ax:Creator"/>
</message>

<message name="add-DCMICreatorResponse">

```

	<pre> <part name="Result" element="ax:Result"/> </message> <message name="add-DCMILanguageRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Language" element="ax:Language"/> </message> <message name="add-DCMILanguageResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-DCMISourceRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Source" element="ax:Source"/> </message> <message name="add-DCMISourceResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-DCMIPublisherRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Publisher" element="ax:Publisher"/> </message> <message name="add-DCMIPublisherResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-DCMIRelationRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Relation" element="ax:Relation"/> </message> <message name="add-DCMIRelationResponse"> <part name="Result" element="ax:Result"/> </message> <message name="add-DCMIRightRequest"> <part name="Axoid" element="ax:Axoid"/> <part name="Right" element="ax:Right"/> </message> <message name="add-DCMIRightResponse"> <part name="Result" element="ax:Result"/> </message> <portType name="Descriptor_supportPortType"> <operation name="set-Did"> <documentation>Service definition of function ax__set_Did</documentation> <input message="tns:set-DidRequest"/> <output message="tns:set-DidResponse"/> </operation> <operation name="set-Axinfo"> <documentation>Service definition of function ax__set_Axinfo</documentation> <input message="tns:set-AxinfoRequest"/> <output message="tns:set-AxinfoResponse"/> </operation> <operation name="add-ProtectionInfo"> <documentation>Service definition of function ax__add_ProtectionInfo</documentation> <input message="tns:add-ProtectionInfoRequest"/> <output message="tns:add-ProtectionInfoResponse"/> </operation> <operation name="update-ProtectionInfo"> <documentation>Service definition of function ax__update_ProtectionInfo</documentation> <input message="tns:update-ProtectionInfoRequest"/> </pre>
--	---

	<pre> <output message="tns:update-ProtectionInfoResponse"/> </operation> <operation name="del-ProtectionInfo"> <documentation>Service definition of function ax__del_ProtectionInfo</documentation> <input message="tns:del-ProtectionInfoRequest"/> <output message="tns:del-ProtectionInfoResponse"/> </operation> <operation name="get-ProtectionInfo"> <documentation>Service definition of function ax__get_ProtectionInfo</documentation> <input message="tns:get-ProtectionInfoRequest"/> <output message="tns:protectioninfo"/> </operation> <operation name="clear-AXOID"> <documentation>Service definition of function ax__clear_AXOID</documentation> <input message="tns:clear-AXOIDRequest"/> <output message="tns:clear-AXOIDResponse"/> </operation> <operation name="add-FingerPrintId"> <documentation>Service definition of function ax__add_FingerPrintId</documentation> <input message="tns:add-FingerPrintIdRequest"/> <output message="tns:add-FingerPrintIdResponse"/> </operation> <operation name="add-PromorOf"> <documentation>Service definition of function ax__add_PromorOf</documentation> <input message="tns:add-PromorOfRequest"/> <output message="tns:add-PromorOfResponse"/> </operation> <operation name="add-Translation"> <documentation>Service definition of function ax__add_Translation</documentation> <input message="tns:add-TranslationRequest"/> <output message="tns:add-TranslationResponse"/> </operation> <operation name="add-OptionalField"> <documentation>Service definition of function ax__add_OptionalField</documentation> <input message="tns:add-OptionalFieldRequest"/> <output message="tns:add-OptionalFieldResponse"/> </operation> <operation name="set-DublinCore"> <documentation>Service definition of function ax__set_DublinCore</documentation> <input message="tns:set-DublinCoreRequest"/> <output message="tns:set-DublinCoreResponse"/> </operation> <operation name="add-DCMICContributor"> <documentation>Service definition of function ax__add_DCMICContributor</documentation> <input message="tns:add-DCMICContributorRequest"/> <output message="tns:add-DCMICContributorResponse"/> </operation> <operation name="add-DCMICreator"> <documentation>Service definition of function ax__add_DCMICreator</documentation> <input message="tns:add-DCMICreatorRequest"/> <output message="tns:add-DCMICreatorResponse"/> </operation> <operation name="add-DCMILanguage"> <documentation>Service definition of function ax__add_DCMILanguage</documentation> <input message="tns:add-DCMILanguageRequest"/> <output message="tns:add-DCMILanguageResponse"/> </operation> <operation name="add-DCMISource"> <documentation>Service definition of function ax__add_DCMISource</documentation> <input message="tns:add-DCMISourceRequest"/> <output message="tns:add-DCMISourceResponse"/> </operation> <operation name="add-DCMIPublisher"> <documentation>Service definition of function ax__add_DCMIPublisher</documentation> <input message="tns:add-DCMIPublisherRequest"/> <output message="tns:add-DCMIPublisherResponse"/> </operation> </pre>
--	---

	<pre> <operation name="add-DCMIRelation"> <documentation>Service definition of function ax__add_DCMIRelation</documentation> <input message="tns:add-DCMIRelationRequest"/> <output message="tns:add-DCMIRelationResponse"/> </operation> <operation name="add-DCMIRight"> <documentation>Service definition of function ax__add_DCMIRight</documentation> <input message="tns:add-DCMIRightRequest"/> <output message="tns:add-DCMIRightResponse"/> </operation> </portType> <binding name="Descriptor_support" type="tns:Descriptor_supportPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="set-Did"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="set-Axinfo"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-ProtectionInfo"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="update-ProtectionInfo"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="del-ProtectionInfo"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="get-ProtectionInfo"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> </pre>
--	--

	<pre> </operation> <operation name="clear-AXOID"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-FingerPrintId"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-PromorOf"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-Translation"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-OptionalField"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="set-DublinCore"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-DCMICContributor"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-DCMICreator"> <SOAP:operation style="rpc" soapAction=""/> <input> </pre>
--	---

	<pre> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-DCMILanguage"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-DCMISource"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-DCMIPublisher"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-DCMIRelation"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="add-DCMIRight"> <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="Descriptor_support"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="Descriptor_support" binding="tns:Descriptor_support"> <SOAP:address location="http://www.axmedis.org/descriptors.cgi"/> </port> </service> </definitions> </pre>
--	--

Descriptor_Support

Method	Set DID
Description	Inserts the correct values in the DID table for the object and removes all the reference to the object from the other tables in order to prepare the object insertion/updating
Input parameters	xsd:string Axoid: AXOID of the object to be update xsd:int Version: version of the object to be updated xsd:anyURI ObjectLocation: URI of the real AXMEDIS object xsd:int NestingLevels: number of nesting levels inside the object xsd:boolean P2PWrapper: flag that mark this object as a P2P wrapper xsd:string WrappedAXOID: if the object is a P2P wrapper, then the AXOID of the wrapped object is reported here xsd:int WrappedVersion: if the object is a P2P wrapper, then the version of the wrapped object is reported here
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:set-Did> <Axoid></Axoid> <Version>0</Version> <ObjectLocation>file://discoc/parta</ObjectLocation> <NestingLevels>2</NestingLevels> <P2PWrapper>false</P2PWrapper> <WrappedAXOID></WrappedAXOID> <WrappedVersion>0</WrappedVersion> </ax:set-Did> </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:set-DidResponse> <Result>0</Result> </ax:set-DidResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor Support	
Method	set_Axinfo
Description	Insert data in the AXInfo table for the object
Input parameters	xsd:string Axoid: AXOID of the object to be update xsd:string Axcid: ID of the creator of the object xsd:string AccessMode: access mode for the object xsd:date CreationDate: creation date of the object xsd:date LastModificationDate: last modification date for the object xsd:string ObjectStatus: status of the object xsd:boolean isBasic: flag that show if the object is a basic one or not xsd:string Ownerid: ID of the owner of the object xsd:string Axdid: ID of the distributor of the object

	xsd:boolean isProtected: flag that shows if the object is protected xsd:boolean isGoverned: flag that shows if the object is governed xsd:string WorkItemId: Id of the work
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:set-Axinfo> <Axoid></Axoid> <Axcid></Axcid> <AccessMode>Free</AccessMode> <CreationDate>2005-01-03</CreationDate> <LastModificationDate>2005-02-02</LastModificationDate> <ObjectStatus>Published</ObjectStatus> <isBasic>false</isBasic> <Ownerid>12343423</Ownerid> <Axdid>43543254</Axdid> <isProtected>false</isProtected> <isGoverned>false</isGoverned> <WorkItemId>1edq323</WorkItemId> </ax:set-Axinfo> </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:set-AxinfoResponse> <Result>0</Result> </ax:set-AxinfoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support	
Method	add_FingerPrintId
Description	Add a FingerPrintID to an Object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string FingerPrintId: fingerprint id to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-FingerPrintId> <Axoid>324235435</Axoid> <FingerPrintId>efwfq423412</FingerPrintId> </ax:add-FingerPrintId> </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:add-FingerPrintIdResponse> <Result>0</Result> </ax:add-FingerPrintIdResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>
-------------------------	--

Descriptor Support	
Method	add_PromorOf
Description	Add a promoted object to a promor object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string AxoidPromoted: axoid promoted
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-PromorOf> <Axoid>214232543</Axoid> <AxoidPromoted>32415</AxoidPromoted> </ax:add-PromorOf> </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:add-PromorOfResponse> <Result>0</Result> </ax:add-PromorOfResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>

Descriptor Support	
Method	add_Translation
Description	Insert a translation for a field of an object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Language: language of the translation xsd:string Fieldname: name of the field translated xsd:string Translation: value of the translation
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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/"</pre>

	<pre> 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:add-Translation> <Axoid></Axoid> <Language>it</Language> <Fieldname>Author</Fieldname> <Translation>Il Mio Autore</Translation> </ax:add-Translation> </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:add-TranslationResponse> <Result>0</Result> </ax:add-TranslationResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support	
Method	add_OptionalField
Description	Insert the couple field/value for an optional field of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string FieldName: optional field name xsd:string Fieldvalue: optional field value
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-OptionalField> <Axoid></Axoid> <FieldName>MyCustomField</FieldName> <Fieldvalue>MyValue</Fieldvalue> </ax:add-OptionalField> </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:add-OptionalFieldResponse> <Result>0</Result> </ax:add-OptionalFieldResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support

Method	set_DublinCore
Description	insert single value information for Dublin Core Metadata Initiative of the object
Input parameters	<p>xsd:string Axioid: AXOID of the object</p> <p>The following fields are described in the DB related to DCMI and are related to DCMI specification. You can refer to these documents for their meaning.</p> <p>xsd:string Coverage:</p> <p>xsd:string Date:</p> <p>xsd:string Description:</p> <p>xsd:string Format:</p> <p>xsd:string Identifier:</p> <p>xsd:string Subject:</p> <p>xsd:string Title:</p> <p>xsd:string Type:</p> <p>xsd:string Abstract:</p> <p>xsd:string AccessRight:</p> <p>xsd:string Alternative:</p> <p>xsd:string AvailableStart:</p> <p>xsd:string AvailableEnd:</p> <p>xsd:string BibliographicCitation:</p> <p>xsd:string ConformsTo:</p> <p>xsd:string Created:</p> <p>xsd:string DateAccepted:</p> <p>xsd:string DateCopyrighted:</p> <p>xsd:string DateSubmitted:</p> <p>xsd:string EducationalLevel:</p> <p>xsd:float Extent:</p> <p>xsd:string HasFormat:</p> <p>xsd:string HasPart:</p> <p>xsd:string HasVersion:</p> <p>xsd:string IsFormatOf:</p> <p>xsd:string IsreferencedBy:</p> <p>xsd:string IsReplacedBy:</p> <p>xsd:string IsRequiredBy:</p> <p>xsd:string Issued:</p> <p>xsd:string IsVersionOf:</p> <p>xsd:string License:</p> <p>xsd:string Mediator:</p> <p>xsd:string Medium:</p> <p>xsd:string Modified:</p> <p>xsd:string Provenance:</p> <p>xsd:string References:</p> <p>xsd:string replaces:</p> <p>xsd:string requires:</p> <p>xsd:string RightsHolder:</p> <p>xsd:string Spatial:</p> <p>xsd:string TableOfContents:</p> <p>xsd:string Temporal:</p> <p>xsd:string ValidStart:</p> <p>xsd:string ValidEnd:</p>
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
Request Sample Message	<pre><?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"</pre>

	<pre> 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:set-DublinCore> <Axoid>12324325</Axoid> <Coverage></Coverage> <Date></Date> <Description></Description> <Format></Format> <Identifier></Identifier> <Subject>eeee</Subject> <Title></Title> <Type></Type> <Abstract>wevffv fdsbvfebebw</Abstract> <AccessRight></AccessRight> <Alternative></Alternative> <AvailableStart></AvailableStart> <AvailableEnd></AvailableEnd> <BibliographicCitation></BibliographicCitation> <ConformsTo></ConformsTo> <Created></Created> <DateAccepted></DateAccepted> <DateCopyrighted></DateCopyrighted> <DateSubmitted></DateSubmitted> <EducationalLevel></EducationalLevel> <Extent>0.0</Extent> <HasFormat></HasFormat> <HasPart></HasPart> <HasVersion></HasVersion> <IsFormatOf></IsFormatOf> <IsreferencedBy></IsreferencedBy> <IsReplacedBy></IsReplacedBy> <IsRequiredBy></IsRequiredBy> <Issued></Issued> <IsVersionOf></IsVersionOf> <License></License> <Mediator></Mediator> <Medium></Medium> <Modified></Modified> <Provenance></Provenance> <References></References> <replaces></replaces> <requires></requires> <RightsHolder></RightsHolder> <Spatial></Spatial> <TableOfContents></TableOfContents> <Temporal></Temporal> <ValidStart>2005-02-05</ValidStart> <ValidEnd></ValidEnd> </ax:set-DublinCore> </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:set-DublinCoreResponse> <Result>0</Result> </ax:set-DublinCoreResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support	
Method	add_DCMICContributor
Description	Add a Contributor to the DCMI of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Contributor: contributor to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-DCMICContributor> <Axoid>3442354235</Axoid> <Contributor>conrttttrrr</Contributor> </ax:add-DCMICContributor> </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:add-DCMICContributorResponse> <Result>0</Result> </ax:add-DCMICContributorResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>

Descriptor_Support	
Method	add_DCMICreator
Description	Add a Creator to the DCMI of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Creator: creator to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-DCMICCreator> <Axoid>435436</Axoid> <Creator>my creator</Creator> </ax:add-DCMICCreator> </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"></pre>

	<pre> xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:add-DCMIDCMICreatorResponse> <Result>0</Result> </ax:add-DCMIDCMICreatorResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
--	--

Descriptor_Support	
Method	add_DCMILanguage
Description	Add a Language to the DCMI of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Language: language to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-DCMILanguage> <Axoid>1234315231</Axoid> <Language>it</Language> </ax:add-DCMILanguage> </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:add-DCMILanguageResponse> <Result>0</Result> </ax:add-DCMILanguageResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support	
Method	add_DCMISource
Description	Add a Source to the DCMI of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Source: source to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-DCMISource> <Axoid>43543235343</Axoid> <Source>it is my source</Source> </ax:add-DCMISource> </SOAP-ENV:Body> </pre>

Response Sample Message	<pre> </SOAP-ENV:Envelope> <?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:add-DCMISourceResponse> <Result>0</Result> </ax:add-DCMISourceResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
-------------------------	---

Descriptor Support	
Method	add DCMIPublisher
Description	Add a Publisher to the DCMI of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Publisher: publisher to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-DCMIPublisher> <Axoid>4354236523</Axoid> <Publisher>my publiher is here</Publisher> </ax:add-DCMIPublisher> </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:add-DCMIPublisherResponse> <Result>0</Result> </ax:add-DCMIPublisherResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor Support	
Method	add DCMIRelation
Description	Add a Relation to the DCMI of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Relation: relation to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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" </pre>

	<pre> xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:add-DCMIRelation> <Axoid>4354235</Axoid> <Relation>fggrwea</Relation> </ax:add-DCMIRelation> </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:add-DCMIRelationResponse> <Result>0</Result> </ax:add-DCMIRelationResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support	
Method	add DCMIRight
Description	Add a Right to the DCMI of the object
Input parameters	xsd:string Axoid: AXOID of the object xsd:string Right: right to be added
Output parameters	xsd:int Result: result value, 0 means OK, other values have an associated error code to be defined
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:add-DCMIRight> <Axoid>23445432</Axoid> <Right>Print</Right> </ax:add-DCMIRight> </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:add-DCMIRightResponse> <Result>0</Result> </ax:add-DCMIRightResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support	
Method	add ProtectionInfo
Description	add a ProtectionInfo record
Input parameters	xsd:string Axoid: AXOID of the object xsd:int Version: version of the object

	xsd:string ProtectionStamp: protection timestamp (maybe UUID field) xsd:string ProtectionInfo: string containing the protection info
Output parameters	xsd:int Result: result 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:add-ProtectionInfo> <Axoid>4315r3fr343</Axoid> <Version>0</Version> <ProtectionStamp>122344cfr545</ProtectionStamp> <ProtectionInfo>this is the protection info for this file</ProtectionInfo> </ax:add-ProtectionInfo> </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:add-ProtectionInfoResponse> <Result>0</Result> </ax:add-ProtectionInfoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>

Descriptor Support	
Method	update ProtectionInfo
Description	update a ProtectionInfo record
Input parameters	xsd:string Axoid: AXOID of the object xsd:int Version: version of the object xsd:string ProtectionStamp: protection timestamp (maybe UUID field) xsd:string ProtectionInfo: string containing the protection info
Output parameters	xsd:int Result: result 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:update-ProtectionInfo> <Axoid>2354f3r43f</Axoid> <Version>0</Version> <ProtectionStamp>341rf31r32f</ProtectionStamp> <ProtectionInfo>pi</ProtectionInfo> </ax:update-ProtectionInfo> </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"></pre>

	<pre> xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:update-ProtectionInfoResponse> <Result>0</Result> </ax:update-ProtectionInfoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
--	--

Descriptor_Support	
Method	get_ProtectionInfo
Description	get a ProtectionInfo record
Input parameters	xsd:string Axoid: AXOID of the object xsd:int Version: version of the object xsd:string ProtectionStamp: protection timestamp (maybe UUID field)
Output parameters	complex type with: xsd:int Result: that is the result of the operation xsd:string ProtectionInfo: protection info requested
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:get-ProtectionInfo> <Axoid>3tfeaqrv43</Axoid> <Version>5</Version> <ProtectionStamp>dsvgferw43agfvr</ProtectionStamp> </ax:get-ProtectionInfo> </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:protectioninfo> <result> <RetCode>0</RetCode> <ProtectionInfo>pi record</ProtectionInfo> </result> </ax:protectioninfo> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor_Support	
Method	del_ProtectionInfo
Description	remove a ProtectionInfo record
Input parameters	xsd:string Axoid: AXOID of the object xsd:int Version: version of the object xsd:string ProtectionStamp: protection timestamp (maybe UUID field)
Output parameters	xsd:int Result: result 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" </pre>

	<pre> xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ax"> <SOAP-ENV:Body> <ax:del-ProtectionInfo> <Axoid>234fr3f</Axoid> <Version>4</Version> <ProtectionStamp>43fce24v5r245</ProtectionStamp> </ax:del-ProtectionInfo> </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:del-ProtectionInfoResponse> <Result>0</Result> </ax:del-ProtectionInfoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Descriptor Support	
Method	clear_AXINFO
Description	clear all descriptors apart from ProtectionInfo records for a certain AXINFO
Input parameters	xsd:string Axoid: AXOID whose Descriptors have to be cleared
Output parameters	xsd:int Result: result 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:clear-AXOID> <Axoid>wcqer343</Axoid> </ax:clear-AXOID> </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:clear-AXOIDResponse> <Result>0</Result> </ax:clear-AXOIDResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

3.7.2 Publication_support (EXITECH, CRS4)

Publication support	
Method	Description
getModifiedObject	This method returns a list of AXOID modified or inserted after a certain timestamp

Publication_Support	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="Publication_support" targetNamespace="http://www.axmedis.org/pub_support.wsdl" xmlns:tns="http://www.axmedis.org/pub_support.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="result"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="NumberOfResult" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="res" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType> <!-- operation request element --> <element name="date" type="xsd:string"/> <!-- operation request element --> <element name="time" type="xsd:string"/> <!-- operation response element --> <element name="return" type="ax:result"/> </schema> </types> <message name="getModifiedObjectRequest"> <part name="date" element="ax:date"/> <part name="time" element="ax:time"/> </message> <message name="getResult"> <part name="return" element="ax:return"/> </message> <portType name="Publication_supportPortType"> <operation name="getModifiedObject"> <documentation>Service definition of function ax__getModifiedObject</documentation> <input message="tns:getModifiedObjectRequest"/> <output message="tns:getResult"/> </operation> </portType> <binding name="Publication_support" type="tns:Publication_supportPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="getModifiedObject"> </pre>

	<pre> <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="Publication_support"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="Publication_support" binding="tns:Publication_support"> <SOAP:address location="http://www.axmedis.org/pub_support.cgi"/> </port> </service> </definitions> </pre>
--	--

Publication_Support	
Method	getModifiedObject
Description	Returns the list of AXOID that have been inserted/updated after a certain timestamp
Input parameters	xsd:string date : date after which the object has been inserted/updated xsd:string time : time after which the object has been inserted/updated
Output parameters	A complex type that has the following components: xsd:int RetCode : return code, where 0 means OK and greater than 0 corresponds to an error xsd:int NumberOfResult : number of results returned in the following sequence unbounded sequence of xsd:string res : a sequence (0 or more elements) of AXOID
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:getModifiedObject> <date>2005-01-31</date> <time>21:30</time> </ax:getModifiedObject> </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:getResult> <return> <RetCode>0</RetCode> <NumberOfResult>3</NumberOfResult> <res>36974216423</res> <res>36974216424</res> </return> </ax:getResult> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

	<pre> <res>36974213423</res> </return> </ax:getResult> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
--	---

3.7.3 User Support

This web service is able to return an error code for a user authentication, when a triplet of user, password and operation is provided. The operation is optional and if not present only the password against the user name will be checked.

The error code returned are:

- 0: the username exists, the password is correct and the operation (if any) is allowed)
- 1: the username does not exist; no check on operation
- 2: the username exists, but the password do not match; no check on operation
- 3: the username exists, the password is correct but the user is not entitled for such operation

Publication support	
Method	Description
authenticateUser	This methods allows to verify if an user is authenticated or authorized to a certain operation.

User Support	
WSDL	<pre> <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/" /> <!-- operation request element --> <element name="userName" type="xsd:string"/> <!-- operation request element --> <element name="userPassword" type="xsd:string"/> <!-- operation request element --> <element name="operation" type="xsd:int"/> <!-- operation response element --> <element name="Result" type="xsd:int"/> </schema> </types> <message name="authenticate-userRequest"> <part name="userName" element="ax:userName"/> <part name="userPassword" element="ax:userPassword"/> <part name="operation" element="ax:operation"/> </message> <message name="authenticate-userResponse"> <part name="Result" element="ax:Result"/> </message> <portType name="User_supportPortType"> <operation name="authenticate-user"> <documentation>Service definition of function ax__authenticate_user</documentation> <input message="tns:authenticate-userRequest"/> <output message="tns:authenticate-userResponse"/> </operation> </portType> </pre>

	<pre> <binding name="User_support" type="tns:User_supportPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="authenticate-user"> <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="User_support"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="User_support" binding="tns:User_support"> <SOAP:address location="http://www.axmedis.org/user_soap.cgi"/> </port> </service> </definitions> </pre>
Method	authenticateUser
Description	Returns an error code with respect to the user authentication procedure
Input parameters	xsd:string userName: userName to be checked xsd:string userPassword: password to be checked xsd:int operationCode: operation code to be verified (optional, or -1 means only authentication)
Output parameters	xsd:int RetCode: return code, where 0 means OK and greater than 0 corresponds to an error
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:authenticate-user> <userName>username</userName> <userPassword>password</userPassword> <operation>13</operation> </ax:authenticate-user> </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:authenticate-userResponse> <Result>0</Result> </ax:authenticate-userResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

3.7.4 CAMART Support

It is necessary to create an access point also for the logs in order to provide to CAMART and other possibly interested service the availability of the log via an interoperable channel.

The interface defined is very similar to that exposed by the CAMART itself for the most general case.

Publication_support	
Method	Description
extractLog	This method allows to extract logs according to certain criteria

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/" 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="AXOID" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="AXUID" type="xsd:string" minOccurs="1" maxOccurs="1"/> <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="1" maxOccurs="1"/> <element name="Detail2" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="Detail3" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="Detail4" type="xsd:string" minOccurs="1" maxOccurs="1"/> </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 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"/> </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 according to some query criteria
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 returned</p> <p>xsd:boolean registration: true if the timestamp is the registration timestamp, false if it is the execution timestamp. Not relevant if timestamp is not reported.</p> <p>xsd:string AXOID: object for which the log that have to be extracted</p> <p>xsd:string AXUID: user for which the log 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 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:extractLog> <user>username</user> <pwd>password</pwd> <timestamp>2005-01-01T00:00:00</timestamp> <registration>true</registration> <AXOID>1343214ewf</AXOID> <AXUID>1432dqwcd</AXUID> </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> <AXOID>244231rfwe</AXOID> <AXUID>21rfc434fl</AXUID> <RegistrationTimestamp>2005-02-01T13:32:45</RegistrationTimestamp> <ExecutionTimestamp>2005-02-01T13:22:45</ExecutionTimestamp> <Detail1>play</Detail1> <Detail2>3</Detail2> <Detail3>PC</Detail3> <Detail4>11</Detail4> </Log> <Log> <AXOID>436534tqf</AXOID> <AXUID>43t2gfv5q43</AXUID> <RegistrationTimestamp>2005-02-03T13:32:46</RegistrationTimestamp> <ExecutionTimestamp>2005-02-02T13:32:46</ExecutionTimestamp> <Detail1>burn</Detail1> <Detail2>4</Detail2> <Detail3></Detail3> <Detail4></Detail4> </Log> </logs> </ax:ResultLog> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

3.7.5 AXDBQuerySupport

Inside the Axmedis Database it is necessary to identify a sub module that is capable of resolving the queries issued to the Query Support, also for a matter of symmetry with the P2P and Crawling module. This module have to export a WebService Interface equals to that exposed by the query support in order to receive from it the queries, resolve them of the DB only and giving back the results.

The interface exposed is therefore the same that is detailed by the WSDL in section 4 of this document.

3.7.6 P2P Hub Node Support

This web service allows P2P node to perform operations on the P2PHub table of the AXDB in order to index objects without transferring them. This web service operates in conjunction with Descriptors support

P2PHub_support	
Method	Description
add_P2PID	This methods allows to add a record on the P2PHub table
update_P2PID	This methods allows to update an existing record on the P2PHub table
del_P2PID	This methods allows to remove an existing record on the P2PHub table
get_P2PID_Details	This methods allows to get AXOID, Version and URI from the P2PID

The formalization in terms of WSDL and SOAP messages of the webservice follows:

P2PHub_Support	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="P2PHub_support" targetNamespace="http://www.axmedis.org/p2phub.wsdl" xmlns:tns="http://www.axmedis.org/p2phub.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="details"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="Axoid" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="Version" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="ObjectLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1" nillable="true"/> </sequence> </complexType> <!-- operation request element --> <element name="P2PID" type="xsd:string"/> <!-- operation request element --> <element name="Axoid" type="xsd:string"/> <!-- operation request element --> <element name="Version" type="xsd:int"/> <!-- operation request element --> </pre>

	<pre> <element name="ObjectLocation" type="xsd:anyURI"/> <!-- operation response element --> <element name="Result" type="xsd:int"/> <!-- operation response element --> <element name="P2PID" type="ax:details"/> </schema> </types> <message name="add-P2PIDRequest"> <part name="P2PID" element="ax:P2PID"/> <part name="Axoid" element="ax:Axoid"/> <part name="Version" element="ax:Version"/> <part name="ObjectLocation" element="ax:ObjectLocation"/> </message> <message name="add-P2PIDResponse"> <part name="Result" element="ax:Result"/> </message> <message name="update-P2PIDRequest"> <part name="P2PID" element="ax:P2PID"/> <part name="Axoid" element="ax:Axoid"/> <part name="Version" element="ax:Version"/> <part name="ObjectLocation" element="ax:ObjectLocation"/> </message> <message name="update-P2PIDResponse"> <part name="Result" element="ax:Result"/> </message> <message name="del-P2PIDRequest"> <part name="P2PID" element="ax:P2PID"/> </message> <message name="del-P2PIDResponse"> <part name="Result" element="ax:Result"/> </message> <message name="get-P2PID-DetailsRequest"> <part name="P2PID" element="ax:P2PID"/> </message> <message name="P2PDDetails"> <part name="P2PID" element="ax:P2PID"/> </message> <portType name="P2PHub_supportPortType"> <operation name="add-P2PID"> <documentation>Service definition of function ax__add_P2PID</documentation> <input message="tns:add-P2PIDRequest"/> <output message="tns:add-P2PIDResponse"/> </operation> <operation name="update-P2PID"> <documentation>Service definition of function ax__update_P2PID</documentation> <input message="tns:update-P2PIDRequest"/> <output message="tns:update-P2PIDResponse"/> </operation> <operation name="del-P2PID"> <documentation>Service definition of function ax__del_P2PID</documentation> <input message="tns:del-P2PIDRequest"/> <output message="tns:del-P2PIDResponse"/> </operation> <operation name="get-P2PID-Details"> <documentation>Service definition of function ax__get_P2PID_Details</documentation> <input message="tns:get-P2PID-DetailsRequest"/> <output message="tns:P2PDDetails"/> </pre>
--	--

	<pre> </operation> </portType> <binding name="P2PHub_support" type="tns:P2PHub_supportPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="add-P2PID"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="update-P2PID"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="del-P2PID"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="get-P2PID-Details"> <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="P2PHub_support"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="P2PHub_support" binding="tns:P2PHub_support"> <SOAP:address location="http://www.axmedis.org/p2phub.cgi"/> </port> </service> </definitions> </pre>
--	---

P2PHub_Support	
Method	add_P2PID
Description	This methods allows to add a record on the P2PHub table
Input parameters	xsd:string P2PID: the unique id of the record and primaty key xsd:string Axoid: the AXOID of the original object xsd:int Version: the version of the original object xsd:anyURI ObjectLocation: the location on the P2P network of the original object
Output parameters	xsd:int Result: result of the operation (0 means OK, other value will be decoded in error codes)
Request	<pre> <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope </pre>

Sample Message	<pre> 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:add-P2PID> <P2PID>12345667788990</P2PID> <Axoid>wdsfweqf434rf</Axoid> <Version>0</Version> <ObjectLocation>p2p://123455/whereIwant</ObjectLocation> </ax:add-P2PID> </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:add-P2PIDResponse> <Result>0</Result> </ax:add-P2PIDResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

P2PHub_Support	
Method	update_P2PID
Description	This methods allows to update an existing record on the P2PHub table
Input parameters	<p>xsd:string P2PID: the unique id of the record and primaty key</p> <p>xsd:string Axoid: the AXOID of the original object</p> <p>xsd:int Version: the version of the original object</p> <p>xsd:anyURI ObjectLocation: the location on the P2P network of the original object</p>
Output parameters	xsd:int Result: result of the operation (0 means OK, other value will be decoded in error codes)
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:update-P2PID> <P2PID>12345667788990</P2PID> <Axoid>wdsfweqf434rf</Axoid> <Version>0</Version> <ObjectLocation>p2p://123455/whereIwant</ObjectLocation> </ax:update-P2PID> </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:update-P2PIDResponse> <Result>0</Result> </ax:update-P2PIDResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

P2PHub_Support	
Method	del_P2PID
Description	This methods allows to remove an existing record on the P2PHub table
Input parameters	xsd:string P2PID: the unique id of the record and primaty key
Output parameters	xsd:int Result: result of the operation (0 means OK, other value will be decoded in error codes)
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:del-P2PID> <P2PID>1234567890</P2PID> </ax:del-P2PID> </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:del-P2PIDResponse> <Result>0</Result> </ax:del-P2PIDResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>

P2PHub_Support	
Method	get_P2PID_Details
Description	This methods allows to get AXOID, Version and URI from the P2PID
Input parameters	xsd:string P2PID: the unique id of the record and primaty key
Output parameters	<p>A complex type that contains the following data:</p> <p>xsd:int Result: result of the operation (0 means OK, other value will be decoded in error codes)</p> <p>xsd:string Axoid: the AXOID of the original object</p> <p>xsd:int Version: the version of the original object</p> <p>xsd:anyURI ObjectLocation: the location on the P2P network of the original object</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:get-P2PID-Details> <P2PID>1234567890</P2PID> </ax:get-P2PID-Details> </SOAP-ENV:Body></pre>

Response Sample Message	<pre> </SOAP-ENV:Envelope> <?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:P2PDetails> <P2PID> <RetCode>0</RetCode> <Axoid>swfweqr4r43</Axoid> <Version>12</Version> <ObjectLocation>p2p://123465/whereIget</ObjectLocation> </P2PID> </ax:P2PDetails> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
-------------------------	---

3.8 AXMEDIS Administrative Web Database Interface (EXITECH, DSI)

Module Profile		
AXMEDIS Administrative Web Database Interface		
Executable or Library(Support)	Web Application	
Single Thread or Multithread	NA	
Language of Development	PHP	
Responsible Name	Fioravanti	
Responsible Partner	EXITECH	
Status (proposed/approved)	Proposed	
Platforms supported	All platform capable of web browsing on the client side and all PHP compliant platform on server side	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
AXMEDIS 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.
Web Interface		
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
PEAR LIBRARY		PHP License, Apache License, LGPL, BSD style (depending on modules)

AXMEDIS Administrative Web Database Interface will provide to the AXMEDIS administrator access to the min functionalities of AXMEDIS that are related to the database.

These functionalities will be mainly:

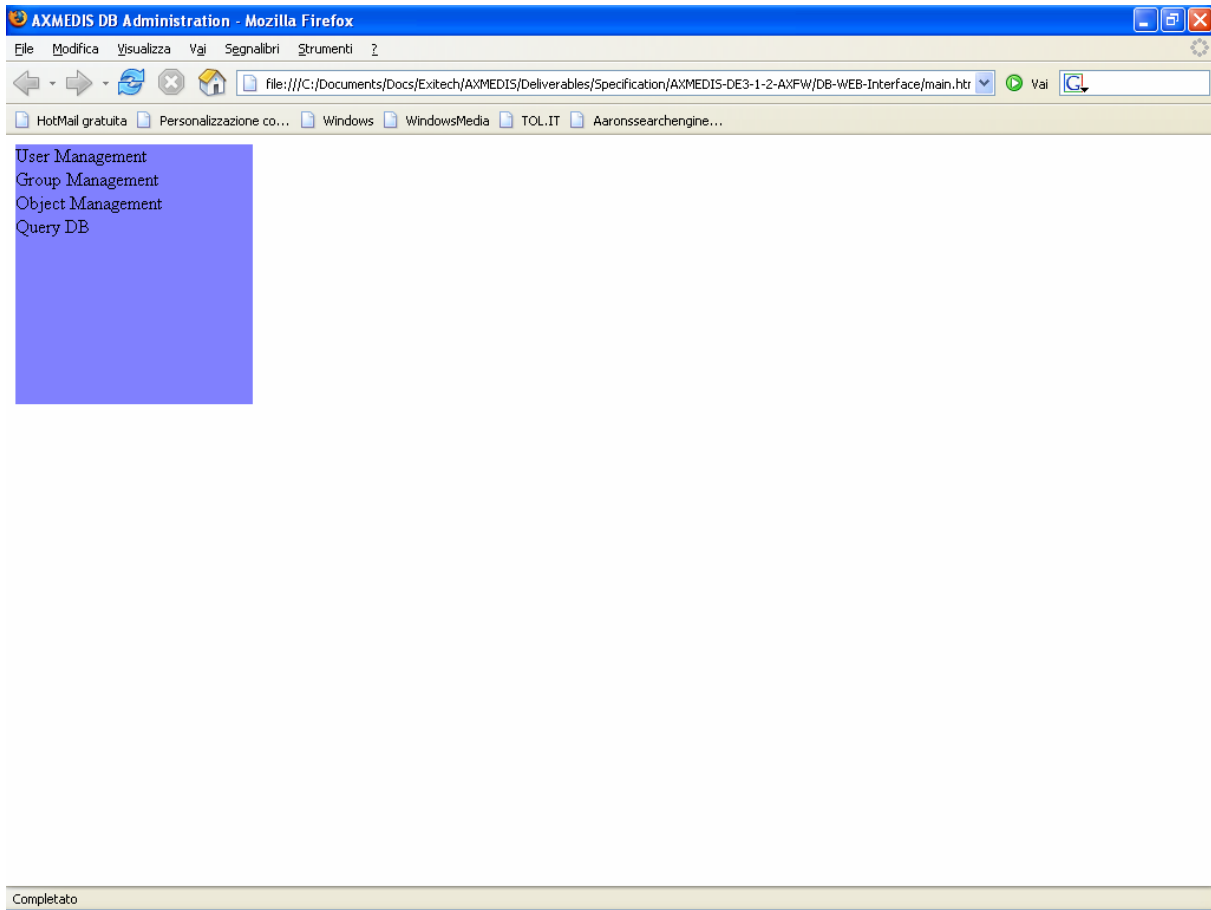
- Add user
- Delete User
- Update user data
- Update user rights
- Add Group
- Delete Group
- Update group data
- Assign users to group
- Assign groups to user
- Lists user of the basis of the rights
- List users on the basis of the user data
- List group
- Download Object for opening
- Delete an object (this means that all the version of the object will be removed)
- Mark an object as unavailable
- Delete a specific version of an object
- Insert objects
- Update object
- Make query on the system (by the means of Query Support Web Service Interface), that means:
 - List object according to certain criteria
 - List Action-Log on the basis of certain criteria
 - Create selection/query
 - Actualize selection/query
 - Delete selection/query
 - List Versions of the objects
 -

The web interface will be simple and effective having on the left side a menu bar for choosing operations and on the centre of the page the parameters for accessing to the requested functionality.

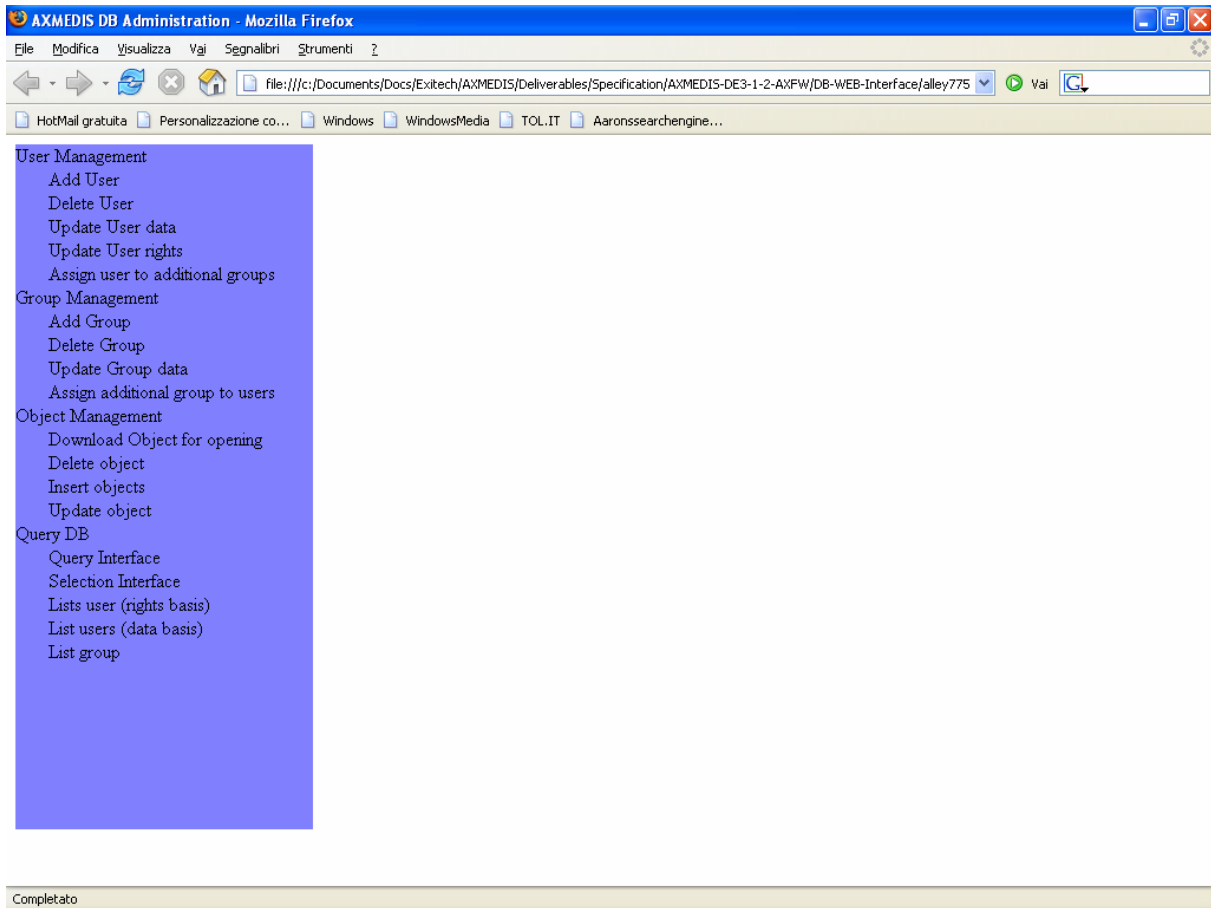
In order to increase accessibility no frame will be used.

A raw interface after the login can be:

DE3.1.2E – Framework and Tools Specification (Database and Gathering)



By clicking on each item, the menu will change, in the following image all the menus are expanded in order to save space in the present document:



Each functionality will have a different mask in the centre of the page that allow to operate on the functionality.

Some of the functionality will be listed below only as sample since several data have already to be defined.

3.8.1 Add user

AXMEDIS DB Administration - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti ?

file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775

HotMail gratuita Personalizzazione co... Windows WindowsMedia TOL.IT Aaronssearchengine...

User Information:

Name:

Company:

Email Address:

Geographical Address:

City/Town:

State/Province:

Zip/Postal:

Country:

Telephone:

Fax:

main Group: group1

User Rights:

Open object: ☐

Insert object: ☐

Delete object: ☐

Update object: ☐

User Management

- Add User
- Delete User
- Update User data
- Update User rights
- Assign user to additional groups

Group Management

Object Management

Query DB

Completato

3.8.2 Delete user

The screenshot shows a web browser window titled "AXMEDIS DB Administration - Mozilla Firefox". The address bar displays a file path: "file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775". The browser's menu bar includes "File", "Modifica", "Visualizza", "Vai", "Segnalibri", and "Strumenti". The toolbar shows navigation icons and a search bar. Below the browser window, the web application interface is visible. On the left, a blue sidebar contains a menu with the following items: "User Management", "Add User", "Delete User", "Update User data", "Update User rights", "Assign user to additional groups", "Group Management", "Object Management", and "Query DB". The main content area is white and contains a "UserID:" label followed by a text input field. Below the input field are two buttons: "Submit" and "Reset". At the bottom of the browser window, a status bar shows the word "Completato".

3.8.3 Update user data

AXMEDIS DB Administration - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti ?

file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775

HotMail gratuita Personalizzazione co... Windows WindowsMedia TOL.IT Aaronssearchengine...

User Management

- Add User
- Delete User
- Update User data
- Update User rights
- Assign user to additional groups

Group Management

Object Management

Query DB

User Information: USERID

Name: username

Company: usercompany

Email Address: useremail@email.com

Geographical Address: useraddress

City/Town: usercity

State/Province: userstate

Zip/Postal: userzip

Country: usercountry

Telephone: userphone

Fax: userfax

main Group: group1

Submit Reset

Completato

3.8.4 Update user rights

AXMEDIS DB Administration - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti ?

file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775

HotMail gratuita Personalizzazione co... Windows WindowsMedia TOL.IT Aaronssearchengine...

User Management

- Add User
- Delete User
- Update User data
- Update User rights
- Assign user to additional groups

Group Management

Object Management

Query DB

User Rights: USERID (Name)

Open object: ☐

Insert object: ☐

Delete object: ☐

Update object: ☐

Submit Reset

Completato

3.8.5 Assign user to additional groups

AXMEDIS DB Administration - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti ?

file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775

HotMail gratuita Personalizzazione co... Windows WindowsMedia TOL.IT Aaronssearchengine...

User Management
 Add User
 Delete User
 Update User data
 Update User rights
 Assign user to additional groups
 Group Management
 Object Management
 Query DB

Select groups for user

Select User Available Groups Current group for user

user 1 Group 2 Group 3 group 1 Group 4

maingroup add remove

Submit Reset

Completato

3.8.6 Add Group

AXMEDIS DB Administration - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti ?

file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775

HotMail gratuita Personalizzazione co... Windows WindowsMedia TOL.IT Aaronssearchengine...

User Management
Group Management
Add Group
Delete Group
Update Group data
Assign additional group to users
Object Management
Query DB

Group Information:

Name:

Description:

Completato

3.8.7 Delete group

AXMEDIS DB Administration - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti ?

file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775

HotMail gratuita Personalizzazione co... Windows WindowsMedia TOL.IT Aaronssearchengine...

User Management
Group Management
Add Group
Delete Group
Update Group data
Assign additional group to users
Object Management
Query DB

Delete Group

GroupID:

Completato

3.8.8 Update group data

AXMEDIS DB Administration - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti ?

file:///c:/Documents/Docs/Exitech/AXMEDIS/Deliverables/Specification/AXMEDIS-DE3-1-2-AXFW/DB-WEB-Interface/alley775

HotMail gratuita Personalizzazione co... Windows WindowsMedia TOL.IT Aaronssearchengine...

User Management
Group Management
Add Group
Delete Group
Update Group data
Assign additional group to users
Object Management
Query DB

Update Group: GroupID

Description:

Completato

3.8.9 Assign additional users to group

The screenshot shows the 'AXMEDIS DB Administration' web application running in Mozilla Firefox. The left sidebar contains a menu with the following items: User Management, Group Management, Add Group, Delete Group, Update Group data, Assign additional group to users, Object Management, and Query DB. The main content area is titled 'Select users for group'. It features a 'Select group' dropdown menu with 'Group 1' selected and 'maingroup' entered in the text field below it. To the right, there are two vertical lists: 'Available Users' containing 'user 2' and 'user 3', and 'Current users for group' containing 'user 1' and 'user 4'. Between these lists are 'add' and 'remove' buttons. At the bottom of the main area are 'Submit' and 'Reset' buttons.

Completato

3.8.10 Delete Object

The screenshot shows the 'AXMEDIS DB Administration' web application. The left sidebar menu is the same as in the previous screenshot, but with 'Delete Object' highlighted under 'Object Management'. The main content area is titled 'Delete AXMEDIS Object'. It contains a text input field labeled 'AXOID:' with an empty box next to it. Below the input field are 'Submit' and 'Reset' buttons.

3.8.11 Mark an Object as unavailable

The screenshot shows the AXMEDIS DB Administration interface in Mozilla Firefox. The left sidebar contains a menu with the following items: User Management, Group Management, Object Management, Delete Object, Download Object, Insert/Update Object, and Query DB. The main content area displays the 'Mark an object as unavailable' form. The form includes a text input field for 'AXOID', a text input field for 'Unavailability Reason', and two buttons: 'Submit' and 'Reset'.

This function will create a new version of the object without any Descriptor or other info available in order to allow the user to remove the version for which the user has no more the necessary right.

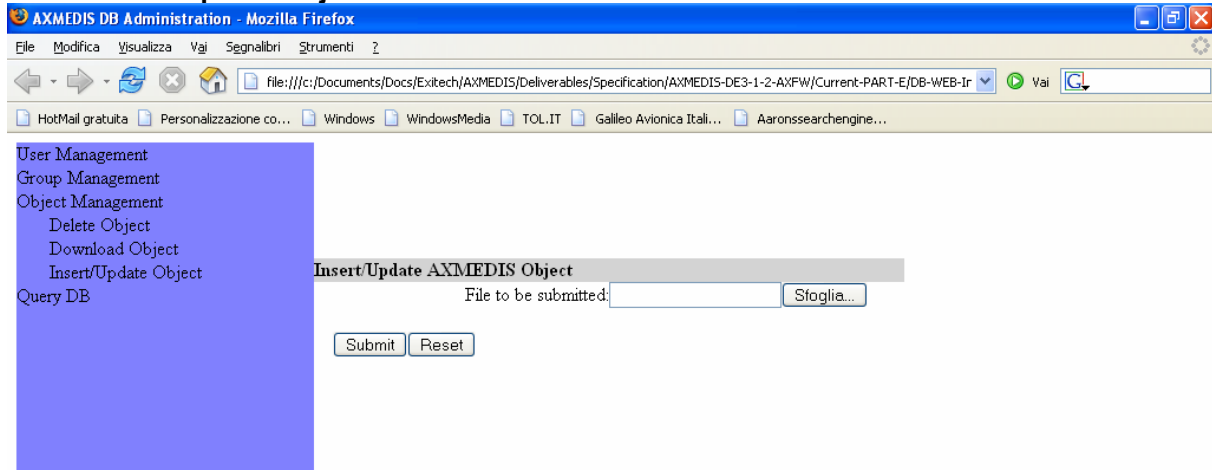
3.8.12 Delete one or more versions of an Object

The screenshot shows the AXMEDIS DB Administration interface in Mozilla Firefox. The left sidebar contains the same menu as in the previous screenshot. The main content area displays the 'Delete one or more version of an AXMEDIS Object' form. The form includes a text input field for 'AXOID', a 'List Versions' button, and a list of checkboxes for versions: version 12, version 10, version 9, version 4, and version 1. There are also 'Submit' and 'Reset' buttons.

3.8.13 Download Object

The screenshot shows the AXMEDIS DB Administration interface in Mozilla Firefox. The left sidebar contains the same menu as in the previous screenshots. The main content area displays the 'Download AXMEDIS Object' form. The form includes a text input field for 'AXOID' and two buttons: 'Submit' and 'Reset'.

3.8.14 Insert/Update Object



3.8.15 Query Interface

See the interface in Query User Interface.

3.8.16 Selection Interface

See the interface in Query User Interface.

3.9 AXMEDIS Object Loader/Saver (EXITECH, DSI, CRS4)

Module Profile		
AXMEDIS Object Loader/Saver		
Executable or Library(Support)	None	
Single Thread or Multithread	Multithread	
Language of Development	C++ and JAVA	
Responsible Name	Fioravanti	
Responsible Partner	EXITECH	
Status (proposed/approved)	Proposed	
Platforms supported	Windows, Unix like, MacOS X (also some mobile are supported by GSOAP)	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
	AXMEDIS database interface	
	AXINFO Metadata Mapper	
	AXMEDIS Object Manager	
	AXMEDIS Database Interface	
	WF Query and database interface	
File Formats Used	Shared with	File format name or reference to a section
AXMEDIS XML format		
WSDL		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Web Service	C++ for the Saver	

	JAVA for Loader	
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
Gsoap (http://gsoap2.sourceforge.net/)		GPL, MPL

AXMEDIS object Loader/Saver is a Web service that is capable of getting an AXMEDIS object (in the MPEG21 compliant format plus the additional information stored inside) and putting it in the database, that is the loading function; and it is also capable, given an Object ID (AXOID) to return the AXMEDIS object in the MPEG21 compliant format, that is the saving function.

The load and save function are always considered from the point of view of the AXMEDIS system, so that Load means “load in AMXEDIS” and save means export outside AMXEDIS.

This module interfaces with the Metadata Mapper module for having the possibility of extracting the significant data from the AXINFO and with the AXMEDIS Object manager in order to have the possibility to have decrypted objects and all the services offered by such tool.

The AXMEDIS Loader/Saver interact, or better is part of, the AXMEDIS database Interface.

This module offers the services for saving in the database objects and for loading objects from the database.

This module can be split in the Loader and Saver in order to have a better understanding of the different behaviour of the two services.

Both services of this module are implemented as web-services and therefore in the following specification, the WSDL of the proposed service together with samples of the SOAP messages will be reported.

The services can operate in synchronous and asynchronous mode. In asynchronous mode a Listener approach will be implemented, in the sense that the user, in order to be allowed to use asynchronous operation must offer a web-service with a known interface to receive the result of the operation.

The saver module will automatically set the version of the object inside the object as soon as the object is saved increasing the number currently present in the database.

3.9.1 Saver/Indexer (EXITECH, DSI, CRS4)

Saver/Indexer module offers basically the commit service, both in sync and async mode.

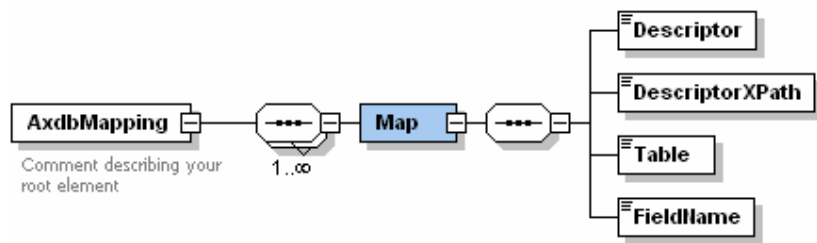
In the following a table with the WSDL of the Saver service is reported, followed by the detailed tables of the different methods offered by the service, with samples of SOAP messages for requestes and responses.

During the saving process it is necessary to know which fields of the Descriptors have to be imported as optional fields, this mapping process can be difficultly obtained with the model proposed for P2P since the needs seems to be very different. To cover this part of mapping that is very simple from the database side, during the installation and configuration of the AXMEDIS system, it will be created a file that will be read by the saver each time it has to import an axmedis object inside the AXDB.

The format proposed for the mapping has to consider the descriptor where the record is, the full path of the item to be found inside the descriptor and the name of the table and of the field that is wanted in the AXDB.

In the case of OptionalFields, the fieldname will be used as a record of the table as for all the other 1 to n relationship in ER model.

The schema for this representation is reported in the following schema, using XPath for describing the fullPath in the DescriptorXPath tag:



The textual version of the schema follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="AxdbMapping">
    <xs:annotation>
      <xs:documentation>Comment describing your root element</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="Map">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Descriptor" type="xs:string"/>
              <xs:element name="DescriptorXPath" type="xs:string"/>
              <xs:element name="Table" type="xs:string"/>
              <xs:element name="FieldName" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

In the following example file, it is reported an example of one of the fields in the mpeg7 descriptor mapped to the field Spatial of DublinCore (in has no sense, it is only for giving an example) .

```

<?xml version="1.0" encoding="UTF-8"?>
<AxdbMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="axdb-mapping-v-1-2.xsd">
  <Map>
    <Descriptor>dsc_mpeg7</Descriptor>
    <DescriptorXPath>/Statement/Mpeg7/DescriptionUnit/MediaFormat/BitRate</DescriptorXPath>
    <Table>DublinCore</Table>
    <FieldName>Spatial</FieldName>
  </Map>
</AxdbMapping>

```

In the following other example the final part of the path is an attribute of a tag instead of a and it is mapped to an OptionalFiled record whose field name is visualcodingformathref.

```

<?xml version="1.0" encoding="UTF-8"?>
<AxdbMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="axdb-mapping-v-1-2.xsd">
  <Map>

```

```

        <Descriptor>dsc_mpeg7</Descriptor>

        <DescriptorXPath>/Statement/Mpeg7/DescriptionUnit/MediaFormat/VisualCodingFormat[@href]</DescriptorXPath>
        <Table>OptionalField</Table>
        <FieldName>visualcodingformathref</FieldName>

    </Map>
</AxdbMapping>

```

In the following the specification of the webservice is reported in terms of WSDL and SOAP messages, while the following table summarizes the webservice methods, with a simple description:

Saver Webservice	
Method	Description
commit_sync	This method allows a synchronous commit of an object provided in a URI by the webservice client. The response will arrive when the operation will be completed
commit_async	This method allows an asynchronous commit of an object provided in a URI by the webservice client. The response will arrive when the operation will be accepted and the result will be communicated to a Listener specified below

Saver	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="Saver" targetNamespace="http://www.axmedis.org/saver.wsdl" xmlns:tns="http://www.axmedis.org/saver.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="sync-result"> <sequence> <element name="result" type="xsd:boolean" minOccurs="1" maxOccurs="1"/> <element name="version" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="errmsg" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="errorCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <!-- operation request element --> <element name="URI" type="xsd:anyURI"/> <!-- operation request element --> </pre>

	<pre> <element name="User" type="xsd:string"/> <!-- operation request element --> <element name="Password" type="xsd:string"/> <!-- operation response element --> <element name="return" type="ax:sync-result"/> <!-- operation request element --> <element name="CommitListenerService" type="xsd:anyURI"/> <!-- operation request element --> <element name="ListenerID" type="xsd:string"/> <!-- operation response element --> <element name="result" type="xsd:boolean"/> </schema> </types> <message name="commit-syncRequest"> <part name="URI" element="ax:URI"/> <part name="User" element="ax:User"/> <part name="Password" element="ax:Password"/> </message> <message name="getSyncResult"> <part name="return" element="ax:return"/> </message> <message name="commit-asyncRequest"> <part name="URI" element="ax:URI"/> <part name="User" element="ax:User"/> <part name="Password" element="ax:Password"/> <part name="CommitListenerService" element="ax:CommitListenerService"/> <part name="ListenerID" element="ax:ListenerID"/> </message> <message name="commit-asyncResponse"> <part name="result" element="ax:result"/> </message> <portType name="SaverPortType"> <operation name="commit-sync"> <documentation>Service definition of function ax:commit_sync</documentation> <input message="tns:commit-syncRequest"/> <output message="tns:getSyncResult"/> </operation> <operation name="commit-async"> <documentation>Service definition of function ax:commit_async</documentation> <input message="tns:commit-asyncRequest"/> <output message="tns:commit-asyncResponse"/> </operation> </portType> <binding name="Saver" type="tns:SaverPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="commit-sync"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </pre>
--	---

	<pre> </operation> <operation name="commit-async"> <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="Saver"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="Saver" binding="tns:Saver"> <SOAP:address location="http://www.axmedis.org/saver.cgi"/> </port> </service> </definitions> </pre>
--	---

Saver	
Method	commit_sync
Description	Allow to synchronously commit an axmedis object in the AXDB
Input parameters	xsd:anyURI URI : URI where the axmedis object can be found xsd:string User : credential for commit xsd:string Password : credential for commit
Output parameters	A complexType with the following fields: xsd:boolean result : result of the operation xsd:int version : version assigned to the committed object xsd:string errmsg : in case of result==false this is the place for an error message xsd:int errorcode : in case of result==false this is the place for an error code
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:commit-sync> <URI>file:///C:/samples/axobject.ax</URI> <User>username</User> <Password>the password</Password> </ax:commit-sync> </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:getSyncResult> <return> </pre>

	<pre> <result>true</result> <version>3</version> <errmsg>OK</errmsg> <errorcode>0</errorcode> </return> </ax:getSyncResult> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
--	--

Saver	
Method	commit_async
Description	Allow to asynchronously commit an axmedis object in the AXDB
Input parameters	xsd:anyURI URI : URI where the axmedis object can be found xsd:string User : credential for commit xsd:string Password : credential for commit xsd:anyURI CommitListenerService : URI of the web service where a message will be send with the results of operation (a getSyncResult tag will be sent) xsd:string ListenerID : ID of the request set by the requester in order to collect back the results in a correct manner
Output parameters	xsd:boolean result : boolean results of the operation that means only that the service has put the request in a list and that will communicate later the result of the operation to the listener specified in input parameters
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:commit-async> <URI>file:///C:/samples/axobject.ax</URI> <User>username</User> <Password>my password</Password> <CommitListenerService>http://who.request.me/CommitListener.cgi</CommitListenerService> <ListenerID>12345677</ListenerID> </ax:commit-async> </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:commit-asyncResponse> <result>true</result> </ax:commit-asyncResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Since a listener is defined in the asynchronous mode, it is necessary to draw the specification also of this webservice that will be offered and implemented by a third party that uses the asynchronous commit. This is a sample of what can be implemented, and is the proposed standard also if different listener can be proposed.

CommitListener	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="CommitListener" targetNamespace="http://www.someone.org/listener.wsdl" </pre>

	<pre> xmlns:tns="http://www.someone.org/listener.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:ns/ax.xsd" xmlns:ns="urn:ns" 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:ns/ax.xsd" 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:ns/ax.xsd" xmlns:ns="urn:ns" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/"> <complexType name="sync-result"> <sequence> <element name="result" type="xsd:boolean" minOccurs="1" maxOccurs="1"/> <element name="version" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="errmsg" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="errorCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <complexType name="getSyncResult"> <sequence> <element name="return" type="ax:sync-result" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> </schema> <schema targetNamespace="urn:ns" 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:ns/ax.xsd" xmlns:ns="urn:ns" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/"> <!-- operation request element --> <element name="result" type="ax:getSyncResult"/> <!-- operation request element --> <element name="ListenerID" type="xsd:string"/> <!-- operation response element --> <element name="r" type="xsd:boolean"/> </schema> </types> <message name="commit-ListenerRequest"> <part name="result" element="ns:result"/> <part name="ListenerID" element="ns:ListenerID"/> </message> </pre>
--	---

	<pre> <message name="commit-ListenerResponse"> <part name="r" element="ns:r"/> </message> <portType name="CommitListenerPortType"> <operation name="commit-Listener"> <documentation>Service definition of function ns:commit_Listener</documentation> <input message="tns:commit-ListenerRequest"/> <output message="tns:commit-ListenerResponse"/> </operation> </portType> <binding name="CommitListener" type="tns:CommitListenerPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="commit-Listener"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ns"/> </input> <output> <SOAP:body use="literal" namespace="urn:ns"/> </output> </operation> </binding> <service name="CommitListener"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="CommitListener" binding="tns:CommitListener"> <SOAP:address location="http://www.someone.org/listener.cgi"/> </port> </service> </definitions> </pre>
Method	commit_Listener
Description	Listener web service that receive the result of the async commit
Input parameters	<p>A complexType with the following fields:</p> <p>xsd:boolean result: result of the operation</p> <p>xsd:int version: version assigned to the committed object</p> <p>xsd:string errormsg: in case of result==false this is the place for an error message</p> <p>xsd:int errorcode: in case of result==false this is the place for an error code</p> <p>xsd:string ListenerID: ID of the request set by the requester in order to collect beack the results in a correct manner</p>
Output parameters	xsd:boolean r: true or false depending on the result 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:ns/ax.xsd" xmlns:ns="urn:ns"> <SOAP-ENV:Body> <ns:commit-Listener> <result> <return> <result>true</result> <version>5</version> <errormsg>OK</errormsg> <errorcode>0</errorcode> </return> </result> <ListenerID>1223445fddd</ListenerID> </ns:commit-Listener> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

	</ns:commit-Listener> </SOAP-ENV:Body> </SOAP-ENV:Envelope>
Response Sample Message	<?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:ns:ax.xsd" xmlns:ns="urn:ns"> <SOAP-ENV:Body> <ns:commit-ListenerResponse> <r>true</r> </ns:commit-ListenerResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope>

3.9.2 Loader

Saver/Indexer module offers basically the checkout service, both in sync and async mode.

In the following a table with the WSDL of the Loader service is reported, followed by the detailed tables of the different methods offered by the service, with samples of SOAP messages for requestes and responses. The following table summarizes the methods of the Loader WebService a short description of the functionalities.

Loader WebService	
Method	Description
checkout_sync	This method allows a synchronous checkout of a predefined version of an object that will be provided in a URI communicated to the webservice client. The response arrives when the operation will be completed. The file in the URI will be removed as soon as it is downloaded, and in any case after a predefined timeout
checkout_async	This method allows a synchronous checkout of a predefined version of an object that will be provided in a URI communicated to the webservice client. The response will arrive when the operation will be accepted and the result will be communicated to a Listener specified below. The file in the URI will be removed as soon as it is downloaded, and in any case after a predefined timeout

Loader	
WSDL	<?xml version="1.0" encoding="UTF-8"?> <definitions name="Loader" targetNamespace="http://www.axmedis.org/loader.wsdl" xmlns:tns="http://www.axmedis.org/loader.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/"

	<pre> 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="checkout-result"> <sequence> <element name="result" type="xsd:boolean" minOccurs="1" maxOccurs="1"/> <element name="downloadURI" type="xsd:anyURI" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="errmsg" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> <element name="errorCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <!-- operation request element --> <element name="AXOID" type="xsd:string"/> <!-- operation request element --> <element name="version" type="xsd:int"/> <!-- operation request element --> <element name="User" type="xsd:string"/> <!-- operation request element --> <element name="Password" type="xsd:string"/> <!-- operation response element --> <element name="return" type="ax:checkout-result"/> <!-- operation request element --> <element name="CheckoutListenerService" type="xsd:anyURI"/> <!-- operation request element --> <element name="ListenerID" type="xsd:string"/> <!-- operation response element --> <element name="result" type="xsd:boolean"/> </schema> </types> <message name="checkout-syncRequest"> <part name="AXOID" element="ax:AXOID"/> <part name="version" element="ax:version"/> <part name="User" element="ax:User"/> <part name="Password" element="ax:Password"/> </message> <message name="getcheckoutResult"> <part name="return" element="ax:return"/> </message> <message name="checkout-asyncRequest"> <part name="AXOID" element="ax:AXOID"/> <part name="version" element="ax:version"/> <part name="User" element="ax:User"/> <part name="Password" element="ax:Password"/> <part name="CheckoutListenerService" element="ax:CheckoutListenerService"/> <part name="ListenerID" element="ax:ListenerID"/> </message> <message name="checkout-asyncResponse"> <part name="result" element="ax:result"/> </message> <portType name="LoaderPortType"> <operation name="checkout-sync"> <documentation>Service definition of function ax:checkout_sync</documentation> <input message="tns:checkout-syncRequest"/> <output message="tns:getcheckoutResult"/> </operation> <operation name="checkout-async"> </pre>
--	--

	<pre> <documentation>Service definition of function ax:checkout_async</documentation> <input message="tns:checkout-asyncRequest"/> <output message="tns:checkout-asyncResponse"/> </operation> </portType> <binding name="Loader" type="tns:LoaderPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="checkout-sync"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="checkout-async"> <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="Loader"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="Loader" binding="tns:Loader"> <SOAP:address location="http://www.axmedis.org/loader.cgi"/> </port> </service> </definitions> </pre>
--	---

Loader	
Method	checkout_sync
Description	Methods that allows to synchronously get a named version of an axmedis object or the last version of it.
Input parameters	xsd:string AXOID: AXOID of the object to be loaded xsd:int version: desired version for the object, if -1, the last version will be returned xsd:string User: credential for commit xsd:string Password: credential for commit
Output parameters	A complexType named ax:getcheckoutResult with the following fields: xsd:boolean result: result of the operation xsd:anyURI downloadURI: URI where the object can be downloaded xsd:string errmsg: in case of result==false this is the place for an error message xsd:int errorcode: in case of result==false this is the place for an error code
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"> </pre>

	<pre> <SOAP-ENV:Body> <ax:checkout-sync> <AXOID>12234567</AXOID> <version>-1</version> <User>my user</User> <Password>my password</Password> </ax:checkout-sync> </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:getcheckoutResult> <return> <result>true</result> <downloadURI>http://db.axmedis.org/myobjectishere.ax</downloadURI> <errmsg>OK</errmsg> <errorcode>0</errorcode> </return> </ax:getcheckoutResult> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Loader	
Method	checkuot_async
Description	Methods that allows to asynchronously get a named version of an axmedis object or the last version of it.
Input parameters	xsd:string AXOID : AXOID of the object to be loaded xsd:int version : desired version for the object, if -1, the last version wil be returned xsd:string User : credential for commit xsd:string Password : credential for commit xsd:anyURI CommitListenerService : URI of the web service where a message will be send with the results of operation (a getSyncResult tag will be sent) xsd:string ListenerID : ID of the request set by the requester in order to collect beack the results in a correct manner
Output parameters	xsd:boolean result : boolean results of the operation that means only that the service has put the request in a list and that will communicate later the result of the operation to the listener specified in input parameters
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:checkout-async> <AXOID>12235546</AXOID> <version>-1</version> <User>my user</User> <Password>my password</Password> <CheckoutListenerService>http://myhost/CheckoutListener.cgi</CheckoutListenerService> <ListenerID>ggt5255288</ListenerID> </ax:checkout-async> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
Response	<pre> <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope </pre>

Sample Message	<pre> 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:checkout-asyncResponse> <result>true</result> </ax:checkout-asyncResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
----------------	--

Since a listener is defined in the asynchronous mode, it is necessary to draw the specification also of this webservice that will be offered and implemented by a third party that uses the asynchronous commit.

CheckoutListener	
WSDL	
Method	ckecout_Listener
Description	Listener that receives the results of a checkout process of an axmedis object
Input parameters	<p>A complexType with name result with the following fields:</p> <p>xsd:boolean result: result of the operation</p> <p>xsd:anyURI URI: URI where the object is available</p> <p>xsd:string errmsg: in case of result==false this is the place for an error message</p> <p>xsd:int errorcode: in case of result==false this is the place for an error code</p> <p>xsd:string ListenerID: ID of the request set by the requester in order to collect back the results in a correct manner</p>
Output parameters	xsd:boolean r : true or false depending on the result 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:ns/ax.xsd" xmlns:ns="urn:ns"> <SOAP-ENV:Body> <ns:commit-Listener> <result>true</result> <ListenerID>612tedu3</ListenerID> </ns:commit-Listener> </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:ns/ax.xsd" xmlns:ns="urn:ns"> <SOAP-ENV:Body> <ns:commit-ListenerResponse> <r>true</r> </ns:commit-ListenerResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

The detailed architecture of the AXMEDIS Object Loader/saver is reported in the following picture, where it split in four parts:

- AXMEDIS Saver Web Interface
- AXMEDIS Loader Web Interface
- AXMEDIS Object Loader
- AXMEDIS Object Saver and Indexer

The main problem that drive the choice of language specification and technologies adoption is due to the fact that AXMEDIS Object Saver and Indexer must include an AXOM, that is written in C++. This implies that AXMEDIS Object Saver and Indexer itself must be written in C++.

The problem propagate in a cascade way to the modules that have a direct interface with this module that are:

- AXMEDIS Database Interface
- AXMEDIS Save Web Service

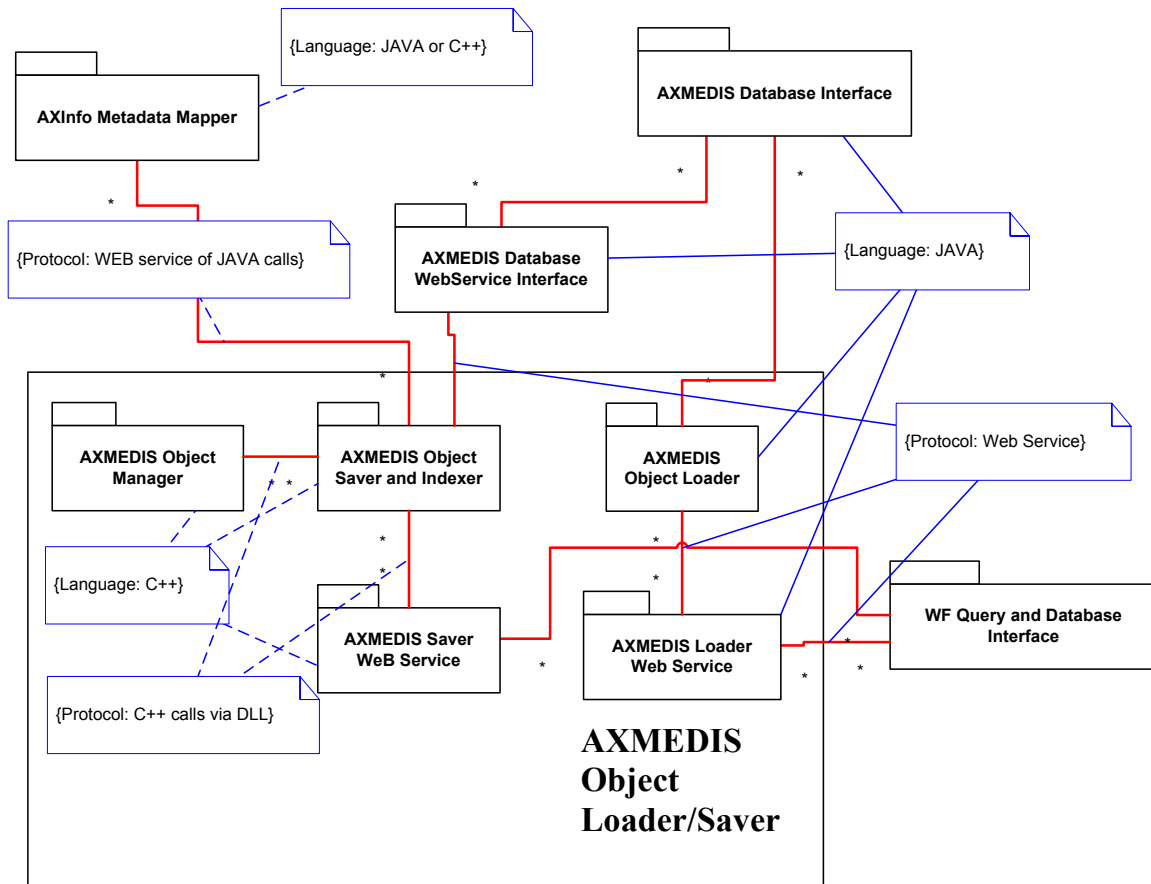
Since it is not feasible to write all the Database Interface in C++ for several reasons, it is better to implement all services in a JAVA package and export some methods also as a WebService, in order to allow modules that are not written in JAVA to easily communicate with the database.

It is better to implement AXMEDIS Saves Web Service directly in C++ in order to avoid not useful complexity at the system level for integrating the different modules.

The other parts of the Loader/Saver will be implemented directly in JAVA.

In the following diagram a detailed view of the Loader/Saver with constraints on languages and protocols is reported.

AXMEDIS Object Loader/Saver



3.10 Protection Models for AXMEDIS Objects repository (FUPF, AFI)

Module Profile		
Protection Models for AXMEDIS Object repository		
Executable or Library(Support)	WebService	
Single Thread or Multithread		
Language of Development		
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.)

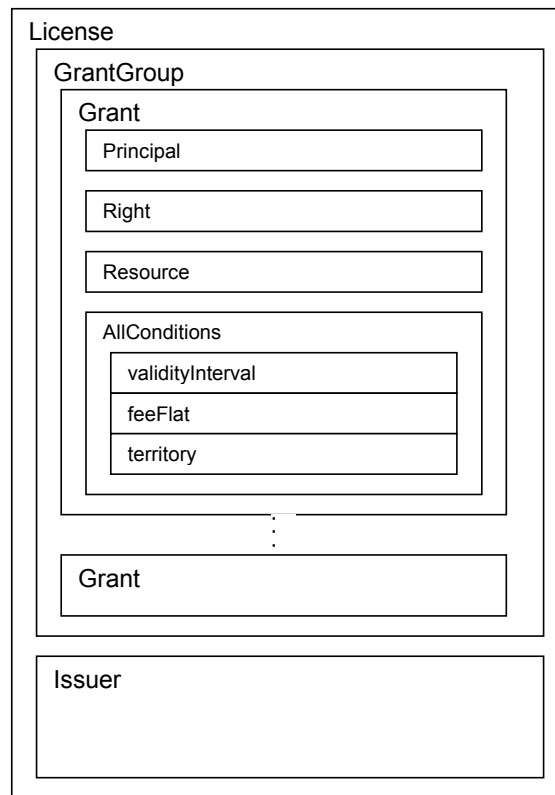
File Formats Used	Shared with	File format name or reference to a section
XML based (MPEG-21 REL)		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
Xerces		
Xalan		
XPath		

3.10.1 From REL licenses to ER model

In order to express DRM rules associated to AXMEDIS objects it has been decided to use MPEG-21 REL as primary rights expression language. Nevertheless, this is not the only possible language that can be used for expressing DRM rules, as stated in the DoW and the research guidelines document. We have planned to consider also ODRL (Open Digital Rights Language) as a language for expressing licenses.

For this reason, we propose a simple but scalable solution for expressing licenses into the UML and ER diagrams explained in more detail in next sections. This structure will simplify moving from one REL language to another (for instance from MPEG-21 REL to ODRL), expressing as many conditions as we want and also adding new conditions, if they appear.

The approach we have followed is to impose a common structure for licenses in order to simplify the parsing from the XML-based license into the ER structure. This structure for a final user license is shown in the next figure. Distributor licenses follow a similar structure.

**REL License structure**

For expressing the different types of conditions we can have, we have defined the following information:

- ConditionType: It indicates which kind of condition we are expressing.
- Five Tvalue fields and two NValue fields (more can be added if desired), whose values depend on the conditionType. See section “Information inside the conditions table” for details.

Using this structure we can easily define new conditions and implement support classes in the corresponding programming language depending on the condition.

We can also make complex queries over the defined ER diagram, only asking for different ConditionType, TvalueN and NvalueN.

Other possible approach for expressing conditions could have been to define a different table for each condition. We have discarded this approach, as conditions expressed in MPEG-21 REL have a lot of different possible and optional values, and making queries over such a structure could be even more complex than using only one table.

3.10.1.1 UML diagram for Licenses

The model that we are proposing allows storing the licenses in a relational database. All MPEG-21 REL Licenses can be stored in the database once parsed properly. With this model we are not restricting the syntactic structure of the licenses, but we need to transform them in a common format in order to facilitate searches.

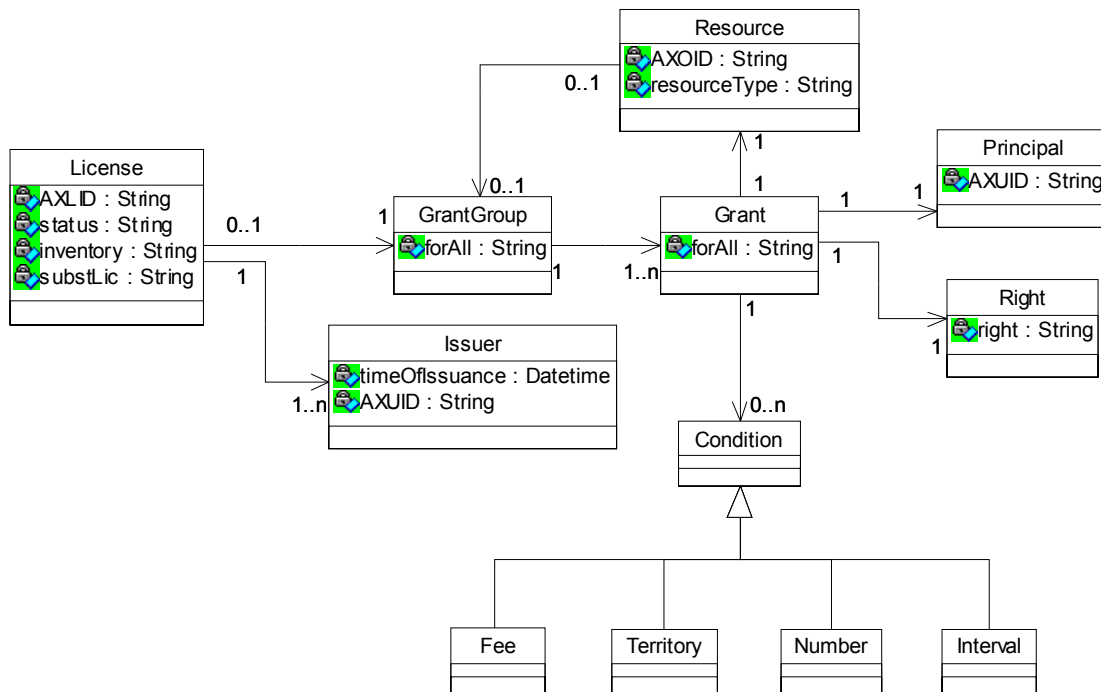
The figure in this section shows the UML class diagram that we are proposing for representing the licenses. We explain the diagram next.

Each *license* can have an *AXLID* element that contains a unique identifier for the license, one or more *issuer* element(s) and a *GrantGroup*, a *status* element contains the status of the license, e.g. revoked, a *substLic* element that contains the license that replaces the revoked one, and a *inventory* element that contains the variables defined in the license that can be referenced within this license.

Each *GrantGroup* contains a set of *Grants* and the *forAll* element where the variables or patterns are defined within this *GrantGroup* are placed.

Each *Grant* contains the information of the *right* granted, the *resource*, the *principal* and an optional set of *conditions* related to that right, and the *forAll* element where the variables or patterns are defined within this grant are placed.

In addition, we have to realise that a *resource* can be a *GrantGroup* (in case of Distributor Licenses).

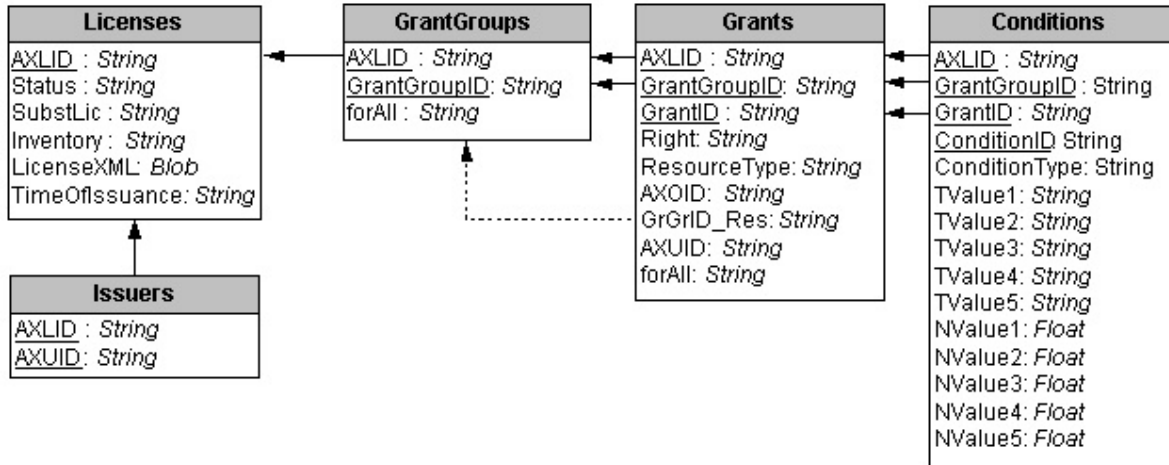


UML diagram for licenses

3.10.1.2 ER diagram for Licenses

To represent the content of a license in an Entity-Relationship diagram, we have to focus on the relations with a multiplicity 0..n. These relations show us the number of different tables that we need to store the represented information. The relations with a multiplicity of 1 – 1 can be stored always in the same table.

The next diagram shows how to create the different tables to store the license information. This solution provides the model for storing End-user Licenses, and also for storing Distributor Licenses.



ER diagram for licenses

Licenses

Number of indexes: ?
 Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
AXLID	PK	I	C-Large Length	Not allowed	
Status			C-Large Length	Not allowed	
SubstLic			C-Large Length	Allowed	
Inventory			C-Large Length	Allowed	
TimeofIssuance			C-Large Length	Not allowed	
LicenseXML			C-Blob	Not allowed	

Column details

1. AXLID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String representing the unique identifier for the license

2. Status

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String that contains the status of the license, possible values are valid or revoked

3. SubstLic

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: String that contains the MPEG-21 REL license that replaces the revoked one, if any

4. Inventory

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: String that contains the variables defined in the license, that can be referenced through this license

5. TimeofIssuance

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String that represents the specific date and time at which the license has been issued

6. LicenseXML

Physical data type: BLOB
 Allow NULLs: Not allowed
 Notes: contains the XML MPEG-21 REL license

Issuers

Number of indexes: ?
 Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
AXLID	PK	I	C-Large Length	Not allowed	
AXUID	PK	I	C-Large Length	Not allowed	

Column details

1. AXLID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String representing the unique identifier for the license

2. AXUID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String that represents the unique identifier of the AXMEDIS user that has issued the license

GrantGroups

Number of indexes: ?
 Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
AXLID	PK	I	C-Large Length	Not allowed	
GrantGroupID	PK	I	C-Large Length	Not allowed	
forAll			C-Large Length	Allowed	

Column details

1. AXLID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String representing the unique identifier for the license

2. GrantGroupID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String containing the unique identifier of the grantGroup

3. forAll

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: String that contains variables whose scope is this entire grantGroup uniquely identified by the GrantGroupID

Grants

Number of indexes: ?
 Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
AXLID	PK	I	C-Large Length	Not allowed	

GrantGroupID	PK	I	C-Large Length	Not allowed
GrantID	PK	I	C-Large Length	Not allowed
Right			C-Large Length	Not allowed
ResourceType			C-Large Length	Not allowed
AXOID		I	C-Large Length	Allowed
GrGrID_Res	FK	I	C-Large Length	Allowed
AXUID			C-Large Length	Not allowed
forAll			C-Large Length	Allowed

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

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String representing the unique identifier for the license

2. GrantGroupID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String containing the unique identifier of the grantGroup

3. GrantID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String containing the unique identifier of the grant

4. Right

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String that specifies the right granted

5. ResourceType

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String that specifies the type of object against which the principal of this grant has the right to perform an action. If the resourceType is Resource, then the object against which the AXMEDIS user can exercise the right is an AXMEDIS object, and if the resourceType is GrantGroup then the object is a grant or grantGroup, typically for distribution licenses

6. AXOID

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String containing the unique identifier of the AXMEDIS object

7. GrGrID_Res (FK)

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String containing the unique identifier of the grantGroup that can be issued

8. AXUID

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String identifying the AXMEDIS user to whom this grant conveys rights

9. forAll

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String that contains variables whose scope is the entire grant uniquely identified by the GrantID

Conditions

Number of indexes: ?
Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
AXLID	PK	I	C-Large Length	Not allowed	
GrantGroupID	PK	I	C-Large Length	Not allowed	
GrantID	PK	I	C-Large Length	Not allowed	
ConditionID	PK	I	C-Large Length	Not allowed	
ConditionType		I	C-Large Length	Not allowed	
TValue1			C-Large Length	Allowed	
TValue2			C-Large Length	Allowed	
TValue3			C-Large Length	Allowed	
TValue4			C-Large Length	Allowed	
TValue5			C-Large Length	Allowed	
NValue1			C-Float	Allowed	
NValue2			C-Float	Allowed	
NValue3			C-Float	Allowed	
NValue4			C-Float	Allowed	
NValue5			C-Float	Allowed	

Column details

1. AXLID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String representing the unique identifier for the license

2. GrantGroupID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String containing the unique identifier of the grantGroup

3. GrantID (PK)

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String containing the unique identifier of the grant

4. ConditionID

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String containing the unique identifier of the condition

5. ConditionType

Physical data type: LONGTEXT
 Allow NULLs: Not allowed
 Notes: String representing the type of the condition. This field can have the values specified in the ConditionType column of the Condition Table (see section 3.10.1.3). For example, this field can take the values territory or validityInterval

6. TValue(1-5)

Physical data type: LONGTEXT
 Allow NULLs: Allowed
 Notes: String that contains information related to the condition according to the ConditionType as defined in the Condition Table (see section 3.10.1.3). For example, if the ConditionType is validityInterval, the TValue1 contains a String that represents the date at which the interval of time defined by this condition begins and the TValue2 contains a String that represents the date at which the interval of time defined by this condition ends

7. Nvalue(1-5)

Physical data type: FLOAT
 Allow NULLs: Allowed
 Notes: Numeric value that contains information related to the condition according to the ConditionType as defined in the Condition Table (see section 3.10.1.3). For example, if the ConditionType is exerciseLimit, the NValue1 represents the limit on the number of times that certain exercises may occur

The relation between Tables and Classes is:

Table (ER)	Classes (UML) stored in the table
Licenses	License
Issuers	Issuer
GrantGroups	GrantGroup
Grants	Grant, Right, Resource, Principal
Conditions	Condition (all types)

To represent all type of conditions, we have decided to store the data in one unique table with a set of “standard” fields. Each field of this table corresponds to an attribute of the condition depending on the condition type.

We provide a table where we describe the mapping between the standard fields of the table (ER) and the condition attributes (UML).

In this model is very easy to add new types of conditions to the system without causing the reimplementation of a lot of modules. And, moreover, it makes easier and much more efficient the search of the information needed in the authorisation model.

3.10.1.3 Information inside the Conditions table

ConditionType	TValue1	TValue2	TValue3	TValue4	TValue5	NValue1	NValue2
territory	location (country, region, state, city, postalCode, street)	domain (uri)					
validityInterval	notBefore	notAfter					
validityIntervalFloating	serviceReference (*1) (serviceDescription (anonymousStateService, uddi, wsdlAddress, wsdlComplete), serviceParameters)	duration					
validityIntervalDurationPattern		duration					
validityIntervalStartsNow	validityInterval - notBefore	validityInterval - notAfter	backwardTolerance	forwardTolerance			
validityTimeMetered	serviceReference (*1)	duration	quantum				
validityTimePeriodic	Start	duration	phase	period		periodCount	
exerciseLimit	serviceReference (*1)					count	
feeFlat	serviceReference (*1)	rate - currency	to (*2) (paymentService (serviceReference , aba (institution, account), any)			rate - amount	
feeMetered	phase	rate - currency	to (*2)	per	by	rate - amount	
feePerInterval	serviceReference (*1)	rate - currency	to (*2)	per		rate - amount	
feePerUse		rate - currency	to (*2)			rate - amount	
feePerUsePrePay	serviceReference (*1)	rate - currency	to (*2)			rate - amount	initialNumberOfUses

DE3.1.2E – Framework and Tools Specification (Database and Gathering)

(*1) Same information as the first *serviceReference* field defined

(*2) Same information as the first *to* field defined

3.10.1.4 Example of information inside the License Database

In this section we present two examples of MPEG-21 REL licenses, for final users and distributors. Then, we present how this license information is stored in the different tables.

Lic1: Example of final user license

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-R-NS rel-
r.xsd urn:mpeg:mpeg21:2003:01-REL-SX-NS rel-sx.xsd urn:mpeg:mpeg21:2003:01-REL-MX-NS rel-mx.xsd">
  <r:grantGroup>
    <r:grant>
      <r:keyHolder>
        <r:info>
          <dsig:KeyValue>
            <dsig:RSAKeyValue>
              <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
              <dsig:Exponent>AQABAA==</dsig:Exponent>
            </dsig:RSAKeyValue>
          </dsig:KeyValue>
        </r:info>
      </r:keyHolder>
      <mx:play/>
      <mx:diReference>
        <mx:identifier>urn:a1-AXOID1-f</mx:identifier>
      </mx:diReference>
      <r:allConditions>
        <r:validityInterval>
          <r:notBefore>2005-01-01T01:00:00</r:notBefore>
          <r:notAfter>2005-04-01T01:00:00</r:notAfter>
        </r:validityInterval>
        <sx:exerciseLimit>
          <r:serviceReference>
            <sx:uddi>
              <sx:serviceKey>
                <sx:uuid>ee1398c0-8abe-11d7-a735-b8a03c50a862</sx:uuid>
              </sx:serviceKey>
            </sx:uddi>
          </r:serviceReference>
          <sx:count>150</sx:count>
        </sx:exerciseLimit>
      </r:allConditions>
    </r:grant>
    <r:grant>
      <r:keyHolder>
        <r:info>
          <dsig:KeyValue>
            <dsig:RSAKeyValue>
              <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
              <dsig:Exponent>AQABAA==</dsig:Exponent>
            </dsig:RSAKeyValue>
          </dsig:KeyValue>
        </r:info>
      </r:keyHolder>
      <mx:adapt/>
      <mx:diReference>
        <mx:identifier>urn:a1-AXOID1-f</mx:identifier>
      </mx:diReference>
      <r:allConditions>
        <sx:feeFlat>
          <r:serviceReference>
            <sx:uddi>
              <sx:serviceKey>
                <sx:uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</sx:uuid>
              </sx:serviceKey>
            </sx:uddi>
          </r:serviceReference>
        </sx:feeFlat>
      </r:allConditions>
    </r:grant>
  </r:grantGroup>

```

```

        </sx:serviceKey>
      </sx:uddi>
    </r:serviceReference>
  <sx:rate>
    <sx:amount>5.00</sx:amount>
    <sx:currency>iso:EUR</sx:currency>
  </sx:rate>
  <sx:to>
    <sx:paymentService>
      <serviceReference>
        <sx:uddi>
          <sx:serviceKey>
            <sx:uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</sx:uuid>
          </sx:serviceKey>
        </sx:uddi>
      </serviceReference>
    </sx:paymentService>
  </sx:to>
</sx:feeFlat>
</r:allConditions>
</r:grant>
</r:grantGroup>
<r:issuer>
  <r:keyHolder>
    <r:info>
      <dsig:KeyValue>
        <dsig:RSAKeyValue>
          <dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
          <dsig:Exponent>AQABAA==</dsig:Exponent>
        </dsig:RSAKeyValue>
      </dsig:KeyValue>
    </r:info>
  </r:keyHolder>
</r:details>
  <r:timeOfIssue>2005-01-01T01:00:00</r:timeOfIssue>
</r:details>
</r:issuer>
</r:license>

```

Lic2: Example of distributor license

```

<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
  xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-R-NS rel-
  r.xsd urn:mpeg:mpeg21:2003:01-REL-SX-NS rel-sx.xsd urn:mpeg:mpeg21:2003:01-REL-MX-NS rel-mx.xsd">
  <r:grantGroup>
    <r:grant>
      <r:keyHolder>
        <r:info>
          <dsig:KeyValue>
            <dsig:RSAKeyValue>
              <dsig:Modulus>g8NRYMG307NqJgmZG8TIUOp+9sQjsAai+hILpkBiLaf4RhvjS3pD0dvy1YosEjKL8mk/KTGniC+pY4ia5kLBy
              Q==</dsig:Modulus>
              <dsig:Exponent>AQABAA==</dsig:Exponent>
            </dsig:RSAKeyValue>
          </dsig:KeyValue>
        </r:info>
      </r:keyHolder>
    </r:grant>
  </r:grantGroup>
  <r:issue/>
  <r:grantGroup>
    <r:grant>
      <mx:play/>
      <mx:diReference>
        <mx:identifier>urn:a1-AXOID2-f</mx:identifier>
      </mx:diReference>
    </r:grant>
  </r:grantGroup>

```

```

</mx:diReference>
<r:allConditions>
  <sx:territory>
    <sx:location>
      <sx:country>iso:ES</sx:country>
    </sx:location>
  </sx:territory>
  <sx:feePerUse>
    <sx:rate>
      <sx:amount>1.00</sx:amount>
      <sx:currency>iso:EUR</sx:currency>
    </sx:rate>
    <sx:to>
      <sx:paymentService>
        <serviceReference>
          <sx:uddi>
            <sx:serviceKey>
              <sx:uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</sx:uuid>
            </sx:serviceKey>
          </sx:uddi>
        </serviceReference>
      </sx:paymentService>
    </sx:to>
  </sx:feePerUse>
</r:allConditions>
</r:grant>
<r:grant>
  <mx:adapt/>
  <mx:diReference>
    <mx:identifier>urn:a1-AXOID2-f</mx:identifier>
  </mx:diReference>
  <r:allConditions>
    <sx:exerciseLimit>
      <r:serviceReference>
        <sx:uddi>
          <sx:serviceKey>
            <sx:uuid>ee1398c0-8abe-11d7-a735-b8a03c50a862</sx:uuid>
          </sx:serviceKey>
        </sx:uddi>
      </r:serviceReference>
      <sx:count>3</sx:count>
    </sx:exerciseLimit>
  </r:allConditions>
</r:grant>
</r:grantGroup>
</r:grant>
</r:grantGroup>
<r:issuer>
  <r:keyHolder>
    <r:info>
      <dsig:KeyValue>
        <dsig:RSAKeyValue>
          <dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
          <dsig:Exponent>AQABAA==</dsig:Exponent>
        </dsig:RSAKeyValue>
      </dsig:KeyValue>
    </r:info>
  </r:keyHolder>
  <r:details>
    <r:timeOfIssue>2005-01-01T01:00:00</r:timeOfIssue>
  </r:details>
</r:issuer>
</r:license>

```

Licenses		
<u>AXLID</u>	<u>Status</u>	<u>TimeOfIssuance</u>
Lic1	Valid	2005-01-01T01:00:00
Lic2	Valid	2005-01-01T01:00:00

Issuers	
<u>AXLID</u>	<u>AXUID</u>
Lic1	Issuer1
Lic2	Issuer2

GrantGroups	
<u>AXLID</u>	<u>GrantGroupID</u>
Lic1	GG1
Lic2	GG1
Lic2	GG2

Grants							
<u>AXLID</u>	<u>GrantGroupID</u>	<u>GrantID</u>	<u>Right</u>	<u>ResourceType</u>	<u>Resource</u>	<u>GrGrID Res</u>	<u>Principal</u>
Lic1	GG1	Gr1	Play	Resource	AXOID1	- null -	Prin1
Lic1	GG1	Gr2	Adapt	Resource	AXOID1	- null -	Prin1
Lic2	GG1	Gr1	Issue	GrantGroup	- null -	GG2	Prin2
Lic2	GG2	Gr1	Play	Resource	AXOID2	- null -	- null -
Lic2	GG2	Gr2	Adapt	Resource	AXOID2	- null -	- null -

Conditions								
<u>AXLID</u>	<u>GrantGroupID</u>	<u>GrantID</u>	<u>ConditionID</u>	<u>ConditionType</u>	<u>Tvalue1</u>	<u>Tvalue2</u>	<u>Tvalue3</u>	<u>Nvalue1</u>
Lic1	GG1	Gr1	Con1	ValidityInterval	2005-01-01T01:00:00	2005-04-01T01:00:00		
Lic1	GG1	Gr1	Con2	exerciseLimit	Service_Referece1.xml			150
Lic1	GG1	Gr2	Con1	feeFlat	Service_Referece2.xml	iso:EUR	TO_1.xml	5
Lic2	GG2	Gr1	Con1	territory	<sx:country>iso:ES</sx:country>			
Lic2	GG2	Gr1	Con2	feePerUse		iso:EUR	TO_2.xml	1
Lic2	GG2	Gr2	Con1	exerciseLimit	Service_Referece1.xml			3

The fields *ServiceReference* and *To* contain an XML node as described in MPEG-21 REL. These fields are not used in searches, so we don't need to extend the information within them to other fields in the table. But, we need the information to perform an authorisation.

3.10.2 Proposal of PAR description based on REL information

Possible Available Rights (PAR) are a simplified version of licenses, as they provide the information of all the possible rights that a user has over an object.

We propose an UML diagram and an ER diagram for representing them, as described in more detail in next sections.

3.10.3 UML diagram for PAR

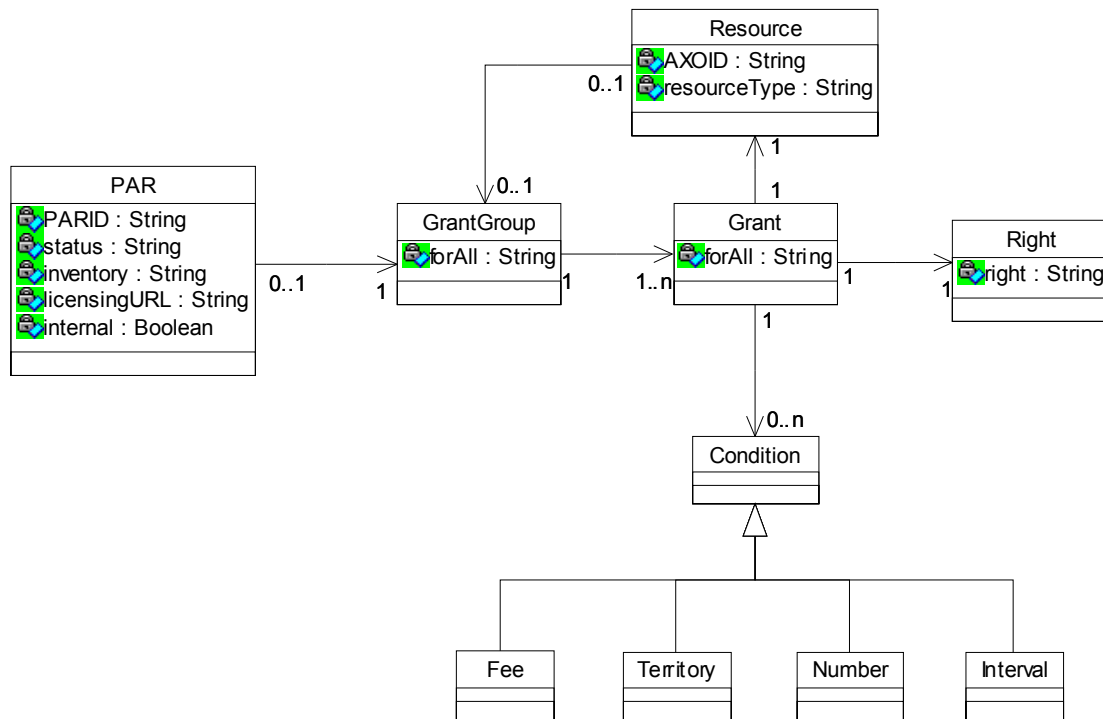
The figure shows the UML class diagram that we are proposing for representing the PAR. We explain it next.

Each *PAR* can have a *PARID* element that contains a unique identifier for the PAR, a *GrantGroup*, a *status* element contains the status of the license, e.g. valid, a *licensingURL* element that contains the reference to the service / distributor site that can issue a license for this PAR, if any, and a *inventory* element that contains the variables defined in the PAR that can be referenced within this PAR.

Each *GrantGroup* contains a set of *Grants* and the *forAll* element where the variables or patterns are defined within this GrantGroup are placed.

Each *Grant* contains the information of the *right* granted, the *resource* and an optional set of *conditions* related to that right, and the *forAll* element where the variables or patterns are defined within this grant are placed.

In addition, we have to realise that a *resource* can be a *GrantGroup* (in case of distribution of PAR, that is, licenses).

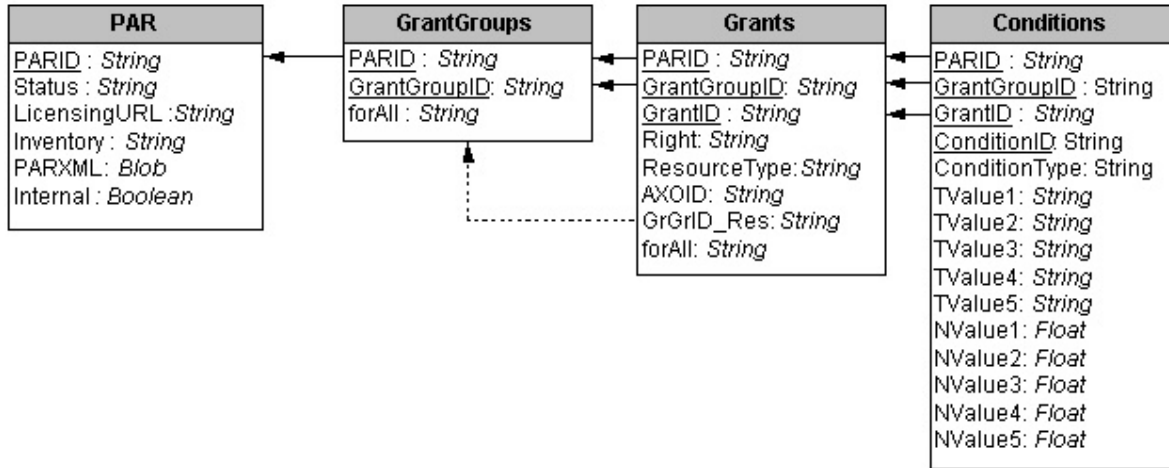


UML diagram for PAR

3.10.3.1 ER diagram for PAR

To store this content in a relational database we need 4 tables. The model is very similar to the License model, but we have substracted the data corresponding to the Issuer and Principal.

With this diagram, we do not pretend to show where or how are stored the resources or the RDD rights. Depending on the implementation tables can be related to that diagram accordingly.



ER diagram for PAR

PAR

Number of indexes: ?
Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
PARID	PK	I	C-Large Length	Not allowed	
Status			C-Large Length	Not allowed	
LicensingURL			C-Large Length	Allowed	
Inventory			C-Large Length	Allowed	
PARXML			C-Blob	Not allowed	
Internal			C-Boolean	Not allowed	

Column details

1. PARID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String representing the unique identifier for the PAR

2. Status

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String that contains the status of the PAR, e.g. verified

3. LicensingURL

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String that contains the URL to acquire a license for the object

4. Inventory

Physical data type: LONGTEXT

Allow NULLs: Allowed
Notes: String that contains the variables defined in the r:license element of the PAR, that can be referenced through this license

5. PARXML

Physical data type: BLOB
Allow NULLs: Not allowed
Notes: contains the XML MPEG-21 REL license of the PAR

6. Internal

Physical data type: BOOLEAN
Allow NULLs: Not allowed
Notes: Boolean identifying if the PAR is an internal PAR if set a true

GrantGroups

Number of indexes: ?
Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
PARID	PK	I	C-Large Length	Not allowed	
GrantGroupID	PK	I	C-Large Length	Not allowed	
forAll			C-Large Length	Allowed	

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

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String representing the unique identifier for the PAR

2. GrantGroupID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String containing the unique identifier of the grantGroup

3. forAll

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String that contains variables whose scope is this entire grantGroup uniquely identified by the GrantGroupID

Grants

Number of indexes: ?
Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
PARID	PK	I	C-Large Length	Not allowed	
GrantGroupID	PK	I	C-Large Length	Not allowed	
GrantID	PK	I	C-Large Length	Not allowed	
Right			C-Large Length	Not allowed	
ResourceType			C-Large Length	Not allowed	
AXOID		I	C-Large Length	Allowed	
GrGrID_Res	FK	I	C-Large Length	Allowed	
forAll			C-Large Length	Allowed	

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

Physical data type: LONGTEXT

AXMEDIS Project

CONFIDENTIAL

Allow NULLs: Not allowed
Notes: String representing the unique identifier for the PAR

2. GrantGroupID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String containing the unique identifier of the grantGroup

3. GrantID (PK)

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String containing the unique identifier of the grant

4. Right

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String that specifies the right granted

5. ResourceType

Physical data type: LONGTEXT
Allow NULLs: Not allowed
Notes: String that specifies the type of object against which the principal of this grant has the right to perform an action. If the resourceType is Resource, then the object against which the AXMEDIS user can exercise the right is an AXMEDIS object, and if the resourceType is GrantGroup then the object is a grant or grantGroup, typically for distribution licenses

6. AXOID

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String containing the unique identifier of the AXMEDIS object

7. GrGrID_Res (FK)

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String containing the unique identifier of the grantGroup that can be issued

8. forAll

Physical data type: LONGTEXT
Allow NULLs: Allowed
Notes: String that contains variables whose scope is the entire grant uniquely identified by the GrantID

Conditions

Number of indexes: ?
Number of foreign keys: ?

Columns		idx	Data type	Allow NULLs	Value/Range
PARID	PK	I	C-Large Length	Not allowed	
GrantGroupID	PK	I	C-Large Length	Not allowed	
GrantID	PK	I	C-Large Length	Not allowed	
ConditionID	PK	I	C-Large Length	Not allowed	
ConditionType		I	C-Large Length	Not allowed	
TValue1			C-Large Length	Allowed	
TValue2			C-Large Length	Allowed	
TValue3			C-Large Length	Allowed	
TValue4			C-Large Length	Allowed	
TValue5			C-Large Length	Allowed	
NValue1			C-Float	Allowed	
NValue2			C-Float	Allowed	
NValue3			C-Float	Allowed	
NValue4			C-Float	Allowed	
NValue5			C-Float	Allowed	

Column details**1. AXLID (PK)****Physical data type:**

LONGTEXT

Allow NULLs:

Not allowed

Notes:

String representing the unique identifier for the PAR

2. GrantGroupID (PK)**Physical data type:**

LONGTEXT

Allow NULLs:

Not allowed

Notes:

String containing the unique identifier of the grantGroup

3. GrantID (PK)**Physical data type:**

LONGTEXT

Allow NULLs:

Not allowed

Notes:

String containing the unique identifier of the grant

4. ConditionID**Physical data type:**

LONGTEXT

Allow NULLs:

Not allowed

Notes:

String containing the unique identifier of the condition

5. ConditionType**Physical data type:**

LONGTEXT

Allow NULLs:

Not allowed

Notes:

String representing the type of the condition. This field can have the values specified in the ConditionType column of the Condition Table (see section 3.10.1.3). For example, this field can take the values territory or validityInterval

6. TValue(1-5)**Physical data type:**

LONGTEXT

Allow NULLs:

Allowed

Notes:

String that contains information related to the condition according to the ConditionType as defined in the Condition Table (see section 3.10.1.3). For example, if the ConditionType is validityInterval, the TValue1 contains a String that represents the date at which the interval of time defined by this condition begins and the TValue2 contains a String that represents the date at which the interval of time defined by this condition ends

7. Nvalue(1-5)**Physical data type:**

FLOAT

Allow NULLs:

Allowed

Notes:

Numeric value that contains information related to the condition according to the ConditionType as defined in the Condition Table (see section 3.10.1.3). For example, if the ConditionType is exerciseLimit, the NValue1 represents the limit on the number of times that certain exercises may occur

The relation between Tables and Classes is:

Table (ER)	Classes (UML) stored in the table
PAR	PAR
GrantGroups	GrantGroup
Grants	Grant, Right, Resource
Conditions	Condition (all types)

To represent all type of conditions, we have decided to store the data in one unique table with a set of “standard” fields. Each field of this table corresponds to an attribute of the condition depending on the condition type.

3.10.3.2 Pricing for Final users


























The pricing of a given action that a final user wants to perform can be expressed in several places: PAR, distribution license and final user license.

Depending on this, the user can be informed through the PMS client that the right he wants to exercise will cost X euros, before buying the right or before exercising it after he has bought it.

3.10.4 Description of Web Services for accessing license and PAR database

This section describes the Web Service interface of the module that will communicate with the license and PAR database.

3.10.4.1 Architecture of LicenseDBManager module

LicenseDBManager	
	insertLicense(license : String) : boolean
	insertPAR(PAR : String) : boolean
	insertIPAR(IPAR : String) : boolean
	insertLicenseModel(licenseModel : String) : boolean
	retrieveLicense(AXLID : String) : String
	retrievePAR(PARID : String) : String
	retrieveIPAR(IPARID : String) : String
	retrieveLicenseModel(licModelID : String) : String
	retrieveLicenseIssuers(AXOID : String, Query : String) : String
	retrieveAXOIDPARs(Query : String) : String
	retrieveAXOIDIPARs(Query : String) : String
	updatePAR(PARID : String, NewPAR : String) : boolean
	updateLicenseStatus(AXLID : String, Status : String, NewAXLID : String) : boolean
	updatePARStatus(PARID : String, Status : String) : boolean
	updateIPARStatus(IPARID : String, Status : String) : boolean
	updatePARLicensingURL(PARID : String, LicensingURL : String) : boolean
	updateIPARLicensingURL(IPARID : String, LicensingURL : String) : boolean
	updateLicenseModel(licModelID : String, licModelNew : String) : boolean
	deleteLicenseModel(licModelID : String) : boolean
	deletePAR(PARID : String) : boolean
	deleteIPAR(PARID : String) : boolean
	isInternalPAR(PARID : String) : boolean
	revokeLicense(licenseID : String) : boolean
	revokeAddLicense(licenseID : String, newLicense : String) : boolean
	LicenseDBManager()

LicenseDBManager	
Method	Description
InsertLicense	This function inserts a license in the License DB
InsertPAR	This function inserts a PAR in the PAR DB
insertLicenseModel	This function inserts a license model in the License DB

retrieveLicense	This function returns a License stored in the License DB
retrieveIPAR	This function returns a internal PAR stored in the PAR DB
retrieveLicenseModel	This function returns a licenseModel stored in the License DB
retrieveLicenseIssuers	This function returns a set of AXUIDs from distributors that can offer a license for a specific AXOID and that satisfies some conditions expressed in a query.
retrieveAXOIDPARs	This function returns a set of AXOID referenced in PARs that satisfy some conditions expressed in a query
retrieveAXOIDIPARs	This function returns a set of AXOID referenced in internal PARs that satisfy some conditions expressed in a query
updatePAR	This function updates the PAR given a PARID and a new description of PAR
updateLicenseStatus	This function updates the status of the license in the License DB
updatePARStatus	This function updates the status of the PAR in the PAR DB
updateIPARStatus	This function updates the status of the IPAR in the PAR DB
updatePARLicensingURL	This function updates the PARLicensingURL in the PAR DB

LicenseDBManager	
WSDL	<pre> <wsdl:definitions name="LicenseDBManager" targetNamespace="urn:LicenseDBManager" xmlns:tns="urn:LicenseDBManager" 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="license" type="xsd:string" nillable="true"/> <xsd:element name="boolean_Response" type="xsd:boolean"/> <xsd:element name="PAR" type="xsd:string" nillable="true"/> <xsd:element name="IPAR" type="xsd:string" nillable="true"/> <xsd:element name="licenseModel" type="xsd:string" nillable="true"/> <xsd:element name="AXLID" type="xsd:string" nillable="true"/> <xsd:element name="string_Response" type="xsd:string" nillable="true"/> <xsd:element name="PARID" type="xsd:string" nillable="true"/> <xsd:element name="IPARID" type="xsd:string" nillable="true"/> <xsd:element name="licModelID" type="xsd:string" nillable="true"/> <xsd:element name="AXOID" type="xsd:string" nillable="true"/> <xsd:element name="Query" type="xsd:string" nillable="true"/> <xsd:element name="NewPAR" type="xsd:string" nillable="true"/> <xsd:element name="Status" type="xsd:string" nillable="true"/> <xsd:element name="NewAXLID" type="xsd:string" nillable="true"/> <xsd:element name="LicensingURL" type="xsd:string" nillable="true"/> <xsd:element name="licModelNew" type="xsd:string" nillable="true"/> <xsd:element name="licenseID" type="xsd:string" nillable="true"/> <xsd:element name="newLicense" type="xsd:string" nillable="true"/> </xsd:schema> </wsdl:types> <wsdl:message name="LicenseDBManager_retrieveAXOIDIPARs_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseDBManager_revokeLicense_1_Request"> <wsdl:part name="licenseID" element="ns0:licenseID"/> </wsdl:message> <wsdl:message name="LicenseDBManager_retrieveLicense_Response"> <wsdl:part name="response" element="ns0:string_Response"/> </wsdl:message> <wsdl:message name="LicenseDBManager_isInternalPAR_Response"> <wsdl:part name="response" element="ns0:boolean_Response"/> </wsdl:message> </pre>

	<pre> <wsdl:message name='LicenseDBManager_deletePAR_1_Request'> <wsdl:part name='PARID' element='ns0:PARID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateLicenseStatus_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrievePAR_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_insertLicense_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_deleteLicenseModel_1_Request'> <wsdl:part name='licModelID' element='ns0:licModelID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_insertLicenseModel_1_Request'> <wsdl:part name='licenseModel' element='ns0:licenseModel'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveAXOIDPARs_1_Request'> <wsdl:part name='Query' element='ns0:Query'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveLicense_1_Request'> <wsdl:part name='AXLID' element='ns0:AXLID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_insertPAR_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_insertLicenseModel_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_insertPAR_1_Request'> <wsdl:part name='PAR' element='ns0:PAR'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveLicenseModel_1_Request'> <wsdl:part name='licModelID' element='ns0:licModelID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updatePARLicensingURL_1_Request'> <wsdl:part name='PARID' element='ns0:PARID'/> <wsdl:part name='LicensingURL' element='ns0:LicensingURL'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updatePARStatus_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveIPAR_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveAXOIDPARs_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updatePAR_1_Request'> <wsdl:part name='PARID' element='ns0:PARID'/> <wsdl:part name='NewPAR' element='ns0:NewPAR'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveLicenseIssuers_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateLicenseStatus_1_Request'> <wsdl:part name='AXLID' element='ns0:AXLID'/> <wsdl:part name='Status' element='ns0:Status'/> <wsdl:part name='NewAXLID' element='ns0:NewAXLID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_revokeLicense_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_deleteIPAR_1_Request'> <wsdl:part name='PARID' element='ns0:PARID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateLicenseModel_1_Request'> <wsdl:part name='licModelID' element='ns0:licModelID'/> <wsdl:part name='licModelNew' element='ns0:licModelNew'/> </pre>
--	--

	<pre> </wsdl:message> <wsdl:message name='LicenseDBManager_insertIPAR_1_Request'> <wsdl:part name='IPAR' element='ns0:IPAR'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateIPARStatus_1_Request'> <wsdl:part name='IPARID' element='ns0:IPARID'/> <wsdl:part name='Status' element='ns0:Status'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateLicenseModel_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateIPARStatus_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updatePAR_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_deleteLicenseModel_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrievePAR_1_Request'> <wsdl:part name='PARID' element='ns0:PARID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveLicenseIssuers_1_Request'> <wsdl:part name='AXOID' element='ns0:AXOID'/> <wsdl:part name='Query' element='ns0:Query'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveLicenseModel_Response'> <wsdl:part name='response' element='ns0:string_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_deleteIPAR_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_revokeAddLicense_1_Request'> <wsdl:part name='licenseID' element='ns0:licenseID'/> <wsdl:part name='newLicense' element='ns0:newLicense'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateIPARLicensingURL_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveAXOIDIPARs_1_Request'> <wsdl:part name='Query' element='ns0:Query'/> </wsdl:message> <wsdl:message name='LicenseDBManager_deletePAR_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_insertIPAR_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updatePARStatus_1_Request'> <wsdl:part name='PARID' element='ns0:PARID'/> <wsdl:part name='Status' element='ns0:Status'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updatePARLicensingURL_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_insertLicense_1_Request'> <wsdl:part name='license' element='ns0:license'/> </wsdl:message> <wsdl:message name='LicenseDBManager_isInternalPAR_1_Request'> <wsdl:part name='PARID' element='ns0:PARID'/> </wsdl:message> <wsdl:message name='LicenseDBManager_revokeAddLicense_Response'> <wsdl:part name='response' element='ns0:boolean_Response'/> </wsdl:message> <wsdl:message name='LicenseDBManager_updateIPARLicensingURL_1_Request'> <wsdl:part name='IPARID' element='ns0:IPARID'/> <wsdl:part name='LicensingURL' element='ns0:LicensingURL'/> </wsdl:message> <wsdl:message name='LicenseDBManager_retrieveIPAR_1_Request'> </pre>
--	---

	<pre> <wsdl:part name='IPARID' element='ns0:IPARID'/> </wsdl:message> <wsdl:portType name='LicenseDBManager'> <wsdl:operation name='insertLicense' parameterOrder='license'> <wsdl:input message='tns:LicenseDBManager_insertLicense_1_Request'/> <wsdl:output message='tns:LicenseDBManager_insertLicense_Response'/> </wsdl:operation> <wsdl:operation name='insertPAR' parameterOrder='PAR'> <wsdl:input message='tns:LicenseDBManager_insertPAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_insertPAR_Response'/> </wsdl:operation> <wsdl:operation name='insertIPAR' parameterOrder='IPAR'> <wsdl:input message='tns:LicenseDBManager_insertIPAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_insertIPAR_Response'/> </wsdl:operation> <wsdl:operation name='insertLicenseModel' parameterOrder='licenseModel'> <wsdl:input message='tns:LicenseDBManager_insertLicenseModel_1_Request'/> <wsdl:output message='tns:LicenseDBManager_insertLicenseModel_Response'/> </wsdl:operation> <wsdl:operation name='retrieveLicense' parameterOrder='AXLID'> <wsdl:input message='tns:LicenseDBManager_retrieveLicense_1_Request'/> <wsdl:output message='tns:LicenseDBManager_retrieveLicense_Response'/> </wsdl:operation> <wsdl:operation name='retrievePAR' parameterOrder='PARID'> <wsdl:input message='tns:LicenseDBManager_retrievePAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_retrievePAR_Response'/> </wsdl:operation> <wsdl:operation name='retrieveIPAR' parameterOrder='IPARID'> <wsdl:input message='tns:LicenseDBManager_retrieveIPAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_retrieveIPAR_Response'/> </wsdl:operation> <wsdl:operation name='retrieveLicenseModel' parameterOrder='licModelID'> <wsdl:input message='tns:LicenseDBManager_retrieveLicenseModel_1_Request'/> <wsdl:output message='tns:LicenseDBManager_retrieveLicenseModel_Response'/> </wsdl:operation> <wsdl:operation name='retrieveLicenseIssuers' parameterOrder='AXOID Query'> <wsdl:input message='tns:LicenseDBManager_retrieveLicenseIssuers_1_Request'/> <wsdl:output message='tns:LicenseDBManager_retrieveLicenseIssuers_Response'/> </wsdl:operation> <wsdl:operation name='retrieveAXOIDPARs' parameterOrder='Query'> <wsdl:input message='tns:LicenseDBManager_retrieveAXOIDPARs_1_Request'/> <wsdl:output message='tns:LicenseDBManager_retrieveAXOIDPARs_Response'/> </wsdl:operation> <wsdl:operation name='retrieveAXOIDIPARs' parameterOrder='Query'> <wsdl:input message='tns:LicenseDBManager_retrieveAXOIDIPARs_1_Request'/> <wsdl:output message='tns:LicenseDBManager_retrieveAXOIDIPARs_Response'/> </wsdl:operation> <wsdl:operation name='updatePAR' parameterOrder='PARID NewPAR'> <wsdl:input message='tns:LicenseDBManager_updatePAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_updatePAR_Response'/> </wsdl:operation> <wsdl:operation name='updateLicenseStatus' parameterOrder='AXLID Status NewAXLID'> <wsdl:input message='tns:LicenseDBManager_updateLicenseStatus_1_Request'/> <wsdl:output message='tns:LicenseDBManager_updateLicenseStatus_Response'/> </wsdl:operation> <wsdl:operation name='updatePARStatus' parameterOrder='PARID Status'> <wsdl:input message='tns:LicenseDBManager_updatePARStatus_1_Request'/> <wsdl:output message='tns:LicenseDBManager_updatePARStatus_Response'/> </wsdl:operation> <wsdl:operation name='updateIPARStatus' parameterOrder='IPARID Status'> <wsdl:input message='tns:LicenseDBManager_updateIPARStatus_1_Request'/> <wsdl:output message='tns:LicenseDBManager_updateIPARStatus_Response'/> </wsdl:operation> <wsdl:operation name='updatePARLicensingURL' parameterOrder='PARID LicensingURL'> <wsdl:input message='tns:LicenseDBManager_updatePARLicensingURL_1_Request'/> <wsdl:output message='tns:LicenseDBManager_updatePARLicensingURL_Response'/> </wsdl:operation> <wsdl:operation name='updateIPARLicensingURL' parameterOrder='IPARID LicensingURL'> <wsdl:input message='tns:LicenseDBManager_updateIPARLicensingURL_1_Request'/> <wsdl:output message='tns:LicenseDBManager_updateIPARLicensingURL_Response'/> </wsdl:operation> </pre>
--	--

	<pre> </wsdl:operation> <wsdl:operation name='updateLicenseModel' parameterOrder='licModelID licModelNew'> <wsdl:input message='tns:LicenseDBManager_updateLicenseModel_1_Request'/> <wsdl:output message='tns:LicenseDBManager_updateLicenseModel_Response'/> </wsdl:operation> <wsdl:operation name='deleteLicenseModel' parameterOrder='licModelID'> <wsdl:input message='tns:LicenseDBManager_deleteLicenseModel_1_Request'/> <wsdl:output message='tns:LicenseDBManager_deleteLicenseModel_Response'/> </wsdl:operation> <wsdl:operation name='deletePAR' parameterOrder='PARID'> <wsdl:input message='tns:LicenseDBManager_deletePAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_deletePAR_Response'/> </wsdl:operation> <wsdl:operation name='deleteIPAR' parameterOrder='PARID'> <wsdl:input message='tns:LicenseDBManager_deleteIPAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_deleteIPAR_Response'/> </wsdl:operation> <wsdl:operation name='isInternalPAR' parameterOrder='PARID'> <wsdl:input message='tns:LicenseDBManager_isInternalPAR_1_Request'/> <wsdl:output message='tns:LicenseDBManager_isInternalPAR_Response'/> </wsdl:operation> <wsdl:operation name='revokeLicense' parameterOrder='licenseID'> <wsdl:input message='tns:LicenseDBManager_revokeLicense_1_Request'/> <wsdl:output message='tns:LicenseDBManager_revokeLicense_Response'/> </wsdl:operation> <wsdl:operation name='revokeAddLicense' parameterOrder='licenseID newLicense'> <wsdl:input message='tns:LicenseDBManager_revokeAddLicense_1_Request'/> <wsdl:output message='tns:LicenseDBManager_revokeAddLicense_Response'/> </wsdl:operation> </wsdl:portType> <wsdl:binding name='LicenseDBManager' type='tns:LicenseDBManager'> <soap:binding transport='http://schemas.xmlsoap.org/soap/http' style='document'/> <wsdl:operation name='insertLicense'> <map:java-operation name='insertLicense' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#insertLicense#KExqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='insertPAR'> <map:java-operation name='insertPAR' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#insertPAR#KExqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='insertIPAR'> <map:java-operation name='insertIPAR' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#insertIPAR#KExqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='insertLicenseModel'> <map:java-operation name='insertLicenseModel' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'> </pre>
--	---

	<pre> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#insertLicenseModel#KExqYXZhL2xhbmvcU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='retrieveLicense'> <map:java-operation name='retrieveLicense' signature='KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs='/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#retrieveLicense#KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='retrievePAR'> <map:java-operation name='retrievePAR' signature='KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs='/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#retrievePAR#KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='retrieveIPAR'> <map:java-operation name='retrieveIPAR' signature='KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs='/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#retrieveIPAR#KExqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=' 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:LicenseDBManagerLicenseDBManager#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='retrieveLicenseIssuers'> <map:java-operation name='retrieveLicenseIssuers' signature='KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs='/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#retrieveLicenseIssuers#KExqYXZhL2xhbmvcU3RyaW5nO0xqYXZhL2xhbmvcU3RyaW5nOylMamF2YS9sYW5nL1N0cmLuZzs=' style='document'/> <wsdl:input> </pre>
--	---

	<pre> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='retrieveAXOIDPARs'> <map:java-operation name='retrieveAXOIDPARs' signature='KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs='/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#retrieveAXOIDPARs#KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='retrieveAXOIDIPARs'> <map:java-operation name='retrieveAXOIDIPARs' signature='KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs='/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#retrieveAXOIDIPARs#KExqYXZhL2xhbmcvU3RyaW5nOylMamF2YS9sYW5nL1N0cmIuZzs=' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='updatePAR'> <map:java-operation name='updatePAR' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla'/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#updatePAR#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='updateLicenseStatus'> <map:java-operation name='updateLicenseStatus' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla'/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#updateLicenseStatus#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='updatePARStatus'> <map:java-operation name='updatePARStatus' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla'/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#updatePARStatus#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' 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='updateIPARStatus'> <map:java-operation name='updateIPARStatus' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' /> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#updateIPARStatus#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' style='document' /> <wsdl:input> <soap:body use='literal' /> </wsdl:input> <wsdl:output> <soap:body use='literal' /> </wsdl:output> </wsdl:operation> <wsdl:operation name='updatePARLicensingURL'> <map:java-operation name='updatePARLicensingURL' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' /> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#updatePARLicensingURL#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' style='document' /> <wsdl:input> <soap:body use='literal' /> </wsdl:input> <wsdl:output> <soap:body use='literal' /> </wsdl:output> </wsdl:operation> <wsdl:operation name='updateIPARLicensingURL'> <map:java-operation name='updateIPARLicensingURL' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' /> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#updateIPARLicensingURL#KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' 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' signature='KExqYXZhL2xhbmcvU3RyaW5nO0xqYXZhL2xhbmcvU3RyaW5nOyla' /> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#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:LicenseDBManagerLicenseDBManager#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='deletePAR'> <map:java-operation name='deletePAR' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla' /> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#deletePAR#KExqYXZhL2xhbmcvU3RyaW5nOyla' </pre>
--	--

	<pre> style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='deleteIPAR'> <map:java-operation name='deleteIPAR' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#deleteIPAR#KExqYXZhL2xhbmcvU3RyaW5nOyla' style='document'/> <wsdl:input> <soap:body use='literal'/> </wsdl:input> <wsdl:output> <soap:body use='literal'/> </wsdl:output> </wsdl:operation> <wsdl:operation name='isInternalPAR'> <map:java-operation name='isInternalPAR' signature='KExqYXZhL2xhbmcvU3RyaW5nOyla'/> <soap:operation soapAction='urn:LicenseDBManagerLicenseDBManager#isInternalPAR#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:LicenseDBManagerLicenseDBManager#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:LicenseDBManagerLicenseDBManager#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:service name='LicenseDBManager'> <wsdl:port name='LicenseDBManager' binding='tns:LicenseDBManager'> <soap:address location='http://cjlal1:6060/LicenseDBManager'/> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre>
Method	InsertLicense
Description	This function inserts a license in the License DB
Request	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" </pre>

Sample Message	<pre> 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;license/&gt;</wn0:license> </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 license: It contains the whole XML document representing the license to be stored in the database
Output parameters	xsd:boolean: This function returns true if the insert has been finished successfully
Method	InsertPAR
Description	This function inserts a PAR in the PAR DB
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;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:IPAR i:type="d:string">&lt;license/&gt;</wn0:IPAR> </e:Body> </e:Envelope> </pre>
Input parameters	xsd:string PAR: It contains the whole XML document representing the license of the PAR to be stored in the database
Output parameters	xsd:boolean: This function returns true if the insert has been finished successfully
Method	InsertIPAR
Description	This function inserts an internal PAR in the PAR DB
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:IPAR i:type="d:string">&lt;license/&gt;</wn0:IPAR> </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 IPAR: It contains the whole XML document representing the license of the internal PAR to be stored in the database
Output parameters	xsd:boolean: This function returns true if the insert has been finished successfully

Method	insertLicenseModel
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;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	retrieveLicense
Description	This function returns a License stored in the License DB
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:AXLID i:type="d:string">AXLID</wn0:AXLID> </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:nil="true"/> </e:Body> </e:Envelope></pre>
Input parameters	xsd:string AXLID: This parameter contains the identifier of the license that has to be retrieved
Output parameters	xsd:string: This function returns a String with the whole XML document representing the License
Method	retrievePAR
Description	This function returns a PAR stored in the PAR DB
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:PARID i:type="d:string">PARID</wn0:PARID> </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:nil="true"/> </e:Body> </e:Envelope></pre>
Input	xsd:string PARID: This parameter contains the identifier of the PAR that has to be

parameters	retrieved
Output parameters	xsd:string: This function returns a String with the whole XML document representing the License of the PAR
Method	retrieveIPAR
Description	This function returns a internal PAR stored in the PAR DB
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:IPARID i:type="d:string">IPARID</wn0:IPARID> </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:nil="true"/> </e:Body> </e:Envelope></pre>
Input parameters	xsd:string IPARID: This parameter contains the identificator of the internal PAR that has to be retrieved
Output parameters	xsd:string: This function returns a String with the whole XML document representing the License of the internal PAR
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/"> <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;licModel/&gt;</wn0:string_Response> </e:Body> </e:Envelope></pre>
Input parameters	xsd:string licModelID: This parameter contains the identificator 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	retrieveLicenseIssuers
Description	This function returns a set of AXUIDs from distributors that can offer a license for a specific AXOID and that satisfies some conditions expressed in a query.
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:Query i:type="d:string">&lt;query/&gt;</wn0:Query> </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> <e:Body> <wn0:string_Response i:nil="true"/> </e:Body> </e:Envelope> </pre>
Input parameters	<p>xsd:string AXOID: This parameter contains the identifier of the AXMEDIS Object, that has to be searched in the Licenses DB.</p> <p>xsd:string Query: This parameter contains the query to be performed in the License Database. The query is expressed according to que schema for AXMEDIS Queries (Section 4.3.2 in this document).</p>
Output parameters	<p>xsd:string: This function returns a set of AXUIDs that satisfy the conditions of the query. They are expressed according to the schema for AXMEDIS Query Results Queries (Section 4.3.3 in this document).</p>
Method	retrieveAXOIDPARs
Description	This function returns a set of AXOID referenced in PARs that satisfy some conditions expressed in a query
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:Query i:type="d:string">&lt;query/&gt;</wn0:Query> </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:nil="true"/> </e:Body> </e:Envelope> </pre>
Input parameters	<p>xsd:string Query: This parameter contains the query to be performed in the PAR Database. The query is expressed according to the schema for AXMEDIS Queries (Section 4.3.2 in this document).</p>
Output parameters	<p>xsd:string: This function returns a set of AXOIDs that satisfy the conditions of the query. They are expressed according to the schema for AXMEDIS Query Results Queries (Section 4.3.3 in this document).</p>
Method	retrieveAXOIDIPARs
Description	This function returns a set of AXOID referenced in internal PARs that satisfy some conditions expressed in a query
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:PARID i:type="d:string">PARID</wn0:PARID> <wn0:NewPAR i:type="d:string">&lt;license/&gt;</wn0:NewPAR> </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 Query: This parameter contains the query to be performed in the PAR Database (over the internal PARs). The query is expressed according to the schema for AXMEDIS Queries (Section 4.3.2 in this document).</p>

Output parameters	xsd:string: This function returns a set of AXOIDs that satisfy the conditions of the query. They are expressed according to the schema for AXMEDIS Query Results Queries (Section 4.3.3 in this document).
Method	updatePAR
Description	This function updates the PAR given a PARID and a new description of 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:PARID i:type="d:string">PARID</wn0:PARID> <wn0:NewPAR i:type="d:string">&lt;license&gt;</wn0:NewPAR> </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 PARID: It contains the identifier of the PAR to be updated</p> <p>xsd:string PAR: It contains the whole XML document representing the license of the PAR to be stored in the database</p>
Output parameters	xsd:boolean: This function returns true if the update has been finished successfully
Method	updateLicenseStatus
Description	This function updates the status of the license in the License DB
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:AXLID i:type="d:string">AXLID</wn0:AXLID> <wn0:Status i:type="d:string">Valid</wn0:Status> <wn0:NewAXLID i:nil="true"/> </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 AXLID: It contains the identifier of the license to be updated</p> <p>xsd:string Status: It contains the new status of the license. It can be valid, revoked, invalid. In case of revocation, the substituting license identifier can be given</p> <p>xsd:string newAXLID: It contains the identifier of the license substituting it in case of revocation</p>
Output parameters	xsd:boolean: This function returns true if the update has been finished successfully
Method	updatePARStatus
Description	This function updates the status of the PAR in the PAR DB
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/"></pre>

	<pre> <e:Body> <wn0:PARID i:type="d:string">PARID</wn0:PARID> <wn0:Status i:type="d:string">Valid</wn0:Status> </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 PARID: It contains the identifier of the PAR to be updated</p> <p>xsd:string Status: It contains the new status of the PAR. It can be valid, invalid.</p>
Output parameters	xsd:boolean: This function returns true if the update has been finished successfully
Method	updateIPARStatus
Description	This function updates the status of the IPAR in the PAR DB
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:IPARID i:type="d:string">IPARID</wn0:IPARID> <wn0:Status i:type="d:string">Valid</wn0:Status> </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 IPARID: It contains the identifier of the IPAR to be updated</p> <p>xsd:string Status: It contains the new status of the IPAR. It can be valid, invalid.</p>
Output parameters	xsd:boolean: This function returns true if the update has been finished successfully
Method	updatePARLicensingURL
Description	This function updates the PARLicensingURL in the PAR DB
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:PARID i:type="d:string">PARID</wn0:PARID> <wn0:LicensingURL i:type="d:string">http://www.axmedis.org/licensing</wn0:LicensingURL> </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 PARID: It contains the identifier of the PAR to be updated</p> <p>xsd:string LicensingURL: It contains the new licensing URL for this PAR</p>
Output	xsd:boolean: This function returns true if the update has been finished successfully

parameters	
Method	updateIPARLicensingURL
Description	This function updates the IPARLicensingURL in the PAR DB
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:IPARID i:type="d:string">IPARID</wn0:IPARID> <wn0:LicensingURL i:type="d:string">http://www.axmedis.org/licensing</wn0:LicensingURL> </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 IPARID: It contains the identifier of the IPAR to be updated xsd:string LicensingURL: It contains the new licensing URL for this IPAR
Output parameters	xsd:boolean: This function returns true if the update 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;it;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	xsd:string licModelID: It contains the identifier of the licenseModel to be updated xsd:string licModelNew: It contains the new licenseModel
Output parameters	xsd:boolean: This function returns true if the update 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></pre>

	<pre> <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	deletePAR
Description	This function deletes a PAR from the PAR 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:PARID i:type="d:string">PARID001</wn0:PARID> </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 PAR to be deleted
Output parameters	xsd:boolean: This function returns true if the delete has been finished successfully
Method	deleteIPAR
Description	This function deletes an IPAR from the PAR 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:PARID i:type="d:string">IPARID001</wn0:PARID> </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 IPARID: It contains the identifier of the IPAR to be deleted
Output parameters	xsd:boolean: This function returns true if the delete has been finished successfully
Method	isInternalPAR
Description	
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:PARID i:type="d:string">PARID</wn0:PARID> </e:Body> </e:Envelope> </pre>
Response	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <e:Envelope xmlns:e="http://schemas.xmlsoap.org/soap/envelope/" </pre>

Sample Message	<pre> 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 PAR for which we want to check if it is internal or not
Output parameters	xsd:boolean: This function returns true if the PAR is internal, if not, it returns false
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;it;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

3.11 Automatic generation of contracts and licenses and feature extraction from Contracts

There are four key objectives in this module, that correspond to four different software tools:

- Generate license templates (or license models) from PARs (Possible Available Rights).
- Generate license templates from contracts.
- Generate licenses from contracts.
- Generate contracts from licenses.

The generated licenses and templates will be stored in the licenses repository. Contracts will be a possible output from the module.

For achieving the aforementioned objectives, this module will include activities for:

- Analysing existing contracts and procedures in order to extract and formalise standard and frequent information included in contracts. This includes both general terms and specific clauses. This will facilitate the creation of license models, or even specific licenses, from contracts. In turn, this process will imply:
 - The information extracted from contracts will be formalised as DRM rules. They will be mapped into electronic licenses, following a standard license format, that could be either MPEG-21 REL or ODRL. The result of this activity could lead to the definition of a subset of a standard RELs, that could be contributed to international standardisation in order to specify a profile. Other situation that might happen is that a specific profile appears before we accomplish this task, so we should consider in that case if it is worth following it.
 - The previous process (from contracts to licenses) will be generalised as much as possible in order to specify generic models for licenses, which would facilitate the process of analysing new contracts, generating new rules and defining the corresponding licenses. Formal techniques, such as semantic web ontologies could be used to help expressing part of the semantics of the different licenses, for example to solve language barriers.
- The legal implications of the specified electronic licenses will be formalised. When dealing with transformation from contracts to licenses there are a series of legal implications that should be clarified.
- The different possible rights (PAR) and conditions will be specified and structured based on requirements and current practices. By doing this, it will be easier to generate license templates from PARs. Since PARs are a first level of restrictions to a digital item usage, they could be a starting point (once converted to a license template) for creating a specific license.
- The analysis of existing contracts and procedures will facilitate the development of a tool for generating paper contracts from specific licenses.

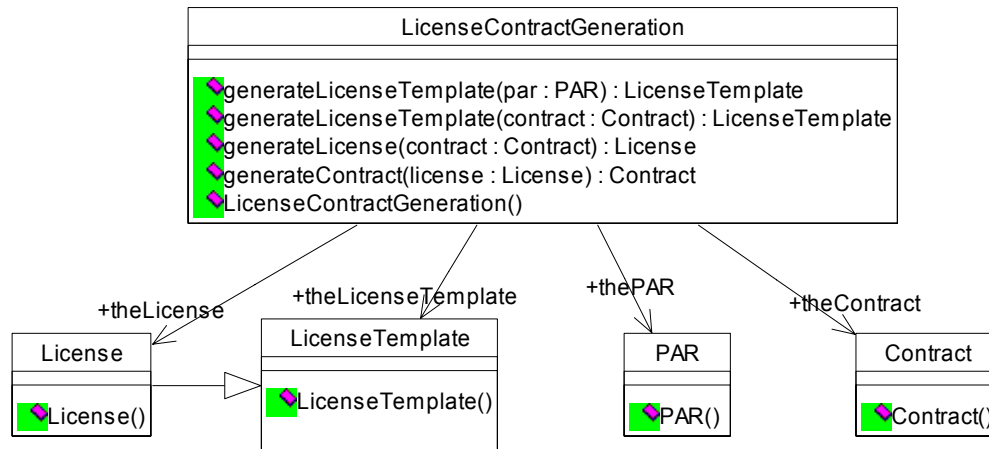
3.11.1 Module profile definition

Module Profile	
Model for Contracts and automatic generation of contracts and feature extraction	
Executable or Library(Support)	Executable or Library(Support)
Single Thread or Multithread	Single Thread or Multithread
Language of Development	Language of Development
Responsible Name	Responsible Name
Responsible Partner	Responsible Partner
Status (proposed/approved)	Status (proposed/approved)

Platforms supported	Platforms supported	
Interfaces with other tools:	Interfaces with other tools:	Interfaces with other tools:
File Formats Used	File Formats Used	File Formats Used
XML based (MPEG-21 REL)	XML based (MPEG-21 REL)	XML based (MPEG-21 REL)
User Interface	User Interface	User Interface
Used Libraries	Used Libraries	Used Libraries
Xerces	Xerces	Xerces
Xalan	Xalan	Xalan
XPath	XPath	XPath

3.11.2 Architecture of the module

The following figure shows the UML diagram of the module.



UML Diagram of the Generation of licenses from contracts module

LicenseContractGeneration: This class provides the functionality for generating licenses, license templates, PAR (Potential available rights) and contracts.

LicenseContractGeneration	
Method	Description
generateLicenseTemplate	This method generates a license template from a PAR description
generateLicenseTemplate	This method generates a license template from the information contained in a contract
generateLicense	This method generates a license from the information contained a contract
generateContract	This method generates a tentative contract from the information contained inside a license

The rest of classes will be the same as defined for license and PAR. LicenseTemplate will have the same structure as license class, but some fields would be optional.

LicenseContractGeneration	
Method	Description
<i>generateLicenseTemplate</i>	This method generates a license template from a PAR description
<i>generateLicenseTemplate</i>	This method generates a license template from the information contained in a contract
<i>generateLicense</i>	This method generates a license from the information contained a contract
<i>generateContract</i>	This method generates a tentative contract from the information contained inside a license

LicenseContractGeneration	
WSDL	<pre> <?xml version="1.0"?> <wsdl:definitions xmlns:tns="urn:LicenseContractGeneration" 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:LicenseContractGeneration" name="LicenseContractGeneration"> <wsdl:types> <xsd:schema elementFormDefault="qualified" targetNamespace="http://systinet.com/xsd/SchemaTypes/" xmlns:tns="http://systinet.com/xsd/SchemaTypes/" xmlns:xns4="http://systinet.com/wsdl/default/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <xsd:import namespace="http://systinet.com/wsdl/default"/> <xsd:element name="contract" nillable="true" type="xns4:Contract"/> <xsd:element name="LicenseTemplate_Response" nillable="true" type="xns4:LicenseTemplate"/> <xsd:element name="par" nillable="true" type="xns4:PAR"/> <xsd:element name="License_Response" nillable="true" type="xns4:License"/> <xsd:element name="license" nillable="true" type="xns4:License"/> <xsd:element name="Contract_Response" nillable="true" type="xns4:Contract"/> </xsd:schema> <xsd:schema elementFormDefault="qualified" targetNamespace="http://systinet.com/wsdl/default/" xmlns:map="http://systinet.com/mapping/" xmlns:tns="http://systinet.com/wsdl/default/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <xsd:complexType name="Contract"> <xsd:sequence/> </xsd:complexType> <xsd:complexType name="LicenseTemplate"> <xsd:sequence/> </xsd:complexType> <xsd:complexType name="PAR"> <xsd:sequence/> </xsd:complexType> <xsd:complexType name="License"> <xsd:complexContent> <xsd:extension base="tns:LicenseTemplate"> <xsd:sequence/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:schema> </wsdl:types> <wsdl:message name="LicenseContractGeneration_generateLicenseTemplate_1_Request"> <wsdl:part name="contract" element="ns0:contract"/> </wsdl:message> <wsdl:message name="LicenseContractGeneration_generateLicenseTemplate_1_Response"> <wsdl:part name="response" element="ns0:LicenseTemplate_Response"/> </wsdl:message> <wsdl:message name="LicenseContractGeneration_generateLicenseTemplate_2_Request"> </pre>

```

<wsdl:part name="par" element="ns0:par"/>
</wsdl:message>

<wsdl:message name="LicenseContractGeneration_generateLicenseTemplate_2_Response">
  <wsdl:part name="response" element="ns0:LicenseTemplate_Response"/>
</wsdl:message>

<wsdl:message name="LicenseContractGeneration_generateLicense_3_Request">
  <wsdl:part name="contract" element="ns0:contract"/>
</wsdl:message>

<wsdl:message name="LicenseContractGeneration_generateLicense_Response">
  <wsdl:part name="response" element="ns0:License_Response"/>
</wsdl:message>

<wsdl:message name="LicenseContractGeneration_generateContract_3_Request">
  <wsdl:part name="license" element="ns0:license"/>
</wsdl:message>

<wsdl:message name="LicenseContractGeneration_generateContract_Response">
  <wsdl:part name="response" element="ns0:Contract_Response"/>
</wsdl:message>

<wsdl:portType name="LicenseContractGeneration">
  <wsdl:operation name="generateLicenseTemplate" parameterOrder="contract">
    <wsdl:input name="generateLicenseTemplate_1_input"
message="tns:LicenseContractGeneration_generateLicenseTemplate_1_Request"/>
    <wsdl:output name="generateLicenseTemplate_1_output"
message="tns:LicenseContractGeneration_generateLicenseTemplate_1_Response"/>
  </wsdl:operation>
  <wsdl:operation name="generateLicenseTemplate" parameterOrder="par">
    <wsdl:input name="generateLicenseTemplate_2_input"
message="tns:LicenseContractGeneration_generateLicenseTemplate_2_Request"/>
    <wsdl:output name="generateLicenseTemplate_2_output"
message="tns:LicenseContractGeneration_generateLicenseTemplate_2_Response"/>
  </wsdl:operation>
  <wsdl:operation name="generateLicense" parameterOrder="contract">
    <wsdl:input message="tns:LicenseContractGeneration_generateLicense_3_Request"/>
    <wsdl:output message="tns:LicenseContractGeneration_generateLicense_Response"/>
  </wsdl:operation>
  <wsdl:operation name="generateContract" parameterOrder="license">
    <wsdl:input message="tns:LicenseContractGeneration_generateContract_3_Request"/>
    <wsdl:output message="tns:LicenseContractGeneration_generateContract_Response"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="LicenseContractGeneration" type="tns:LicenseContractGeneration">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="generateLicenseTemplate">
    <map:java-operation name="generateLicenseTemplate"
signature="KExDb250cmFjdDspTExpY2Vuc2VUZW1wbGF0ZTs="/>
    <soap:operation
soapAction="urn:LicenseContractGenerationLicenseContractGeneration#generateLicenseTemplate#KExDb250cmFjdDspTExpY2Vuc2VUZW1wbGF0ZTs=" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="generateLicenseTemplate">
    <map:java-operation name="generateLicenseTemplate"
signature="KExQQVl7KUxMaWNlbnNlVGltGxhdGU7"/>
    <soap:operation
soapAction="urn:LicenseContractGenerationLicenseContractGeneration#generateLicenseTemplate#KExQQVl7KUxMaWNlbnNlVGltGxhdGU7" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>

```

	<pre> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateLicense"> <map:java-operation name="generateLicense" signature="KExDb250cmFjdDspTExpY2Vuc2U7"/> <soap:operation soapAction="urn:LicenseContractGenerationLicenseContractGeneration#generateLicense#KExDb250cmFjdDspTExpY2Vuc2U7" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="generateContract"> <map:java-operation name="generateContract" signature="KExMaWNlbnNIOyIMQ29udHJhY3Q7"/> <soap:operation soapAction="urn:LicenseContractGenerationLicenseContractGeneration#generateContract#KExMaWNlbnNIOyIMQ29udHJhY3Q7" 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="LicenseContractGeneration"> <wsdl:port name="LicenseContractGeneration" binding="tns:LicenseContractGeneration"> <soap:address location="http://slim:6060/LicenseContractGeneration/"> </wsdl:port> </wsdl:service> </wsdl:definitions> </pre>
--	---

LicenseContractGeneration	
Method	<i>generateLicenseTemplate</i>
Description	This method generates a license template from a PAR description
Input parameters	PAR
Output parameters	LicenseTemplate
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"> <e:Body> <wn0:par i:type="wn1: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/" xmlns:wn1="http://systinet.com/wsdl/default"> <e:Body> <wn0:LicenseTemplate_Response i:nil="true"/> </e:Body> </e:Envelope> </pre>
Method	<i>generateLicenseTemplate</i>
Description	This method generates a license template from the information contained in a contract
Input	Contract

parameters	
Output parameters	LicenseTemplate
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/"> <e:Body> <wn0:contract i:type="wn1:Contract"/> </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/"> <e:Body> <wn0:contract i:type="wn1:Contract"/> </e:Body> </e:Envelope></pre>
Method	<i>generateLicense</i>
Description	This method generates a license from the information contained a contract
Input parameters	Contract
Output parameters	License
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/"> <e:Body> <wn0:contract i:type="wn1:Contract"/> </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/"> <e:Body> <wn0:License_Response i:nil="true"/> </e:Body> </e:Envelope></pre>
Method	<i>generateContract</i>
Description	This method generates a tentative contract from the information contained inside a license
Input parameters	License
Output parameters	Contract
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/"> <e:Body> <wn0:license i:type="wn1:License"/> </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/"></pre>

	<pre> <e:Body> <wn0:Contract_Response i:nil="true"/> </e:Body> </e:Envelope> </pre>
--	---

3.12 History of AXMEDIS Objects Evolution in Production Repository (EXITECH, FUPF)

For all the AXMEDIS objects it is necessary to track and store different versions of each object. Object can differ for several aspects. The main aspects are:

- Content itself that has changed (the DIDL is not modified but the external reference is a different object);
- Part of the content (also the DIDL can be modified if a new Item is added for example);
- Metadata associated to the object (i.e. DIDL is the same, but AXINFO is changed).

AXMEDIS has to track all version and has to be capable of having immediately available the last version of the object for indexing and for fast retrieval, and has to be capable of retrieving one of the previous version of the object, also if not indexed in the database.

It is necessary, in the Database Interface, to have methods for listing versions, recovering different versions of an object and to update objects generating a new version.

It is necessary to distinguish between the different situation in which a version is updated:

1. The content of the object has been modified, but apart from the external reference no other change has been introduced: in this case a new version is created with reference to the new content, while the previous object (full AXMEDIS object) with the previous external references are tracked in the system;
2. Part of the content and at least the DIDL are changed: the object is imported in the system main database in order to have the indexing of the new object online. The old version is stored together with the external references;
3. The AXINFO or metadata have been changed: the new metadata are mapped in the DB and the old object is stored together with the old external references.

The problem related to the name of duplicated external references has to be considered, since if I have for example the content FlorencePicture.jpg that is part of one AXMEDIS object and I have updated such object with a new version of the FlorencePicture.jpg (maybe different in resolution, filtering or whatever) keeping the same name, it is necessary to be able to distinguish between these versions in order to give back the correct object.

In the resource is embedded in the XML object the problem do not exists since the whole object is stored together with versioning information related to the versions that are not the current.

It is possible to create a hierarchical structure on the disc in order to store the physical contents; such structure can be very simply implemented by having a main directory with an unique ID (i.e the AXOID) that in the database is referenced to the content with a set of subdirectories in which for each version only the new version of the content are stored; if a version is missing in the hierarchy it means that for that version no update of the content has been made.

```

AXOID ----- 1.0 ----- FlorencePicture.jpg
               |--- 1.3 ----- FlorencePicture.jpg

```

If I want version 1.2 of the previous content I have to see in the 1.0 directory. This mechanism can be implemented at the database level for having a fast access to the object. The solution on the file system is only for databases that are not capable of storing large blobs or are unable to manage huge tables. In general all objects are stored externally and we have to take care of size of embedded object and also of time related to transfer of object for example for transcoding.

All the operations described in this section are in charge to Saver/Indexer module, that is the entry point to the database and therefore the best place where taking care of new versions added to the DB.

3.13 ER diagram extended description

In order to obtain a more manageable document the contents of this section have been moved in a separate document named AXDB-vx-yy.rtf, where x.yy is the last version of the document.

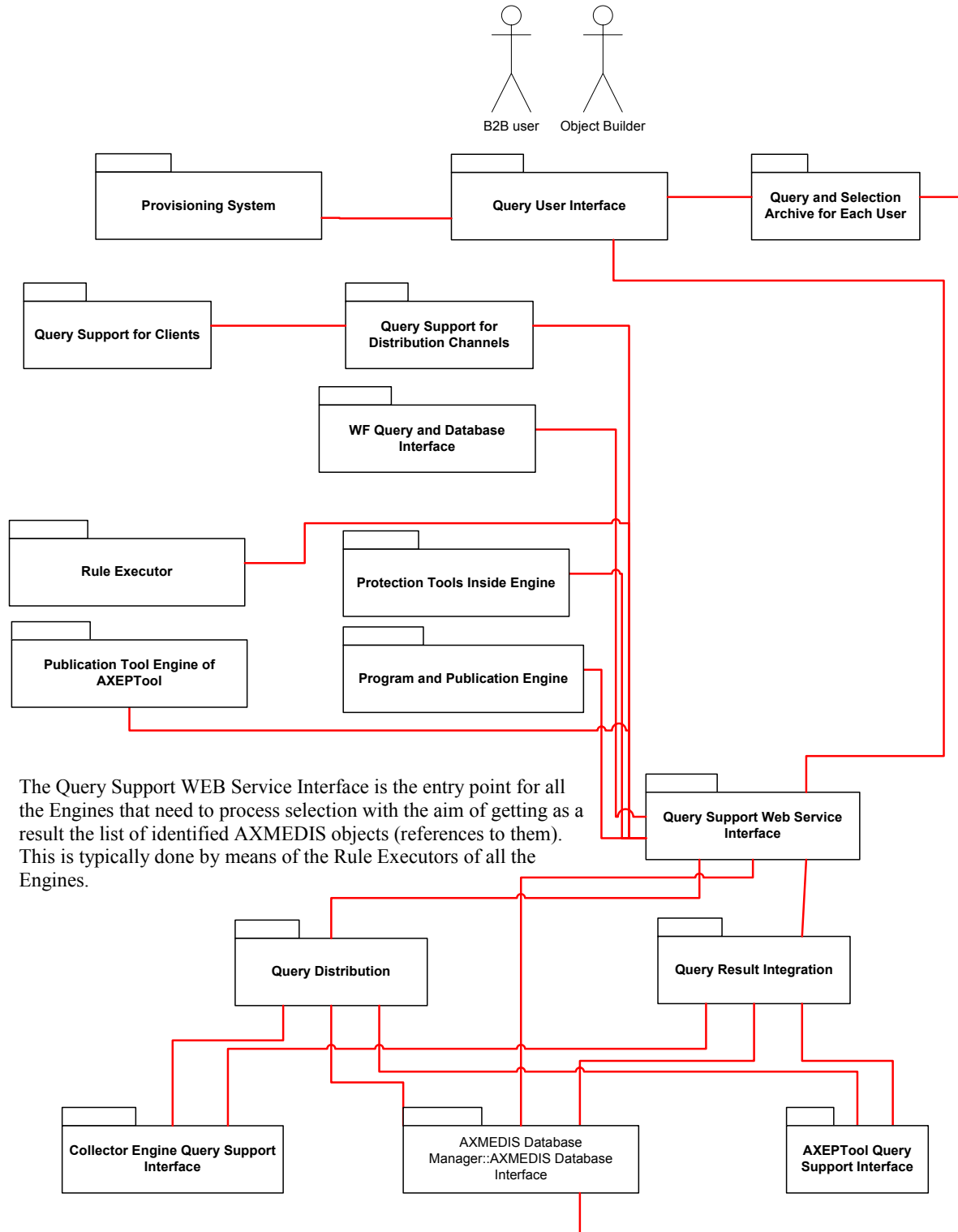
4 AXMEDIS Query Support (EXITECH)

The query support is a combination of query mechanisms for retrieving content objects among the indexed information, in the AXMEDIS database (for content processing, composition, formatting, etc.), on the P2P AXEPTool, on the content collector referencing the content contained in the CMSs (Crawled Results Integrated Database). From the AXMEDIS database interface it has to be possible to make valid queries for all the environments and read unified results. From the AXEPTool or from the Collector the queries can be performed only on their environment.

The query support allows the specification of technical/professional query including metadata, technical information, business and licensing aspects, content based, DRM rules, etc.

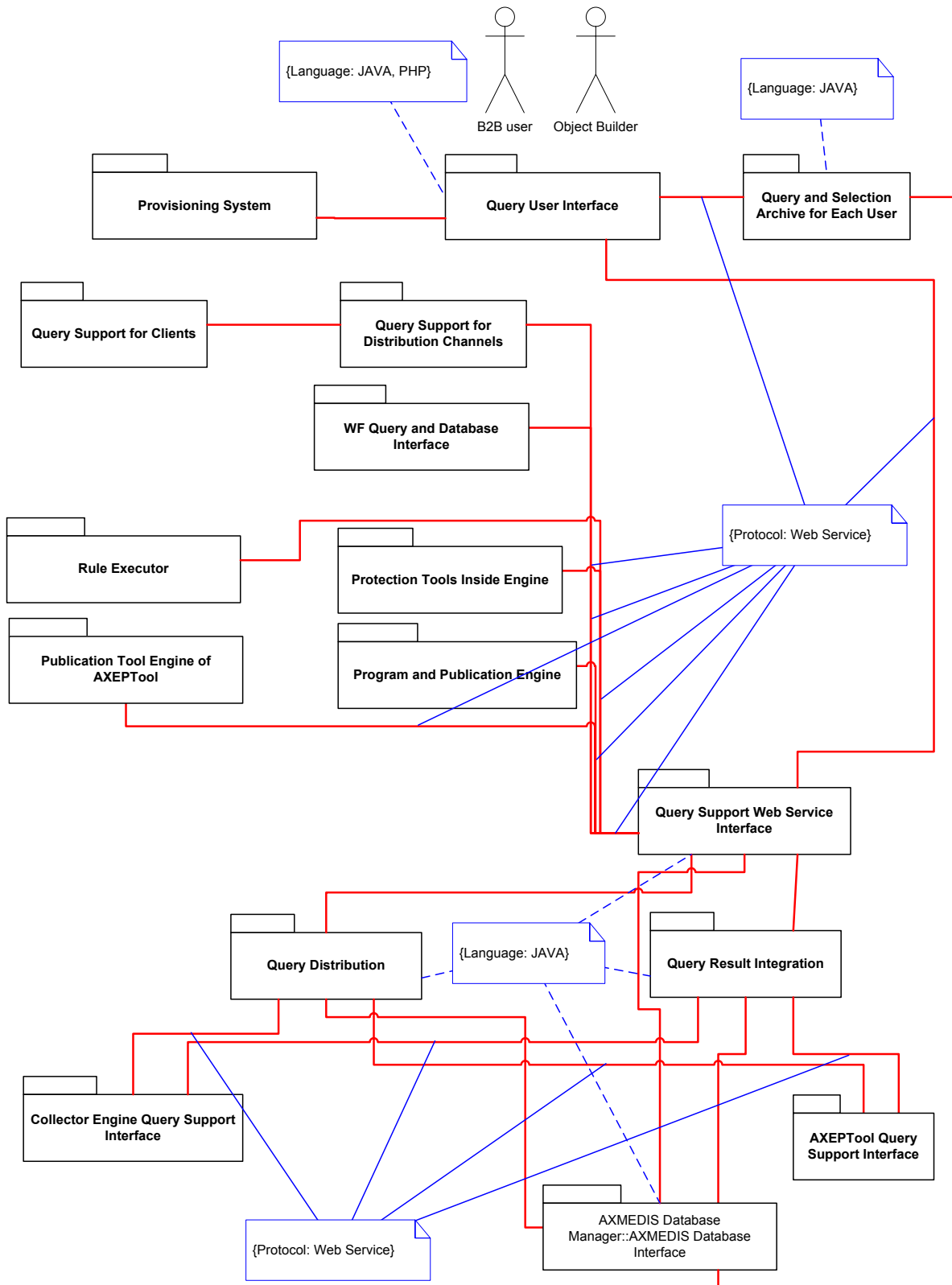
4.1 Query Support general architecture

The general architecture for AXMEDIS query support is depicted in the following picture:



The Query Support WEB Service Interface is the entry point for all the Engines that need to process selection with the aim of getting as a result the list of identified AXMEDIS objects (references to them). This is typically done by means of the Rule Executors of all the Engines.

Some constraints on languages and protocols have been imposed in the following diagram.



In the architecture it is evidenced that the Query support offers a service to the other tools of the system by the means of a Web service that is able to receive query, distribute such queries on the different channels and recollecting aggregated data in order to communicate back the results.

The query Support Web Service Interface communicate with the backend of the AXDB for gathering information directly on the object stored in the AXMEDIS database, while access via a web service interface to the crawling system and to the AXEPTOOL. The web service interface and therefore the WSDL offered by the AXEPTOOL and by the Crawling system must be the same as that offered by Query Support in order to have a unique format for data exchanging.

The Query User Interface access via web service to the Query support in order to send query and gather information.

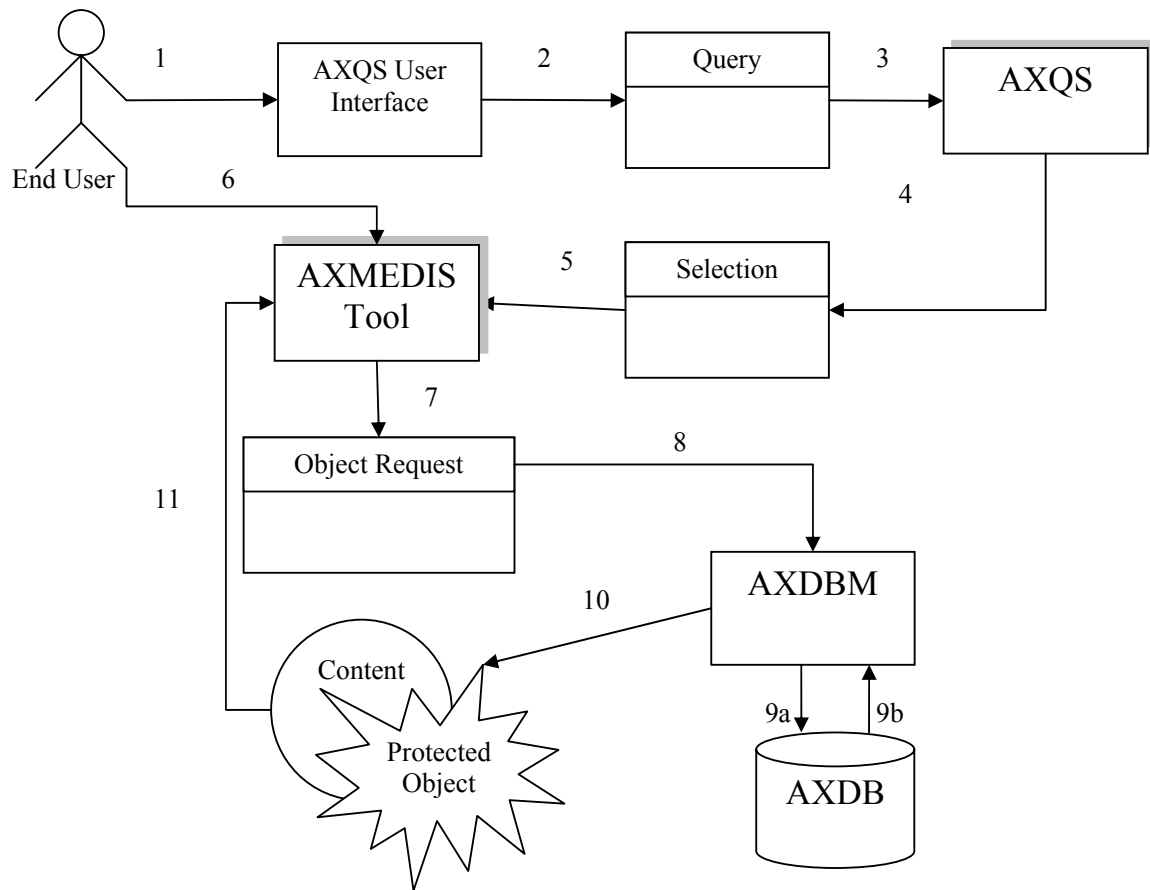
4.2 AXMEDIS Query Support Scenario

The main functionality that have been identified in the system that are also related to the query support are depicted in the scenario reported below.

The main scenario that have been identified are:

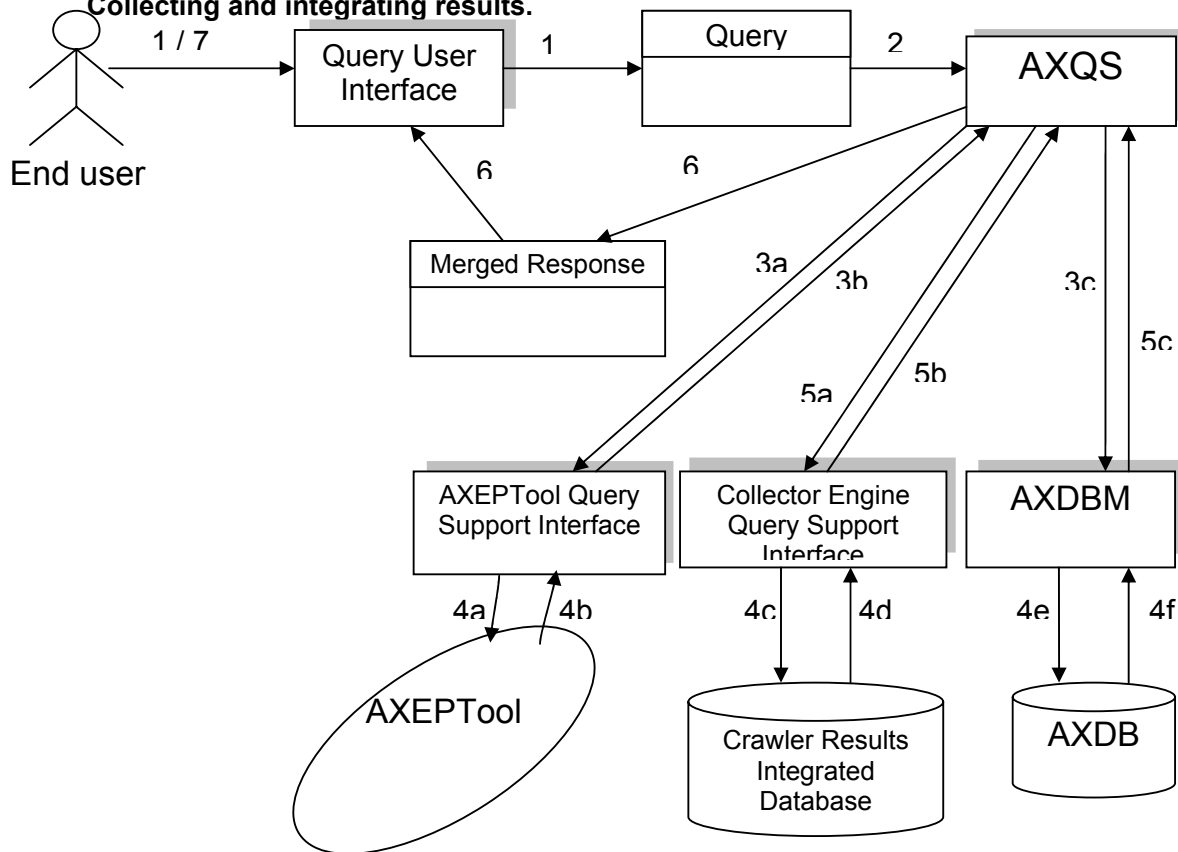
- SCENARIO 6: Content accessibility, querying
- SCENARIO 6.1: Making query on the several direction. Technical Querying, Collecting and integrating results.
- SCENARIO 6.2: Producing a selection.
- SCENARIO 6.5: Store a query/Selection and make it active later

For each scenario a brief description is reported.

4.2.1 SCENARIO 6: Content accessibility, querying

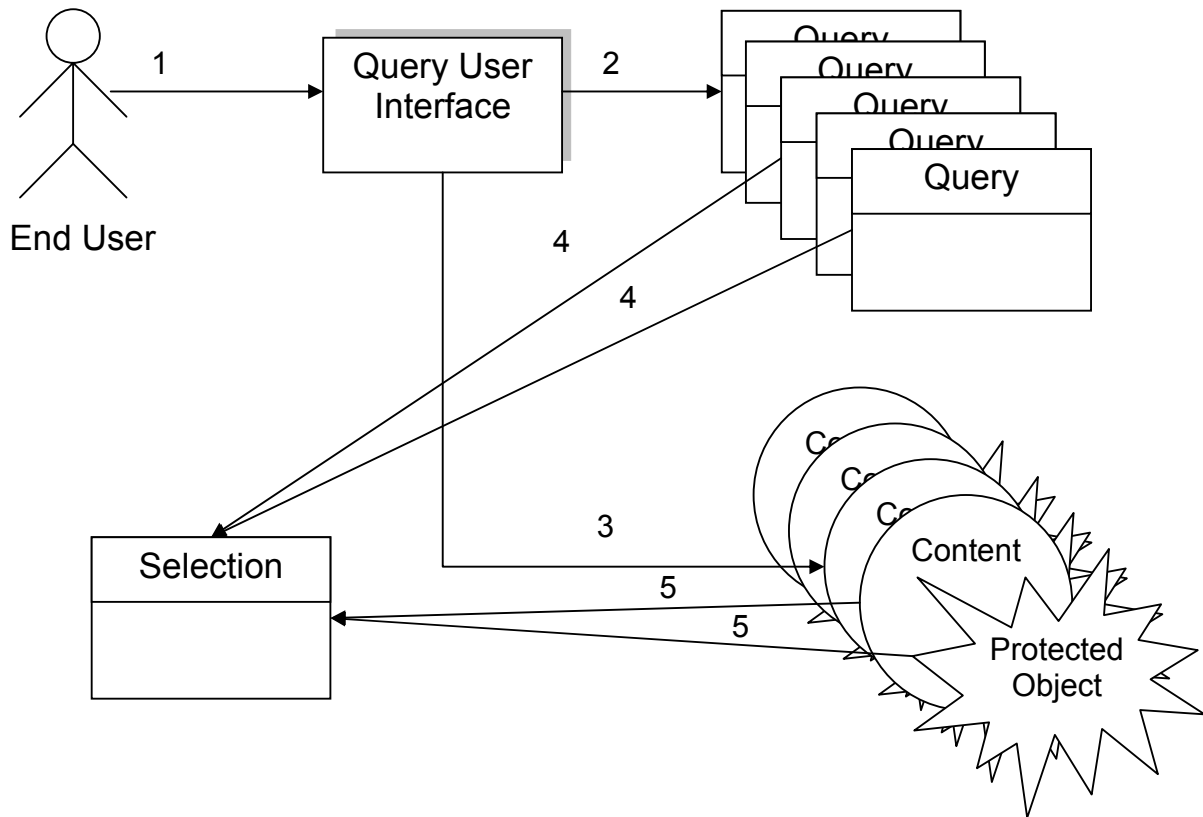
1. The End-User creates a query on the user interface
- 2., 3. The Query is send to the AXQS by the user interface
4. The AXQS returns a Selection (see Scenario 6.1)
5. The selection is processed by an AXMEDIS tool
6. The End-User selects one of more objects in the selection
- 7., 8. The AXMEDIS Tool issues a request for the set of objects to the AXDBM
- 9., 10. The AXDB returns the set of requested objects
11. The objects are ready to be used in the AXMEDIS Tool

4.2.2 SCENARIO 6.1: Making query on the several direction. Technical Querying, Collecting and integrating results.



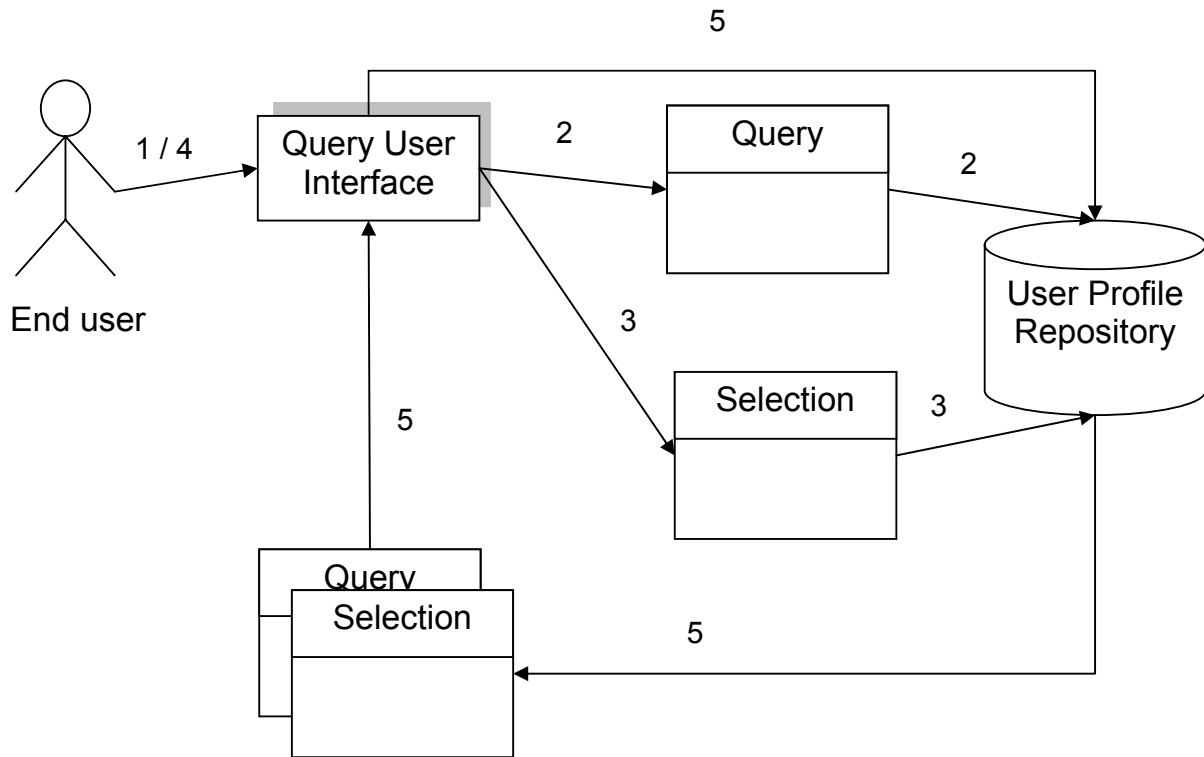
1. The End user, using the AXQS User Interface, composes a Query on aspects of interest (technical, DRM or feature related). Furthermore, the Actor chooses “where” to search for available AXMEDIS objects: within local AXMEDIS Database (and within AXMEDIS objects contained within the local AXDB), on AXEPTool network or among those contents which have to be collected, by the Collector Engine, and have not yet.
2. The End User submits the queries previously composed
3. AXQS submits the Actor’s query to each of the chosen search “places” by using the corresponding specific interface: (i) Collector Engine Query Support Interface, (ii) AXEPTool Query Support Interface and (iii) AXMEDIS Database Manager
4. Each query interface (see step 3) looks for the required features in the corresponding domain
5. AXQS collects all the responses from the query interfaces
6. AXQS merges the results all together and return the complete list to the AXQS User Interface
7. AXQS User Interface shows the result to the Actor in an adequate manner, i.e. in such a way that the Actor can understand: (i) from which source an object come (ii) which are the restriction on the object (iii) etc...

4.2.3 SCENARIO 6.2: Producing a selection.



- 1., 2. End user can access through the AXQS User Interface to a collection of queries (previously issued)
3. End User can access also to a collection of Object (reference to object in the case of AXQS user interface)
- 4., 5. The set of an arbitrary number (0 or more) of queries and of an arbitrary number of reference to objects (0 or more) are put together to form a Selection

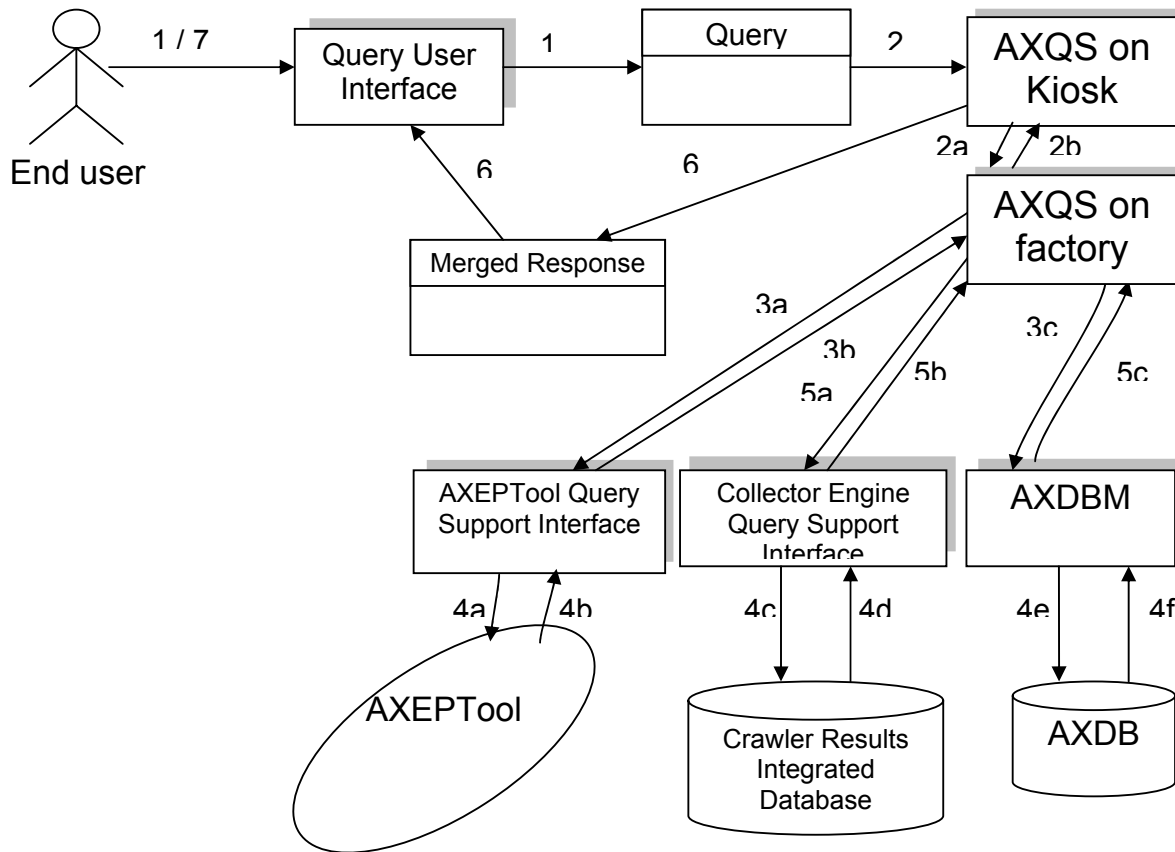
4.2.4 SCENARIO 6.5: Store a query/Selection and make it active later



1. End User uses AXQS user interface to interact with his/her personal profile
2. End User can save in his/her user profile some Queries
3. End User can save in his/her user profile some Selections
4. End User can recall a Selection/Query from user profile
5. Query or selection is returned to the User Interface for activation/modification

4.2.5 SCENARIO 6.6: Make a query on a Kiosk (EXITECH, DSI, ILABS)

This scenario allows to query the local Kiosk QS or to forward the query to the remote QS that is in the factory that in turn can forward the query to P2P, Factory AXDB and Factory crawling mechanism.



1. The End user, using the AXQS User Interface, composes a Query on aspects of interest (technical, DRM or feature related). Furthermore, the Actor chooses “where” to search for available AXMEDIS objects: within local AXMEDIS Database (and within AXMEDIS objects contained within the local AXDB), on AXEPTool network or among those contents which have to be collected, by the Collector Engine, and have not yet.
2. The End User submits the queries previously composed on the Kiosk
- 2a,b The query support of the Kiosk can forward the query to the Factory QS that in turn forwards it to the other information sources.
3. AXQS submits the Actor’s query to each of the chosen search “places” by using the corresponding specific interface: (i) Collector Engine Query Support Interface, (ii) AXEPTool Query Support Interface and (iii) AXMEDIS Database Manager
4. Each query interface (see step 3) looks for the required features in the corresponding domain
5. AXQS collects all the responses from the query interfaces
6. AXQS merges the results all together and return the complete list to the AXQS User Interface
7. AXQS User Interface shows the result to the Actor in an adequate manner, i.e. in such a way that the Actor can understand: (i) from which source an object come (ii) which are the restriction on the object (iii) etc...

4.3 Query Format and language (EXITECH)

The query support interface with the rest of the AXMEDIS component is realized by the means of a Webservice interface. This fact implies that the messages (that are the queries and the results of the queries) that are exchanged among the different AXMEDIS component must be in XML format.

To this end, in this section a basic schema for defining how to express a query and a query result is defined. These schemas will be the foundation also for the Webservice WSDL.

4.3.1 General consideration on queries and on object search

In principle the query a user performs can be summarized in a research for one or more object that have some characteristics. The taxonomy of a query can be the one reported in the following:

- **Known Knowns (KK-Requests)**

Requesting a known object whose AXOID is precisely, unmistakeably and uniquely known to both the requester and service provider. So that you request an object that you know that exists and for which you provide a known identifier

- **Known Semi-Unknowns (KSU-Requests)**

Requesting a known object whose object_ID is uncertain or partially unknown at least as far as the requester is concerned. This case can exist when you partially remember the AXOID and you are looking for all AXOID similar to that. Similarity can be applied also to other fields.

- **Known Unknowns (KU-Requests)**

Requesting a known object whose object_ID is completely unknown at least as far as the requester is concerned. This is the classical query for which you do not know if the object exists but you can classify it on the basis of its parameters.

- **Unknown Unknowns (UU-Requests)**

Requesting any objects hitherto unknowable by the requester (and possibly also the service provider) which may satisfy some requester-stipulated interestingness criteria. These kind of queries are the most difficult to be solved since the concept of an interestingness function is very difficult to be mapped in standard query. In any case if the interestingness function can operate on AXINFO of the object and can be provided by a remote server or locally evaluated, it can be addressed in the next revision of the specification after the first 18 months of the project.

The UU-Request could be further specified to be at any one of two levels:

- **UU-Request-Level-1 Object Discovery Requests: Request for *Chance Discoveries* in any Pools**

This is the type of request which submits an interestingness or other evaluation function parameters with a request for an object fishing exercise (exploration) from unspecified (don't care), pools or domains of exploration.

- **UU-Request-Level-2 Object Discovery Requests: Request for *Discovering Chances*:**

This is the type of request that may or may not be expecting any specific digital asset(s) as a found object(s) in response. The UU-Request-Level-2 Object Discover Request will essentially be requesting the identity of any Pools/Repositories and associated Query_ID which resulted in the return of certain objects with a particular property or level of interestingness. Thus this will return Pool_IDs, Request_IDs with associated ranking/probability of finding certain category of objects satisfying certain interestingness or other evaluation function (optionally it might be expected to return a sample of the respective objects).

4.3.2 Schema for an AXMEDIS query

In this section the general schema for representing an AXMEDIS query is reported and commented.

The schema of the query is mainly composed by three parts. In the first part the location on which the query has to be distributed is reported, in the second part a list of fields that the query should give as the result is contained and in the third and final part, the conditions under which a query has to be executed are imposed.

The three different parts are separately commented after the textual and graphic representation of the Query schema in sections

The proposed schema is reported below, in text format and in a more fashionable graphic format that can be useful to have a general overview.

According to the last revision of the query model and of the way in which query can be submitted by the user, it can be noted that user performs queries on Descriptors and PAR/InternalPar for obtaining a list of results that satisfy the criteria. The DRM part is managed in a second step of the query, where from the licence DB, all the user that have a licence to sell a licence to the user according to the PAR limitation imposed, are automatically extracted, so that the user the performed the query can select the most appropriate to its needs. For these reasons the limitation on DRM that were present in the previous query model have been removed, moving this functionality to the automatic generation of a list of possible licensors.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="AXInfofield" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="query">
    <xs:annotation>
      <xs:documentation>root element for an AXMEDIS query, supported by AXQS</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="source">
          <xs:complexType>
            <xs:sequence maxOccurs="4">
              <xs:element name="location">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration
value="CRAWLER"/>
                    <xs:enumeration
value="AXEPTOOL"/>
                    <xs:enumeration value="AXDB"/>
                    <xs:enumeration value="QS"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="locationURI" type="xs:anyURI"
minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element ref="result" minOccurs="0"/>
        <xs:sequence>
          <xs:element name="PARquery" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="querycondition"/>
              </xs:sequence>
              <xs:attribute name="InternalPAR" type="xs:boolean"
use="optional"/>
            </xs:complexType>
          </xs:element>
          <xs:element name="AXinfoQuery" minOccurs="0">
            <xs:complexType>
```



```

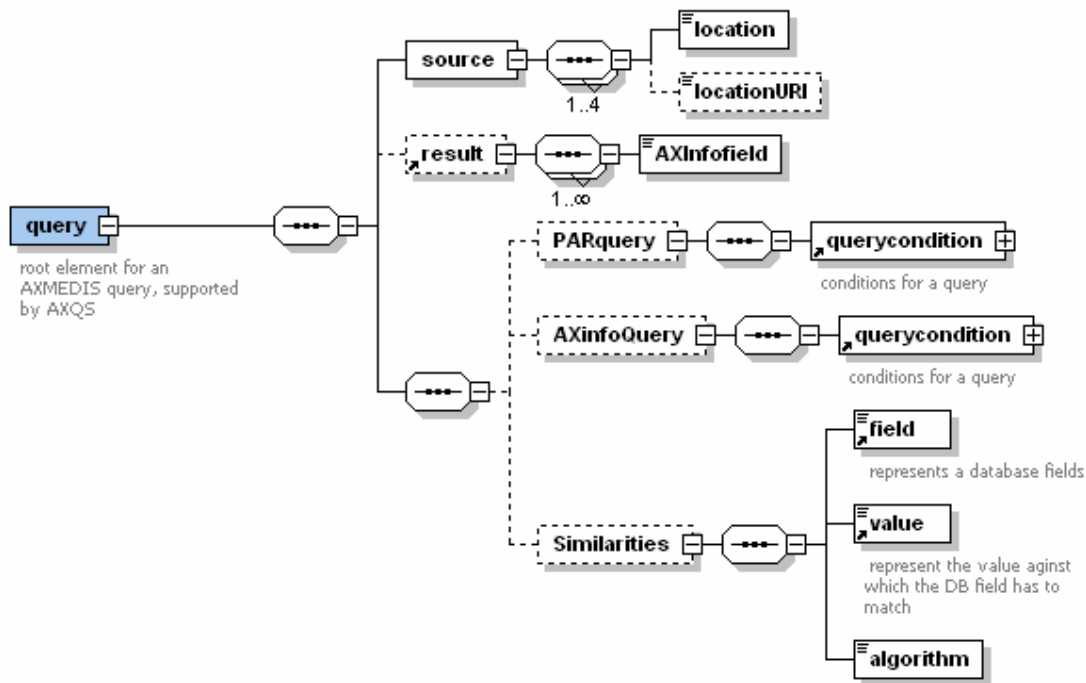
<xs:sequence>
  <xs:element ref="querycondition"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Similarities" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="field"/>
      <xs:element ref="value"/>
      <xs:element name="algorithm">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration
value="REGULAREXPRESSION"/>
            <xs:enumeration
value=""/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:attribute name="name" type="xs:string" use="optional" default="QUERY"/>
</xs:complexType>
</xs:element>
<xs:element name="field" type="xs:string">
  <xs:annotation>
    <xs:documentation> represents a database fields</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="value" type="xs:anySimpleType">
  <xs:annotation>
    <xs:documentation> represent the value against which the DB field has to match</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="test">
  <xs:annotation>
    <xs:documentation> test element of a query: it is in the form field OP value</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="field"/>
      <xs:element name="operator">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="GT"/>
            <xs:enumeration value="LT"/>
            <xs:enumeration value="EQ"/>
            <xs:enumeration value="GE"/>
            <xs:enumeration value="LE"/>
            <xs:enumeration value="NE"/>
            <xs:enumeration value="STARTWITH"/>
            <xs:enumeration value="ENDWITH"/>
            <xs:enumeration value="CONTAINS"/>
            <xs:enumeration value=""/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element ref="value"/>
    </xs:sequence>
    <xs:attribute name="NOT" type="xs:boolean" use="optional" default="false"/>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="nesting">
  <xs:annotation>
    <xs:documentation>grouping by parentheses</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:sequence>
        <xs:element ref="test"/>
        <xs:choice minOccurs="0">
          <xs:element ref="and"/>
          <xs:element ref="or"/>
        </xs:choice>
      </xs:sequence>
      <xs:sequence>
        <xs:element ref="nesting"/>
        <xs:choice minOccurs="0">
          <xs:element ref="and"/>
          <xs:element ref="or"/>
        </xs:choice>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="querycondition">
  <xs:annotation>
    <xs:documentation>conditions for a query</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nesting"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="and" type="xs:token">
  <xs:annotation>
    <xs:documentation>and operator</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="or" type="xs:token">
  <xs:annotation>
    <xs:documentation>or operator</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:schema>

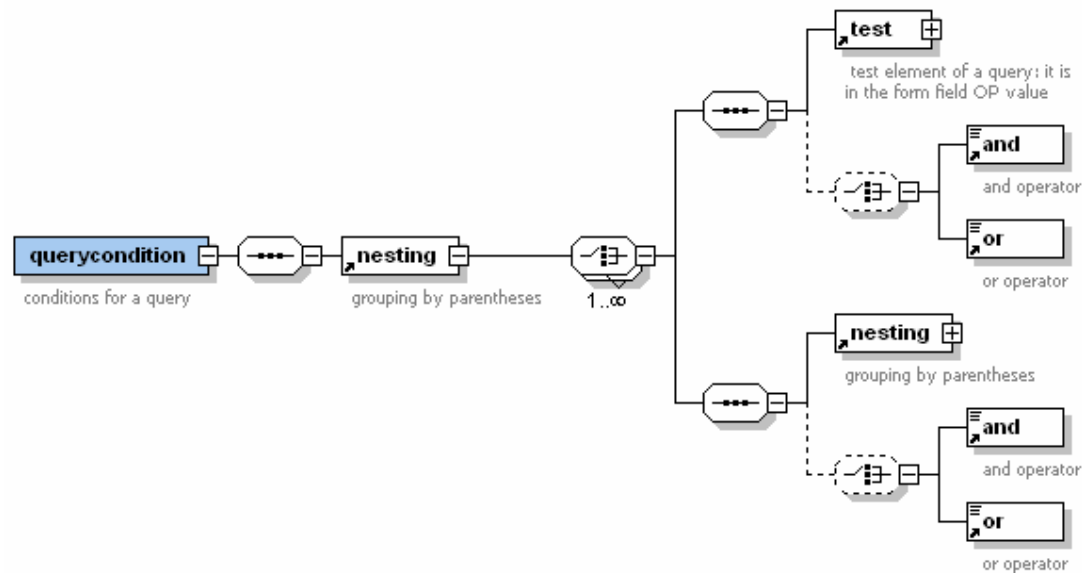
```



In this first diagram we suppose that a query can be performed on different sources, returning different AXINFO fields and that the query conditions can be expressed on three different information sources:

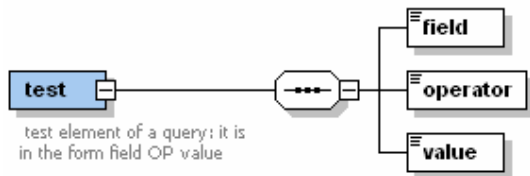
- PARquery, that are the info related to Possible Available Rights
- DRMMquery, that are the information about licensing
- AXinfoQuery, that are the info related to the metadata of the AXMEDIS object.

For each one of these groups a query condition applies and the groups are merged on an AND basis. Querycondition is reported in the following schema:



Querycondition is then a set of nesting elements that represents basically the parentheses levels.

A nesting can be nested in another nesting element of it can be a part of the real query, in the case it is part of a real query it is a combination of test items that are reported in the following:



Test is the basic element where a fields can be compared to a value by the means of an operator that will be discussed in more details in the following.

Test element has also an attribute that allows to specify if it must be negated or not.

4.3.2.1 Query sources

The AXMEDIS query can be distributed on four different locations that are: CRAWLER, AXDB, AXEPTOOL, and other remote query supports. This is the way the source tag has between 1 and 4 locations that can have a value in the enumeration CRAWLER, AXEPTOOL, AXDB, QS and have also an optional URL for covering the need of a remote Query Support that is needed by the Kiosk, when the kiosk will look in the kiosk factory if an object is preseeent or not. This new addition allows also to create a network of query supports that are cooperatively in an hierarchy tree.

4.3.2.2 Query result

The second part of the schema is used to describe the fields that the query should return together with the AXOID (or the temporary AXOID in the case of CRAWLER) that is considered a mandatory field. If the result tag is not present, then only the list of AXOID will be returned; otherwise at least one field must be specified. After the field list an optional array of sorting parameter can be inserter; each sortby tag has an optional attribute (not shown in the graphic version) that by default assumes the asc=true value, that means ascending sorting. If imposed to false, it means that sorting is descending.

The field tag will be limited to the possible list of fields, once the database schema will be defined and a selecting among the possible fields to be queried will be defined.

4.3.2.3 Query conditions

The most relevant part of a query is the section related to query conditions that is of course optional, but if present must have the structure reported in the previous XML schema. All the field tag must contains the database field that must match, all value field must contain the value against which the field is checked and operator is one of the following:

- GT: greater than
- LT: less than
- GE: greater equal
- LE: less equal
- EQ: Equal
- NE: Not equal
- STARTWITH: field must start with the value
- ENDWITH: field must end with the value
- CONTAINS: field must contain in any position the value

The field tag will be limited to the possible list of fields, once the database schema will be defined and a selecting among the possible fields to be queried will be defined.

4.3.2.4 Similarities

Similarities are a particular way of making a specific similarity request on the set ob objects selected by the previous queryconditions. By now only one similarity field as been inserted in order to reduce the

complexity of the query. Similarities can be addressed by regular expressions (i.e. for searching a partially known AXOID) or by more advanced algorithms to be defined on the basis of the needs of the project.

4.3.2.5 XML Query example file (EXITECH, ILABS, COMVERSE)

In this example the query proposed as a sample by ILABS that is:

SELECT * FROM * WHERE ((AUTHOR .EQ. "BOTTICELLI" .AND. (MEDIA .EQ. "VIDEO" .OR. MEDIA .EQ. "AUDIO" .OR. MEDIA .EQ. "TEXT")) .AND. (IPR .EQ. "FREE" .AND. COST .LT. "10.00"))

Apart from the fields for which we suppose to have only the AXOID and

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="QUERY-v1-6.xsd">
  <source>
    <location>CRAWLER</location>
    <location>AXEPTOOL</location>
    <location>AXDB</location>
  </source>
  <AXinfoQuery>
    <querycondition>
      <nesting>
        <nesting>
          <nesting>
            <test>
              <field>AUTHOR</field>
              <operator>EQ</operator>
              <value>BOTTICELLI</value>
            </test>
          </nesting>
          <and/>
          <nesting>
            <test>
              <field>MEDIA</field>
              <operator>EQ</operator>
              <value>VIDEO</value>
            </test>
            <or/>
            <test>
              <field>MEDIA</field>
              <operator>EQ</operator>
              <value>AUDIO</value>
            </test>
            <or/>
            <test>
              <field>MEDIA</field>
              <operator>EQ</operator>
              <value>TEXT</value>
            </test>
          </nesting>
        </nesting>
      </nesting>
      <and/>
      <nesting>
        <test>
          <field>IPR</field>
          <operator>EQ</operator>
          <value>FREE</value>
        </test>
        <and/>
        <test>
          <field>COST</field>
          <operator>LT</operator>
          <value>10.00</value>
        </test>
      </nesting>
    </querycondition>
  </AXinfoQuery>
</query>
```

Examples can be as complex as we want since the possibility of combining with AND and OR the number of desired nesting levels.

Another sample of a simpler query is reported in the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation=" QUERY-v1-6.xsd">
  <source>
    <location>AXDB</location>
  </source>
  <AXinfoQuery>
    <querycondition>
      <nesting>
        <test>
          <field>PIPP0</field>
          <operator>EQ</operator>
          <value>3</value>
        </test>
        <and/>
        <nesting>
          <test>
            <field>PAPERINO</field>
            <operator>EQ</operator>
            <value>1</value>
          </test>
          <or/>
          <test>
            <field>PLUTO</field>
            <operator>GT</operator>
            <value>1</value>
          </test>
        </nesting>
      </nesting>
    </querycondition>
  </AXinfoQuery>
</query>
```

Other query samples in an SQL like dialect, have been collected and are reported below for the sake of completeness.

ILABS samples:

```
SELECT * FROM * WHERE (AUTHOR .EQ. "BOTTICELLI" .AND. SUBJECT .EQ. *
.AND. TYPE .EQ. "IMAGE" .AND. COST .LT. "350.00")
```

```
SELECT * FROM * WHERE (AUTHOR .EQ. * .AND. SUBJECT .EQ. * .AND. TYPE .EQ.
"IMAGE" .AND. (IPR .EQ. "FREE" .OR. COST .LT. "50.00"))
```

```
SELECT * FROM * WHERE ((AUTHOR .EQ. "BOTTICELLI" .AND. TYPE .EQ. "TEXT")
.AND. (IPR .EQ. "FREE" .OR. COST .LT. "50.00"))
```

```
SELECT * FROM * WHERE ( AUTHOR .EQ. "BOTTICELLI" .AND. ((SUBJECT .EQ.
"ART" .OR. SUBJECT .EQ. "PAINTING") .AND. TITLE .EQ. "BIRTH OF VENUS") .AND.
((TYPE .EQ. "IMAGE" .AND. (FORMAT .EQ. "JPG" .OR. FORMAT .EQ. "TIF")) .OR.
(TYPE .EQ. "MOVINGIMAGE" .AND.(FORMAT .EQ. "MOV" .OR. FORMAT .EQ. "MPG"))))
.AND. LANGUAGE .EQ. "EN" .AND. (RIGHTS .EQ. "FREE" .AND. ACCESSRIGHTS .NE.
"HAS RESTRICTIONS" .AND. LICENSE .EQ. "ANY") .AND. COST .LT. "350.00")
```

COMVERSE samples:

1. Comverse distribution model:

- a. The existing and AXMEDIS enabled systems work by this model, in which Comverse buys content once (B2B), and distributes the content to the end consumers (B2C) in great numbers. The B2C distribution is not reported to any of the AXMEDIS platform entities.
- b. The AXMEDIS enabled distribution system will require the object's License/Available rights to match the following Criteria, which will be used with every Query:
 - i. The distributor must pay only once for purchasing an AXMEDIS object.
 - ii. The distributor is not required to pay for downloading a purchased object. The distributor must be able to download the purchased object more than one time without paying additional fee.
 - iii. The distributor may distribute the content of the purchased object to end consumers.
 - iv. The distributor is not required to use the AXMEDIS platform to report any b2c actions that are applied on the object (e.g. when the consumer buys content via the Comverse distribution system the AXMEDIS entities are not aware of the purchase action or that the content was sent to the consumer).

2. Transcoding Selection:

- a. Purpose:
 - i. Get a list of all/selected AXMEDIS objects
 1. that match criteria where objects:
 - a. are available for transcoding within a date range
 - b. can be transcoded to media type and formats that are supported by the Comverse distribution system and the target devices (i.e. mobile phones)
 - c. the content generated by the transcoding process can be packed to objects with license that agrees with the [Comverse distribution model](#)
 - d. the objects created by the transcoding process are available for distribution within a date range
 - e. are directly associated with the media content; e.g. Song Collection object, Picture object (No complex objects like Encyclopedia Britannica or the BBC Archive)
 2. Ordered by priority; e.g. most recent publications, AND/OR distribution availability time, AND/OR fastest to transcode (by codec, encryption, size, etc)).
 - ii. Download the highest priority objects to be transcoded while not exceeding the storage capacity allocated for downloaded objects.
 - iii. Removal of the downloaded objects and rejects that no longer match the criteria from the storage.
- b. Criteria:
 - i. Technical Criteria:
 1. The content of the objects must be of the types (e.g. Audio, Graphics, Text) and formats (e.g. WMV, MP3, JPG, length, dimensions, bitrate, color depth) that are desired by the distributor and can be transcoded by the Transcoding Server
 - a. Audio attribute criteria:
 - i. Between Min and Max length (audio, text/doc?)
 - ii. Between Min and Max bitrate
 - iii. Between Min and Max frequency
 - iv. Mono, Stereo, Joint Stereo, surround
 - v. Other (?)
 - b. Midi attribute criteria:

- i. Other (?)
 - c. Raster graphics (JPEG, GIF, TIF, etc) attribute criteria:
 - i. Between Min and Max of Mega-pixel or X and Y dimensions.
 - ii. Between Min and Max color depth.
 - iii. Other (?)
 - d. Vector graphics (Visio, WMF, CDR, etc) attribute criteria (?):
 - i. Other (?)
- ii. Legal Criteria (License and terms of usage):
 - 1. allow the Transcoding Action to the desired formats
 - 2. The object is available for transcoding within a date range
 - 3. a fee per-transcode is in range of \$x.cc to \$x.cc (? I'm not sure this can be accomplished because the issue of pricing-in-license is not clear)
 - 4. the product of transcoding the object can be generated a license that agrees with the [Comverse distribution model](#)
 - 5. the product of transcoding the object is available for distribution within a date range
- c. EXAMPLE 1:
 - i. Retrieve a list of all AXMEDIS objects that:
 - 1. their content is in one of the following formats:
 - a. WAV, WMA audio, with the following attributes:
 - i. Bitrate is between 64 to 256
 - ii. Length is no shorter than 2 minutes and no longer than 4 minutes
 - b. MP3 audio, with the following attributes:
 - i. Bitrate is between 128 to 384
 - ii. Length is no shorter than 2 minutes and no longer than 4 minutes
 - c. JPG, GIF graphics, with the following attributes:
 - i. Color depth is greater than 4
 - ii. Mega-pixel size is equal to or smaller than 2
 - iii. Width is greater than 45 pixels and not larger than 1200 pixels.
 - 2. their license allows
 - a. transcoding to the following formats:
 - i. Audio to format WMA-2min-freq48, AND/OR Ring-tone,
 - ii. Graphic to formats JPG-48x48x24bit, AND/OR GIF-89a-4x4x256.
 - b. Transcoding between Jan 1, 2005 – Dec 31, 2005.
 - c. packing and distribution of the transcoding product objects to the following devices:
 - i. All Nokia mobile phones
 - ii. Motorola-T28oM
 - iii. Siemens S40 series
 - d. Distribution starting Jan 1, 2005 (until forever)
 - e. Available rights that agree with the [Comverse distribution model](#).

3. Distribution Criteria:

- a. Purpose:
 - i. Get a list of all/selected AXMEDIS objects
 - 1. that match criteria where objects:

- a. are available for distribution within a date range
 - b. contain or reference content type and formats that are supported by the Comverse distribution system and the target devices (i.e. mobile phones)
 - c. their license agrees with the Comverse distribution model (buy once, bulk distribution to mobile devices, does not require reporting of distribution via AXMEDIS platform)
 - d. are directly associated with the media content; e.g. Song Collection object, Picture object - No complex objects like Encyclopedia Britannica or the BBC Archive.
2. Ordered by priority; e.g. most recent publications, AND/OR distribution availability date, AND/OR most popular publication (to what degree is popularity/statistics supported by the DB/AXMEDIS?)
 - ii. Download the highest priority objects to be distributed via the Comverse distribution system.
 - iii. Removal of the downloaded objects that no longer match the criteria from the storage.
- b. Criteria:
 - i. Technical Criteria:
 1. The content of the objects must be of the types (e.g. Audio, Graphics, Text) and formats (e.g. WMV, MP3, JPG, length, dimensions, bitrate, color depth) that are desired by the distributor
 - a. Audio attribute criteria:
 - i. Between Min and Max length (audio, text/doc?)
 - ii. Between Min and Max bitrate
 - iii. Between Min and Max frequency
 - iv. Mono, Stereo, Joint Stereo, surround
 - v. (?)
 - b. Midi attribute criteria:
 - i. (?)
 - c. Raster graphics (JPEG, GIF, TIF, etc) attribute criteria:
 - i. Between Min and Max of Mega-pixel or X and Y dimensions.
 - ii. Between Min and Max color depth.
 - iii. (?)
 - d. Vector graphics (Visio, WMF, CDR, etc) attribute criteria (?):
 - i. (?)
 - ii. Legal Criteria (License and terms of usage):
 1. See The object is available for distribution within a date range
 2. The object's license agrees with the [Comverse distribution model](#).
- c. EXAMPLE 1:
 - i. Retrieve a list of all AXMEDIS objects that:
 1. their content is in one of the following formats:
 - a. WAV, WMA audio, with the following attributes:
 - i. Bitrate is between 64 to 256
 - ii. Length is no shorter than 2 minutes and no longer than 4 minutes
 - b. Midi audio, with the following attributes:
 - i. Level is between 4 to 24
 - ii. Length is no shorter than 2 minutes
 - c. JPG, GIF graphics, with the following attributes:
 - i. Color depth is greater than 4
 - ii. 15KB in size or smaller
 - iii. 45 pixels in width and 45 pixels in height.

2. their license allows
 - a. distribution to the following devices:
 - i. All Nokia mobile phones
 - ii. Motorola-T280M
 - iii. Siemens S40 series
 - b. Distribution between starting Jan 1, 2005 (until forever)
 - c. Purchase for the Distribution with the [Comverse distribution model](#).

4.3.3 Schema for an AXMEDIS query result

In this section the schema for the results that are returned back after a query is reported and discussed.

It is important to track if the result is coming from a source (i.e. CRAWLER, that means that some automatic operation for the creation of an AXMEDIS object starting from the content in the CMS have to be performed before using the object) or from a remote channel such as AXEPTOOL that means that the content is not immediately available, but has to be downloaded from a remote peer.

The information related to the source of information must come together with the fields requested by the user in the query.

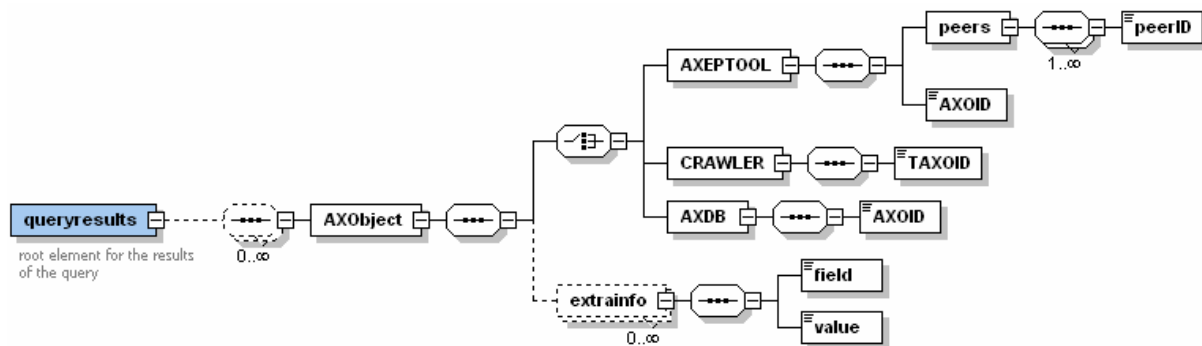
The proposed schema is reported below, in text format and in a more fashionable graphic format that can be useful to have a general overview.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="queryresults">
    <xs:annotation>
      <xs:documentation>root element for the results of the query</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="AXObject">
          <xs:complexType>
            <xs:sequence>
              <xs:choice>
                <xs:element name="AXEPTOOL">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element
name="peers">
      <xs:complexType>
        <xs:sequence maxOccurs="unbounded">
          <xs:element name="peerID" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="CRAWLER">
  <xs:complexType>
    <xs:sequence>
      <xs:element
name="TAXOID" type="xs:string"/>
    </xs:sequence>
```

```

name="AXOID" type="xs:string"/>
maxOccurs="unbounded">
type="xs:string"/>
type="xs:string"/>
</xs:schema>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:choice>
<xs:element name="extrainfo" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element name="field">
<xs:element name="value">
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```



The schema of the result is quite simple since it is an optional list of AXMEDIS objects (AXObject) and for each opbject the information source is reported (AXEPTOOL, CRAWLER, AXDB) together with the informatins that are relevant for the source, the list of peers that have the object together with the AXIOD for AXEPTOOL, the AXOID for AXDB and the temporary AXOID if the object is recovered by the crawler. For each AXObject a set of extainfo can be optionally reported. Each extrainfo tag is comprised of a couple of field and value. In this case also, when the AXDB schema will be stable we will be able to limit the valid content for field tag.

4.3.3.1 XML results example file

In this section are collected sample XML file compliant to the proposed schema that can be useful to understand the practical structure of the results of a query.

```
<?xml version="1.0" encoding="UTF-8"?>
  <AXObject>
    <AXEPTOOL>
      <peers>
        <peerID>12234</peerID>
```

```

        <peerID>56454</peerID>
        <peerID>12784</peerID>
    </peers>
    <AXOID>12248766-gftr</AXOID>
</AXEPTOOL>
<extrainfo>
    <field>Author</field>
    <value>Ramazzotti</value>
</extrainfo>
<extrainfo>
    <field>YearOfPublication</field>
    <value>1996</value>
</extrainfo>
<extrainfo>
    <field>Duration</field>
    <value>315</value>
</extrainfo>
</AXObject>
<AXObject>
    <CRAWLER>
        <TAXOID>5383999-temp</TAXOID>
    </CRAWLER>
    <extrainfo>
        <field>Author</field>
        <value>Ramazzotti</value>
    </extrainfo>
    <extrainfo>
        <field>YearOfPublication</field>
        <value>1997</value>
    </extrainfo>
    <extrainfo>
        <field>Duration</field>
        <value>417</value>
    </extrainfo>
</AXObject>
<AXObject>
    <AXDB>
        <AXOID>35423912843.fkhwygywe</AXOID>
    </AXDB>
    <extrainfo>
        <field>Author</field>
        <value>Ramazzotti</value>
    </extrainfo>
    <extrainfo>
        <field>YearOfPublication</field>
        <value>1995</value>
    </extrainfo>
    <extrainfo>
        <field>Duration</field>
        <value>234</value>
    </extrainfo>
</AXObject>
</queryresults>

```

4.4 Selection Format and storage (EXITECH)

Selections are collections of symbolic queries and AXMEDIS Objects IDs. Selections are then a set comprised of symbolic queries not expanded or actualized such as S1= {Q1, Q2} or a selection of AXMEDIS Object IDs with some symbolic queries such as S2={Q3, AXOB1, AXOB444, AXOB3412}. Selection can be expanded or actualized so that S1 at time t1 can be S1@t1= {AXO1-1, AXO1-2, AXO1-3, AXO2-1, AXO2-2, AXO2-3, AXO2-4, AXO2-5, AXO2-6, AXO2-7, AXO2-8, AXO2-9}, while the same

query at time t2 can be S1@t2= { AXO1-3, AXO2-1, AXO2-2, AXO2-3, AXO2-4, AXO2-5, AXO2-6, AXO2-8, AXO2-9, AXO3-1, AXO3-2}

A selection can be formalized in XML starting from what has been stated for query considering that in a selection only AXMEDIS objects and symbolic queries can be found. Considering the selection in a deeper detail, it is evident that only objects that are in AXDB can be part of the selection together with symbolic queries. This means that object extracted from the crawling system need to be transformed in AXMEDIS object to be inserted in a selection with their AXOID, and objects got from AXEPTOOL need to be imported in the AXDB before being used put in a selection.

The formal model for a selection can be summarized as a collection of queries and of AXMEDIS objects that can be recovered in the AXDB, and therefore a collection of AXOIDs.

4.4.1 XML Selection Schema

A selection is a collection of zero or more query with zero or more AXOID. It is necessary that at least an AXOID or a query is inside the selection.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="nesting">
    <xs:annotation>
      <xs:documentation>grouping by parentheses</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:sequence>
          <xs:element ref="test"/>
          <xs:choice minOccurs="0">
            <xs:element ref="and"/>
            <xs:element ref="or"/>
          </xs:choice>
        </xs:sequence>
        <xs:sequence>
          <xs:element ref="nesting"/>
          <xs:choice minOccurs="0">
            <xs:element ref="and"/>
            <xs:element ref="or"/>
          </xs:choice>
        </xs:sequence>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="AXInfofield" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="query">
    <xs:annotation>
      <xs:documentation>root element for an AXMEDIS query, supported by AXQS</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="source">
          <xs:complexType>
            <xs:sequence maxOccurs="4">
              <xs:element name="location">
                <xs:simpleType>
```

```

value="CRAWLER"/>
value="AXEPTOOL"/>
value="REGULAREXPRESSSION"/>
value=""/>
value="test"/>
<xs:restriction base="xs:string">
  <xs:enumeration value="QS"/>
  <xs:enumeration
    <xs:enumeration value="AXDB"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="locationURI" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element ref="result" minOccurs="0"/>
<xs:sequence>
  <xs:element name="PARquery" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="querycondition"/>
      </xs:sequence>
      <xs:attribute name="InternalPAR" type="xs:boolean"
        use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="AXinfoQuery" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="querycondition"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Similarities" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="field"/>
        <xs:element ref="value"/>
        <xs:element name="algorithm">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration
                </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="optional" default="QUERY"/>
</xs:complexType>
</xs:element>
<xs:element name="field" type="xs:string">
  <xs:annotation>
    <xs:documentation> represents a database fields</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="value" type="xs:anySimpleType">
  <xs:annotation>
    <xs:documentation> represent the value against which the DB field has to match</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="test">

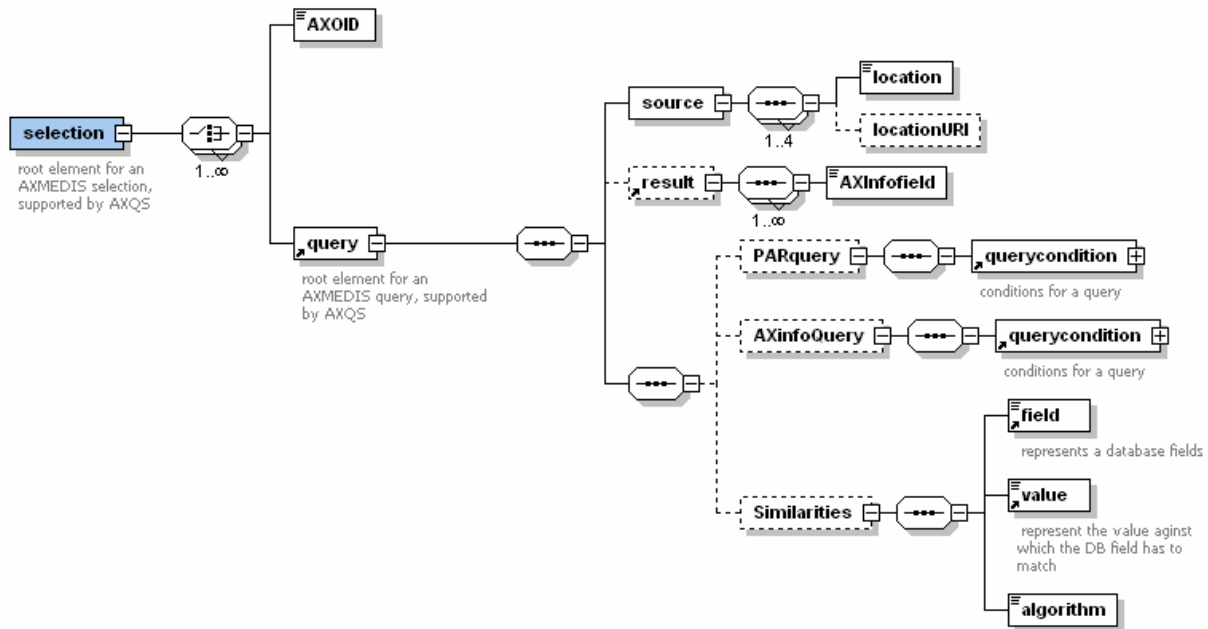
```

```

<xs:annotation>
  <xs:documentation> test element of a query: it is in the form field OP value</xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element ref="field"/>
    <xs:element name="operator">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="GT"/>
          <xs:enumeration value="LT"/>
          <xs:enumeration value="EQ"/>
          <xs:enumeration value="GE"/>
          <xs:enumeration value="LE"/>
          <xs:enumeration value="NE"/>
          <xs:enumeration value="STARTWITH"/>
          <xs:enumeration value="ENDWITH"/>
          <xs:enumeration value="CONTAINS"/>
          <xs:enumeration value=""/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element ref="value"/>
  </xs:sequence>
  <xs:attribute name="NOT" type="xs:boolean" use="optional" default="false"/>
</xs:complexType>
</xs:element>
<xs:element name="querycondition">
  <xs:annotation>
    <xs:documentation>conditions for a query</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nesting"/>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="and"/>
          <xs:element ref="or"/>
        </xs:choice>
        <xs:element ref="nesting"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="and" type="xs:token">
  <xs:annotation>
    <xs:documentation> and operator</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="or" type="xs:token">
  <xs:annotation>
    <xs:documentation> or operator</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="selection">
  <xs:annotation>
    <xs:documentation>root element for an AXMEDIS selection, supported by AXQS</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element name="AXOID" type="xs:string"/>
      <xs:element ref="query"/>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="timestamp" type="xs:dateTime" use="required"/>
  </xs:complexType>

```

```
</xs:element>
</xs:schema>
```



4.4.2 XML Selection Samples

```
<?xml version="1.0" encoding="UTF-8"?>
<selection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents\Docs\Exitech\AXMEDIS\Deliverables\Specific
ation\AXMEDIS-DE3-1-2-AXFW\Current-PART-E\Selection-v1-1.xsd" name="TEST"
timestamp="2005-01-20T18:20:46.275+01:00">
  <AXOID>3y7932469236</AXOID>
  <AXOID>824375832741723</AXOID>
  <query>
    <source>
      <location>CRAWLER</location>
    </source>
    <AXinfoQuery>
      <querycondition>
        <nesting>
          <test>
            <field>AUTHOR</field>
            <operator>STARTWITH</operator>
            <value>MOZ</value>
          </test>
        </nesting>
      </querycondition>
    </AXinfoQuery>
  </query>
</selection>
```



```

</AXinfoQuery>
</query>
</selection>

```

4.4.3 XML Selection Storage

Selection are created by the user of the system or of the tool and therefore the logical place in which they can be stored is in the user profile. For the specification see section 4.6

4.5 Query User Interface (EXITECH, CRS4, DSI)

Query User Interface will not be specified in this document since it is in charge to CRS4. It is present in AXFW Part F related to AXEPTool and P2P.

4.6 Query and Selection Archive for Each User (EXITECH, DSI, HP, XIM, IRC)

Module Profile		
Query and Selection Archive for Each User		
Executable or Library(Support)	Executable	
Single Thread or Multithread	Multithread	
Language of Development	JAVA	
Responsible Name	TBD	
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.)
All tools and engines that need to get aselection or to store an user selection		
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.
None	WebService	
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

This module take care of storing the query and the selection of each user in order to have a personal archive for each user that want to store his/her queries and selection for future reuse. The query or the selection must

have some characteristics in order to be easily identified by the user. It can be supposed to have the following data stored with each query or selection:

- User that has created the query or selection
- ID of the query or selection (Mandatory);
- Name of the query or selection assigned by the user (Mandatory): this is for allowing the user to assign a symbolic name to the query for a future fast recovery without browsing all the items stored in the personal repository;
- Timestamp of the query or selection assigned by the user (Mandatory): this is useful for expanded or actualized selections since people can have more than one snapshot of the same selection;
- List of groups of users that are entitled to operate on the selection (Optional);
- List of keyword (Optional): the user assign a list of keywords for recovering queries and selection by keywords instead of directly pointing to a selection;

The described parameters allow the maximum flexibility in browsing, recovering and selecting query and selections.

Query and Selection Archive for Each User has to provide some services for allowing the user to interact with selection stored in each personal profile.

Since Query can be regarded as a selection with only a query inside, all queries will be stored as selections.

A first definition of the services provided is:

- List all selection created by the user;
- List all selection for which the user is entitled (this is because a user can create a selection that can be managed by a group of users);
- Load a selection;
- Save a selection (several selection with the same name but different timestamps can exist in the database);
- Delete a selection;

This module offers its service with a web service interface that operates according to the WSDL and methods described in the following.

Selection_Archive	
Method	Description
ListUserSelection	This method will return all the selections created by the user
ListEntitledSelection	This method will return all the selections for which the user is entitled
LoadSelection	This method get a selection from the AXDB
SaveSelection	This method save a selection to the AXDB
DeleteSelection	This method eliminate a selection for the AXDB
ActualizeSelection_sync	This methods return synchronously the list of AXOID that the selection represent in that moment. The list of AXOID is in the format of a selection as formalized in the selection definition
ActualizeSelection_async	This methods return asynchronously the list of AXOID that the selection represent in that moment. The list of AXOID is in the format of a selection as formalized in the selection definition and will be returned to a listener provided by the who uses this method. A possible listener is proposed.

Selection_Archive	
WSDL	<pre><?xml version="1.0" encoding="UTF-8"?> <definitions name="Selection_Archive" targetNamespace="http://www.axmedis.org/selection_archive.wsdl" xmlns:tns="http://www.axmedis.org/selection_archive.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"></pre>

	<pre> 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="SelectionDetail"> <sequence> <element name="Id" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="Name" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="Timestamp" type="xsd:string" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <complexType name="SL"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="NumberOfSelection" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="Selection" type="ax:SelectionDetail" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType> <complexType name="selection"> <sequence> <element name="AXOID" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType> <complexType name="AL"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="NumberOfAXOID" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="selection" type="ax:selection" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <!-- operation request element --> <element name="user" type="xsd:string"/> <!-- operation request element --> <element name="pwd" type="xsd:string"/> <!-- operation response element --> <element name="selectionresult" type="ax:SL"/> <!-- operation request element --> <element name="SelectionID" type="xsd:string"/> <!-- operation response element --> <element name="Selection" type="xsd:string"/> <!-- operation response element --> <element name="result" type="xsd:boolean"/> <!-- operation request element --> <element name="selectionID" type="xsd:string"/> <!-- operation response element --> <element name="axoidresult" type="ax:AL"/> <!-- operation request element --> <element name="User" type="xsd:string"/> <!-- operation request element --> <element name="Password" type="xsd:string"/> <!-- operation request element --> <element name="ActualizeListenerService" type="xsd:anyURI"/> </pre>
--	---

	<pre> <!-- operation request element --> <element name="ListenerID" type="xsd:string"/> </schema> </types> <message name="ListUserSelectionRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> </message> <message name="SelectionList"> <part name="selectionresult" element="ax:selectionresult"/> </message> <message name="ListEntitledSelectionRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> </message> <message name="LoadSelectionRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> <part name="SelectionID" element="ax:SelectionID"/> </message> <message name="LoadSelectionResponse"> <part name="Selection" element="ax:Selection"/> </message> <message name="SaveSelectionRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> <part name="Selection" element="ax:Selection"/> </message> <message name="SaveSelectionResponse"> <part name="result" element="ax:result"/> </message> <message name="DeleteSelectionRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> <part name="SelectionID" element="ax:SelectionID"/> </message> <message name="DeleteSelectionResponse"> <part name="result" element="ax:result"/> </message> <message name="ActualizeSelection-syncRequest"> <part name="user" element="ax:user"/> <part name="pwd" element="ax:pwd"/> <part name="selectionID" element="ax:selectionID"/> </message> <message name="AXOIDList"> <part name="axoidresult" element="ax:axoidresult"/> </message> <message name="ActualizeSelection-asyncRequest"> <part name="User" element="ax:User"/> <part name="Password" element="ax:Password"/> <part name="selectionID" element="ax:selectionID"/> <part name="ActualizeListenerService" element="ax:ActualizeListenerService"/> <part name="ListenerID" element="ax:ListenerID"/> </pre>
--	---

	<pre> </message> <message name="ActualizeSelection-asyncResponse"> <part name="result" element="ax:result"/> </message> <portType name="Selection_ArchivePortType"> <operation name="ListUserSelection"> <documentation>Service definition of function ax__ListUserSelection</documentation> <input message="tns:ListUserSelectionRequest"/> <output message="tns:SelectionList"/> </operation> <operation name="ListEntitledSelection"> <documentation>Service definition of function ax__ListEntitledSelection</documentation> <input message="tns:ListEntitledSelectionRequest"/> <output message="tns:SelectionList"/> </operation> <operation name="LoadSelection"> <documentation>Service definition of function ax__LoadSelection</documentation> <input message="tns:LoadSelectionRequest"/> <output message="tns:LoadSelectionResponse"/> </operation> <operation name="SaveSelection"> <documentation>Service definition of function ax__SaveSelection</documentation> <input message="tns:SaveSelectionRequest"/> <output message="tns:SaveSelectionResponse"/> </operation> <operation name="DeleteSelection"> <documentation>Service definition of function ax__DeleteSelection</documentation> <input message="tns:DeleteSelectionRequest"/> <output message="tns:DeleteSelectionResponse"/> </operation> <operation name="ActualizeSelection-sync"> <documentation>Service definition of function ax__ActualizeSelection_sync</documentation> <input message="tns:ActualizeSelection-syncRequest"/> <output message="tns:AXOIDList"/> </operation> <operation name="ActualizeSelection-async"> <documentation>Service definition of function ax__ActualizeSelection_async</documentation> <input message="tns:ActualizeSelection-asyncRequest"/> <output message="tns:ActualizeSelection-asyncResponse"/> </operation> </portType> <binding name="Selection_Archive" type="tns:Selection_ArchivePortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"> <operation name="ListUserSelection"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="ListEntitledSelection"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="LoadSelection"> <SOAP:operation style="rpc" soapAction=""/> </pre>
--	--

	<pre> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="SaveSelection"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="DeleteSelection"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="ActualizeSelection-sync"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ax"/> </input> <output> <SOAP:body use="literal" namespace="urn:ax"/> </output> </operation> <operation name="ActualizeSelection-async"> <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="Selection_Archive"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="Selection_Archive" binding="tns:Selection_Archive"> <SOAP:address location="http://www.axmedis.org/selection_archive.cgi"/> </port> </service> </definitions> </pre>
--	---

Selection_Archive	
Method	DeleteSelection
Description	This method allows to delete a selection that is present in the archive
Input parameters	xsd:string user: user name for authentication and authorization xsd:string pwd: password for authentication and authorization xsd:string SelectionID: ID of the selection to be deleted

Output parameters	xsd:boolean result: true or false depending on the fact that the selection has been deleted or not.
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:DeleteSelection> <user>username</user> <pwd>password</pwd> <SelectionID>234543grre</SelectionID> </ax:DeleteSelection> </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:DeleteSelectionResponse> <result>true</result> </ax:DeleteSelectionResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>

Selection Archive	
Method	SaveSelection
Description	This method allow to save a selection created by the user
Input parameters	xsd:string user: user name for authentication and authorization xsd:string pwd: password for authentication and authorization xsd:string Selection: the selection in XML format that have to be stored
Output parameters	xsd:boolean result: true of false depending on the success of the save 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:SaveSelection> <user>username</user> <pwd>password</pwd> <Selection>342352435435</Selection> </ax:SaveSelection> </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></pre>

	<pre><ax:SaveSelectionResponse> <result>true</result> </ax:SaveSelectionResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>
--	---

Selection Archive	
Method	LoadSelection
Description	this methods allows to load a selection from the archive
Input parameters	xsd:string user: user name for authentication and authorization xsd:string pwd: password for authentication and authorization xsd:string SelectionID: ID of the selection to be loaded
Output parameters	xsd:string Selection: the selection in XML format that have been loaded
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:LoadSelection> <user>username</user> <pwd>password</pwd> <SelectionID>134324wefrq423</SelectionID> </ax:LoadSelection> </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:LoadSelectionResponse> <Selection>.....selection in the XML format.....</Selection> </ax:LoadSelectionResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>

Selection Archive	
Method	ListUserSelection
Description	This method returns the selection created by the user
Input parameters	xsd:string user: user name for authentication and authorization xsd:string pwd: password for authentication and authorization
Output parameters	ax:SelectionList Selections: list of selection that have been created by the user; this parameter is a sequence of: xsd:int RetCode, that is the return code, where 0 means operation success xsd:int NumberOfSelection, that is the number of selection returned a list of ax:SelectionDetail, where each element is defined by: xsd:string Id, that is the ID of the Selection xsd:string Name, that is the name of the selection xsd:string Timestamp, that is the timestamp of the selection
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/"</pre>

	<pre> 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:ListUserSelection> <user>username</user> <pwd>password</pwd> </ax:ListUserSelection> </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:SelectionList> <result> <RetCode>0</RetCode> <NumberOfSelection>2</NumberOfSelection> <Selection> <Id>43215423rffe</Id> <Name>My selection1</Name> <Timestamp>2005-01-30T12:12:00</Timestamp> </Selection> <Selection> <Id>43654teth</Id> <Name>My selection 4</Name> <Timestamp>2005-02-03T16:14:50</Timestamp> </Selection> </result> </ax:SelectionList> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Selection Archive	
Method	ListEntitledSelection
Description	This method returns the selection for which the user is entitled that are the selection created by the user and the selection for which has been entitled as the user become to a group
Input parameters	xsd:string user: user name for authentication and authorization xsd:string pwd: password for authentication and authorization
Output parameters	ax:SelectionList Selections: list of selection that have been created by the user; this parameter is a sequence of: xsd:int RetCode, that is the return code, where 0 means operation success xsd:int NumberOfSelection, that is the number of selection returned a list of ax:SelectionDetail, where each element is defined by: xsd:string Id, that is the ID of the Selection xsd:string Name, that is the name of the selection xsd:string Timestamp, that is the timestamp of the selection
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"> </pre>

	<pre> <SOAP-ENV:Body> <ax:ListEntitledSelection> <user>username</user> <pwd>password</pwd> </ax:ListEntitledSelection> </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-SelectionList> <result> <RetCode>0</RetCode> <NumberOfSelection>1</NumberOfSelection> <Selection> <Id>254235243</Id> <Name>user selection</Name> <Timestamp>2005-02-01T18:45:45</Timestamp> </Selection> </result> </ax-SelectionList> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Selection Archive	
Method	ListUserSelection
Description	This method returns the selection created by the user
Input parameters	xsd:string user: user name for authentication and authorization xsd:string pwd: password for authentication and authorization
Output parameters	ax:SelectionList Selections: list of selection that have been created by the user; this parameter is a sequence of: <ul style="list-style-type: none"> xsd:int RetCode, that is the return code, where 0 means operation success xsd:int NumberOfSelection, that is the number of selection returned a list of ax:SelectionDetail, where each element is defined by: <ul style="list-style-type: none"> xsd:string Id, that is the ID of the Selection xsd:string Name, that is the name of the selection xsd:string Timestamp, that is the timestamp of the selection
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:ListUserSelection> <user>username</user> <pwd>password</pwd> </ax:ListUserSelection> </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/" </pre>

	<pre> 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-SelectionList> <result> <RetCode>0</RetCode> <NumberOfSelection>2</NumberOfSelection> <Selection> <Id>43215423rffe</Id> <Name>My selection 1</Name> <Timestamp>2005-01-30T12:12:00</Timestamp> </Selection> <Selection> <Id>43654teth</Id> <Name>My selection 4</Name> <Timestamp>2005-02-03T16:14:50</Timestamp> </Selection> </result> </ax-SelectionList> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
--	--

Selection Archive	
Method	ActualizeSelection_sync
Description	This methods return synchronously the list of AXOID that the selection represent in that moment. The list of AXOID is in the format of a selection as formalized in the selection definition
Input parameters	xsd:string user: username to the authorized to perform the operation xsd:string pwd: password of the user for authentication xsd:string selectionID: ID of the selection that have to be actualized
Output parameters	is a complex type which parts are: xsd::int RetCode xsd::int NumberofAXOID ax:selection selection, where selection is a list xsd:string representing AXOID
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:ActualizeSelection-sync> <user>username</user> <pwd>mypassword</pwd> <selectionID>452543543</selectionID> </ax:ActualizeSelection-sync> </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:AXOIDList> <axoidresult> </pre>

	<pre> <RetCode>0</RetCode> <NumberOfAXOID>4</NumberOfAXOID> <selection> <AXOID>34524frr43</AXOID> <AXOID>344werrdfr43</AXOID> <AXOID>6775gg43</AXOID> <AXOID>fer5ft55</AXOID> </selection> </axoidresult> </ax:AXOIDList> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
--	--

Selection Archive	
Method	ActualizeSelection_async
Description	This methods return asynchronously the list of AXOID that the selection represent in that moment. The list of AXOID is in the format of a selection as formalized in the selection definition and will be returned to a listener provided by the who uses this method. A possible listener is proposed.
Input parameters	xsd:string user: username to the authorized to perform the operation xsd:string pwd: password of the user for authentication xsd:string selectionID: ID of the selection that have to be actualized xsd:anyURI ActualizeListenerService: address of the listening web service xsd:string ListenerID: ID of the notification
Output parameters	xsd:Boolean result: true if the operation has been put in the queue of the operations to be performed
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:ActualizeSelection-async> <User>username</User> <Password>mypassword</Password> <selectionID>32432rffcf44</selectionID> <ActualizeListenerService>http://mylistener/listener</ActualizeListenerService> <ListenerID>123</ListenerID> </ax:ActualizeSelection-async> </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:ActualizeSelection-asyncResponse> <result>true</result> </ax:ActualizeSelection-asyncResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

4.6.1 Actualize Listener

In this paragraph a model for the listener to be implemented is present. It will have only one method named QueryResult_Listener to which all the results will be communicated as soon as they are ready.

ActualizeListener	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="ActualizeListener" targetNamespace="http://www.someone.org/actualizelistener.wsdl" xmlns:tns="http://www.someone.org/actualizelistener.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:ns/ax.xsd" xmlns:ns="urn:ns" 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:ns/ax.xsd" 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:ns/ax.xsd" xmlns:ns="urn:ns" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="selection"> <sequence> <element name="AXOID" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType> <complexType name="sr"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="NumberOfresult" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="selection" type="ax:selection" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <complexType name="selectionresults"> <sequence> <element name="result" type="ax:sr" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> </schema> <schema targetNamespace="urn:ns" 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:ns/ax.xsd" xmlns:ns="urn:ns" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" attributeFormDefault="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <!-- operation request element --> <element name="result" type="ax:selectionresults"/> <!-- operation request element --> <element name="ListenerID" type="xsd:string"/> <!-- operation response element --> </pre>

	<pre> <element name="r" type="xsd:boolean"/> </schema> </types> <message name="QueryResult-ListenerRequest"> <part name="result" element="ns:result"/> <part name="ListenerID" element="ns:ListenerID"/> </message> <message name="QueryResult-ListenerResponse"> <part name="r" element="ns:r"/> </message> <portType name="ActualizeListenerPortType"> <operation name="QueryResult-Listener"> <documentation>Service definition of function ns__QueryResult_Listener</documentation> <input message="tns:QueryResult-ListenerRequest"/> <output message="tns:QueryResult-ListenerResponse"/> </operation> </portType> <binding name="ActualizeListener" type="tns:ActualizeListenerPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="QueryResult-Listener"> <SOAP:operation style="rpc" soapAction=""/> <input> <SOAP:body use="literal" namespace="urn:ns"/> </input> <output> <SOAP:body use="literal" namespace="urn:ns"/> </output> </operation> </binding> <service name="ActualizeListener"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="ActualizeListener" binding="tns:ActualizeListener"> <SOAP:address location="http://www.someone.org/actualizelistener.cgi"/> </port> </service> </definitions> </pre>
Method	QueryResult_Listener
Description	This method is called to notify the actualized result of a selection
Input parameters	<p>ax:selectionresults result, that is a complex:</p> <ul style="list-style-type: none"> xsd:int RetCode: that is the error code of the operation (0 means OK) xsd:int NumberOfresult: that is the number of results sent in terms of number of AXOID ax:selection selection, that is a list of xsd:string that represent the AXOIDs <p>xsd:string ListenerID: id of the operation</p>
Output parameters	xsd:boolean r: Boolean ack of the receipt of data
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:ns/ax.xsd" xmlns:ns="urn:ns"> <SOAP-ENV:Body> <ns:QueryResult-Listener> <result> </pre>

	<pre> </result> <RetCode>0</RetCode> <NumberOfresult>4</NumberOfresult> <selection> <AXOID>23563fre3</AXOID> <AXOID>23324563fre3</AXOID> <AXOID>23532463fre3</AXOID> <AXOID>23563f324re3</AXOID> </selection> </result> </result> <ListenerID/> </ns:QueryResult-Listener> </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:ns/ax.xsd" xmlns:ns="urn:ns"> <SOAP-ENV:Body> <ns:QueryResult-ListenerResponse> <r>true</r> </ns:QueryResult-ListenerResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

4.7 Query Support WEB Service Interface (EXITECH)

Module Profile		
Query Support WEB Service Interface		
Executable or Library(Support)	Executable (WebService)	
Single Thread or Multithread	Multithread	
Language of Development	JAVA	
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.)
All tool and engine plus Query Distribution and Query Result Integration		
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.
None	WebService	

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

The Web service technology is a widely adopted methods for exchanging information among servers distributed in a LAN/WAN environment. According to www.w3.org the definition of web service is:

“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

The Query Support web service interface is the module that offers an external interface for querying the AXMEDIX database (as well as AXEPTool and Crawling system) and its interface must be suitably defined by a WSDL that have to take care of the general structure of *AXMEDIS data structure schema*, in terms of query fields that can be adopted.

This module will have also the need for a database for supporting the operation of query distribution and Integration.

The ER diagram for the database section related to query support web interface is reported in section 3.5.4, while in the following the explanation of the tables that have been projected is commented.

Once the Query Support Web Service Interface receives a query, it passes the query to the Query Distribution module that stores the query in the Received Table with the reception timestamp and generating an unique ID for the query.

Once the query has been processed and optimized for the channels, a set of different queries are issued on the different channels. For each query a record in the Distributed table is created. The record comprises the queryID, the timestamp and the channel over which the query has been issued.

Each time a channel provide an answer asynchronously

With respect to the query, the Query Result Integration module write a record in the Integrated Table, setting the QueryID, the Channel from which the result has been received, the timestamp and the result.

Results will be back to the query issuer in two manners:

- Asynchronous: each time a result is obtained from one of the channel it is passed back to the requester
- Synchronous: Integrator waits to have all results before sending back the results set to the requester.

The database has to support both modes.

Since we have a webservice, we have two choices for sending and receiving a query, the first is to map inside a soap message and therefore also in the programming language the complex type that has been modelled for the query. This can be an hard task to be done since the query structure is very complex, but allows a formal verification on the query.

The second approach consists in putting the XML query in a string (a simple type for a SOAP message) that in easily mapped both in C++ and Java. This string can then be parsed by an XML parser, but it is impossible to check the correctness of the soap message with a schema before it is processed.

From a preliminary check with the identified library for webservices in C++, it has been evident that the mapping in C++ is very hard to be obtained and requires several manipulation to bring to an acceptable code that can be processed by gsoap parser for client and server side and for , and therefore, at least in this phase, where the query structure will be probably revised it is preferred to adopt a string approach. This approach will be revised in the second phase of the specification when the model for the query and for the query results will be more stable.

The same approach will be used for results also both in synchronous and asynchronous mode.

The model of the web service is detailed below:

Query_Support

Method	Description
Make_Query_Sync	Method that allows to synchronously issue a query and get the corresponding result.
Make_Query_ASync	Method that allows to asynchronously issue a query and get the corresponding result on a listener specified by the user.

Query_Support	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="Query_Support" targetNamespace="http://www.axmedis.org/query_support.wsdl" xmlns:tns="http://www.axmedis.org/query_support.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="q"> <sequence> <element name="user" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="pwd" type="xsd:string" minOccurs="1" maxOccurs="1"/> <element name="query" type="xsd:string" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <complexType name="query"> <sequence> <element name="send" type="ax:q" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <complexType name="qr"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="XMLResult" type="xsd:string" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> <!-- operation request element --> <element name="queryparm" type="ax:query"/> <!-- operation response element --> <element name="return" type="ax:qr"/> <!-- operation request element --> <element name="QueryresultListenerService" type="xsd:anyURI"/> <!-- operation request element --> <element name="ListenerID" type="xsd:string"/> <!-- operation response element --> <element name="result" type="xsd:boolean"/> </schema> </pre>

```

</types>

<message name="Make-Query-SyncRequest">
  <part name="queryparm" element="ax:queryparm"/>
</message>

<message name="queryresults">
  <part name="return" element="ax:return"/>
</message>

<message name="Make-Query-ASyncRequest">
  <part name="queryparm" element="ax:queryparm"/>
  <part name="QueryresultListenerService" element="ax:QueryresultListenerService"/>
  <part name="ListenerID" element="ax:ListenerID"/>
</message>

<message name="Make-Query-ASyncResponse">
  <part name="result" element="ax:result"/>
</message>

<portType name="Query_SupportPortType">
  <operation name="Make-Query-Sync">
    <documentation>Service definition of function ax__Make_Query_Sync</documentation>
    <input message="tns:Make-Query-SyncRequest"/>
    <output message="tns:queryresults"/>
  </operation>
  <operation name="Make-Query-ASync">
    <documentation>Service definition of function ax__Make_Query_ASync</documentation>
    <input message="tns:Make-Query-ASyncRequest"/>
    <output message="tns:Make-Query-ASyncResponse"/>
  </operation>
</portType>

<binding name="Query_Support" type="tns:Query_SupportPortType">
  <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="Make-Query-Sync">
    <SOAP:operation style="rpc" soapAction=""/>
    <input>
      <SOAP:body use="literal" namespace="urn:ax"/>
    </input>
    <output>
      <SOAP:body use="literal" namespace="urn:ax"/>
    </output>
  </operation>
  <operation name="Make-Query-ASync">
    <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="Query_Support">
  <documentation>gSOAP 2.7.0e generated service definition</documentation>
  <port name="Query_Support" binding="tns:Query_Support">
    <SOAP:address location="http://www.axmedis.org/query_support.cgi"/>
  </port>
</service>

</definitions>

```

Query_Support	
Method	Make_Query_Sync
Description	Method that allows to synchronously issue a query and get the corresponding result.
Input parameters	ax:query, that is a complex type composed by a list of: xsd:string user, user for authentication xsd:string pwd, password for authentication xsd:string query, query to be performed
Output parameters	ax :queryresults, that is a complex type composed by a list of : xsd:int RetCode, that is the return code of the query, 0 if OK, other if an error occurs; xsd:string XMLResult, that is the result of the query in the format specified by the query result schema defined in this document
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:Make-Query-Sync> <queryparm> <send> <user>username</user> <pwd>password</pwd> <query>&lt;query&gt;&lt;source&gt;&lt;location&gt;CRAWLER&lt;/location &gt;&lt;/location&gt;AXEPTOOL&lt;/location&gt;&lt;location&gt;AXDB&lt;/location &gt;&lt;/source&gt;&lt;AXinfoQuery&gt;&lt;querycondition&gt;&lt;nesting &gt;&lt;/nesting&gt;&lt;test&gt;&lt;field&gt;AUTHOR&lt;/field &gt;&lt;/operator&gt;EQ&lt;/operator&gt;&lt;value&gt;BOTTICELLI&lt;/value &gt;&lt;/test&gt;&lt;/nesting&gt;&lt;and&gt;&lt;/nesting&gt;&lt;test&gt;&lt;field &gt;MEDIA&lt;/field&gt;&lt;/operator&gt;EQ&lt;/operator&gt;&lt;value &gt;VIDEO&lt;/value&gt;&lt;/test&gt;&lt;/or&gt;&lt;/test&gt;&lt;/field &gt;MEDIA&lt;/field&gt;&lt;/operator&gt;EQ&lt;/operator&gt;&lt;value &gt;AUDIO&lt;/value&gt;&lt;/test&gt;&lt;/or&gt;&lt;/test&gt;&lt;/field &gt;MEDIA&lt;/field&gt;&lt;/operator&gt;EQ&lt;/operator&gt;&lt;value &gt;TEXT&lt;/value&gt;&lt;/test&gt;&lt;/nesting&gt;&lt;/nesting&gt;&lt;/and/ &gt;&lt;/nesting&gt;&lt;test&gt;&lt;field&gt;IPR&lt;/field&gt;&lt;/operator &gt;EQ&lt;/operator&gt;&lt;value&gt;FREE&lt;/value&gt;&lt;/test&gt;&lt;/and/ &gt;&lt;/test&gt;&lt;/field&gt;COST&lt;/field&gt;&lt;/operator&gt;LT&lt;/operator &gt;&lt;value&gt;10.00&lt;/value&gt;&lt;/test&gt;&lt;/nesting&gt;&lt;/nesting &gt;&lt;/querycondition&gt;&lt;/AXinfoQuery&gt;&lt;/query&gt;</query> </send> </queryparm> </ax:Make-Query-Sync> </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:queryresults> <return> <RetCode>0</RetCode> </return> </ax:queryresults> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

	<pre> <XMLResult>&lt;queryresults&gt;&lt;AXObject&gt;&lt;AXEPTOOL &gt;&lt;peers&gt;&lt;peerID&gt;1234&lt;/peerID&gt;&lt;peerID&gt;3434&lt;/peerID &gt;&lt;peerID&gt;5634&lt;/peerID&gt;&lt;peers&gt;&lt;AXOID&gt;3243543654365 &lt;/AXOID&gt;&lt;/AXEPTOOL&gt;&lt;/AXObject&gt;&lt;AXObject&gt;&lt;AXDB &gt;&lt;AXOID&gt;21338921742734&lt;/AXOID&gt;&lt;/AXDB&gt;&lt;/AXObject &gt;&lt;AXObject&gt;&lt;CRAWLER&gt;&lt;TAXOID &gt;JHGSiheduwjsdnhIIUUYUYU(/HSHHJsjewfhweuhfwekkhlcwe&lt;/TAXOID &gt;&lt;/CRAWLER&gt;&lt;/AXObject&gt;&lt;/queryresults&gt;</XMLResult> </return> </ax:queryresults> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
--	--

Query_Support	
Method	Make_Query_ASync
Description	Method that allows to asynchronously issue a query and get the corresponding result on a listener specified by the user.
Input parameters	<p>ax:query , that is a complex type composed by a list of:</p> <ul style="list-style-type: none"> xsd:string user, user for authentication xsd:string pwd, password for authentication xsd:string query, query to be performed <p>xsd:anyURI QueryresultListenerService, that is the URI of the listener to which results have to be provided</p> <p>xsd:string ListenerID, listenerID that identifies on the listener the ID of the request</p>
Output parameters	xsd:boolean result that is true or false depending on the fact that the operation has been issued or not.
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:Make-Query-AAsync> <queryparm> <send> <user>username</user> <pwd>password</pwd> <query>&lt;query&gt;&lt;source&gt;&lt;location&gt;CRAWLER&lt;/location &gt;&lt;location&gt;AXEPTOOL&lt;/location&gt;&lt;location&gt;AXDB&lt;/location &gt;&lt;/source&gt;&lt;AXinfoQuery&gt;&lt;querycondition&gt;&lt;nesting &gt;&lt;nesting&gt;&lt;nesting&gt;&lt;test&gt;&lt;field&gt;AUTHOR&lt;/field &gt;&lt;operator&gt;EQ&lt;/operator&gt;&lt;value&gt;BOTTICELLI&lt;/value &gt;&lt;/test&gt;&lt;/nesting&gt;&lt;and&gt;&lt;nesting&gt;&lt;test&gt;&lt;field &gt;&lt;MEDIA&lt;/field&gt;&lt;operator&gt;EQ&lt;/operator&gt;&lt;value &gt;&lt;VIDEO&lt;/value&gt;&lt;/test&gt;&lt;/or&gt;&lt;test&gt;&lt;field &gt;&lt;MEDIA&lt;/field&gt;&lt;operator&gt;EQ&lt;/operator&gt;&lt;value &gt;&lt;AUDIO&lt;/value&gt;&lt;/test&gt;&lt;/or&gt;&lt;test&gt;&lt;field &gt;&lt;MEDIA&lt;/field&gt;&lt;operator&gt;EQ&lt;/operator&gt;&lt;value &gt;&lt;TEXT&lt;/value&gt;&lt;/test&gt;&lt;/nesting&gt;&lt;/nesting&gt;&lt;and/ &gt;&lt;nesting&gt;&lt;test&gt;&lt;field&gt;IPR&lt;/field&gt;&lt;operator &gt;&lt;EQ&lt;/operator&gt;&lt;value&gt;FREE&lt;/value&gt;&lt;/test&gt;&lt;/and/ &gt;&lt;test&gt;&lt;field&gt;COST&lt;/field&gt;&lt;operator&gt;LT&lt;/operator &gt;&lt;value&gt;10.00&lt;/value&gt;&lt;/test&gt;&lt;/nesting&gt;&lt;/nesting &gt;&lt;/querycondition&gt;&lt;/AXinfoQuery&gt;&lt;/query&gt;</query> </send> </queryparm> </ax:Make-Query-AAsync> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

	<pre> <QueryresultListenerService>http://query.results.org/listener</QueryresultListenerService> <ListenerID>hr32y73yriue</ListenerID> </ax:Make-Query-ASync> </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:Make-Query-ASyncResponse> <result>true</result> </ax:Make-Query-ASyncResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

Since a listener is defined in the asynchronous mode, it is necessary to draw the specification also of this webservice that will be offered and implemented by a third party that uses the asynchronous query result mechanism.

QueryResultListener	
WSDL	<pre> <?xml version="1.0" encoding="UTF-8"?> <definitions name="QueryResultListener" targetNamespace="http://www.someone.org/querylistener.wsdl" :tns="http://www.someone.org/querylistener.wsdl" :SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" :SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" :xsi="http://www.w3.org/2001/XMLSchema-instance" :xsd="http://www.w3.org/2001/XMLSchema" :ax="urn:ns/ax.xsd" :ns="urn:ns" :SOAP="http://schemas.xmlsoap.org/wsdl/soap/" :MIME="http://schemas.xmlsoap.org/wsdl/mime/" :DIME="http://schemas.xmlsoap.org/ws/2002/04/dime/wsdl/" :WSDL="http://schemas.xmlsoap.org/wsdl/" ="http://schemas.xmlsoap.org/wsdl/"> <types> <schema targetNamespace="urn:ns/ax.xsd" :SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" :SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" :xsi="http://www.w3.org/2001/XMLSchema-instance" :xsd="http://www.w3.org/2001/XMLSchema" xmlns:ax="urn:ns/ax.xsd" :ns="urn:ns" xmlns="http://www.w3.org/2001/XMLSchema" ="unqualified" ="unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <complexType name="qr"> <sequence> <element name="RetCode" type="xsd:int" minOccurs="1" maxOccurs="1"/> <element name="XMLResult" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/> </sequence> </complexType> <complexType name="queryresults"> <sequence> <element name="return" type="ax:qr" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> </schema> <schema targetNamespace="urn:ns" :SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" :SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" :xsi="http://www.w3.org/2001/XMLSchema-instance" :xsd="http://www.w3.org/2001/XMLSchema" :ax="urn:ns/ax.xsd" :ns="urn:ns"> </pre>

	<pre> =<http://www.w3.org/2001/XMLSchema =<unqualified" =<unqualified"> <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /> <!-- operation request element --> <element name="result" type="ax:queryresults"/> <!-- operation request element --> <element name="ListenerID" type="xsd:string"/> <!-- operation response element --> <element name="r" type="xsd:boolean"/> </schema> </types> <message name="QueryResult-ListenerRequest"> <part name="result" element="ns:result"/> <part name="ListenerID" element="ns:ListenerID"/> </message> <message name="QueryResult-ListenerResponse"> <part name="r" element="ns:r"/> </message> <portType name="QueryResultistenerPortType"> <operation name="QueryResult-Listener"> <documentation>Service definition of function ns__QueryResult_Listener</documentation> <input message="tns:QueryResult-ListenerRequest"/> <output message="tns:QueryResult-ListenerResponse"/> </operation> </portType> <binding name="QueryResultistener" type="tns:QueryResultistenerPortType"> <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="QueryResult-Listener"> <SOAP:operation style="rpc" soapAction="" /> <input> <SOAP:body use="literal" namespace="urn:ns"/> </input> <output> <SOAP:body use="literal" namespace="urn:ns"/> </output> </operation> </binding> <service name="QueryResultistener"> <documentation>gSOAP 2.7.0e generated service definition</documentation> <port name="QueryResultistener" binding="tns:QueryResultistener"> <SOAP:address location="http://www.someone.org/querylistener.cgi"/> </port> </service> </definitions> </pre>
Method	QueryResult_Listener
Description	This service will wait for the results generated by the query results and will consume them in asynchronous manner differentiating results
Input parameters	<p>ax:queryresult , that is a complex type composed by a list of:</p> <p>xsd:int RetCode, this return code will be 0 in the case a list of results is provided, greater than zero in the case we have an error, lower than 0 if this is the last set of results for the query.</p> <p>xsd:string XMLResult, the results if the query in the predefined format.</p> <p>xsd:string ListenerID, listenerID that identifies on the listener the ID of the request</p>
Output parameters	xsd:boolean r, true if the result has been processed, false in all the other cases.
Request Sample Message	<pre> <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope :SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" :SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" :xsi="http://www.w3.org/2001/XMLSchema-instance" :xsd="http://www.w3.org/2001/XMLSchema" :ax="urn:ns/ax.xsd" :ns="urn:ns"> <SOAP-ENV:Body> <ns:QueryResult-Listener> </pre>

	<pre> </result> </return> <RetCode>0</RetCode> <XMLResult>&lt;&lt;queryresults&gt;&lt;&lt;AXObject&gt;&lt;&lt;AXEPTOOL &gt;&lt;&lt;peers&gt;&lt;&lt;peerID&gt;&lt;1234&lt;/peerID&gt;&lt;&lt;peerID&gt;&lt;3434&lt;/peerID &gt;&lt;&lt;peerID&gt;&lt;5634&lt;/peerID&gt;&lt;&lt;peers&gt;&lt;&lt;AXOID&gt;&lt;3243543654365 &lt;/AXOID&gt;&lt;&lt;AXEPTOOL&gt;&lt;&lt;AXObject&gt;&lt;&lt;AXObject&gt;&lt;&lt;AXDB &gt;&lt;&lt;AXOID&gt;&lt;21338921742734&lt;/AXOID&gt;&lt;&lt;AXDB&gt;&lt;&lt;AXObject &gt;&lt;&lt;AXObject&gt;&lt;&lt;CRAWLER&gt;&lt;&lt;TAXOID &gt;&lt;&lt;JHGSiheduwjdsnhIIUUYUYU(/HSHHJsJwefhweuhfwekhlhwe&lt;/TAXOID &gt;&lt;&lt;/CRAWLER&gt;&lt;&lt;AXObject&gt;&lt;&lt;/queryresults&gt;&lt;</XMLResult> </return> </result> <ListenerID>2342344ret3t542gt</ListenerID> </ns:QueryResult-Listener> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
Response Sample Message	<pre> <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xsi="http://www.w3.org/2001/XMLSchema-instance" xsd="http://www.w3.org/2001/XMLSchema" ax="urn:ns:ax.xsd" ns="urn:ns"> <SOAP-ENV:Body> <ns:QueryResult-ListenerResponse> <r>true</r> </ns:QueryResult-ListenerResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

4.8 Query Distribution (EXITECH)

Module Profile		
Query Distribution		
Executable or Library(Support)	Executable	
Single Thread or Multithread	Multithread	
Language of Development	JAVA	
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.)
Collector Engine Query Support interface		
AXMEDIS database manager		
AXEPTOOL Query support Interface		
File Formats Used	Shared with	File format name or reference to a section
XML, WSDL		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
None	Web Service Client	

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

Query distribution is a module that is capable of distributing a query to different subsystems that are able to accept queries from the query support in order to collect result in a different phase by the means of the Query Result Integration module.

This module collects queries from the Query Support WebService Interface, assign a unique ID to the query in order to have the capability of collecting in the future the results, and store in temporary DB the queries and the associated IDs with the channel over the query has been issued.

This module in mainly a web service client of the tool it interfaces:

- Collector Engine Query Support interface
- AXMEDIS database manager
- AXEPTOOL Query support Interface

All these tools have to export a WebService interface that accept a query and must be capable of issuing notifications to the Query Result Integrator, by the means of a web service call, that will be discussed in the paragraph related to the Integrator.

In order to simplify the work, it is suggested to adopt the same WSDL provided by the query support to exchange results in response to a query both for synchronous and asynchronous model.

In the case, the synchronous mode will be requested, Query Distribution module will save directly on the DB the results or will call directly the Query results Integrator for doing such operation.

4.9 Query Results Integration (EXITECH)

Module Profile		
Query Results Integration		
Executable or Library(Support)	Executable	
Single Thread or Multithread	Multithread	
Language of Development	JAVA	
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.)
Collector Engine Query Support interface		
AXMEDIS database manager		
AXEPTOOL Query support Interface		
File Formats Used	Shared with	File format name or reference to a section
XML, WSDL		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.

None	Web service	
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

This module interact via SOAP messages with all the web services capable of giving information on multimedia contents in the AXMEDIS architecture. By now it is limited to the interaction with AXMEDIS, Crawler component and AXEPTool but the fact the it interrogates web services allow to extend its field of action to different entities that can arise during or after the project.

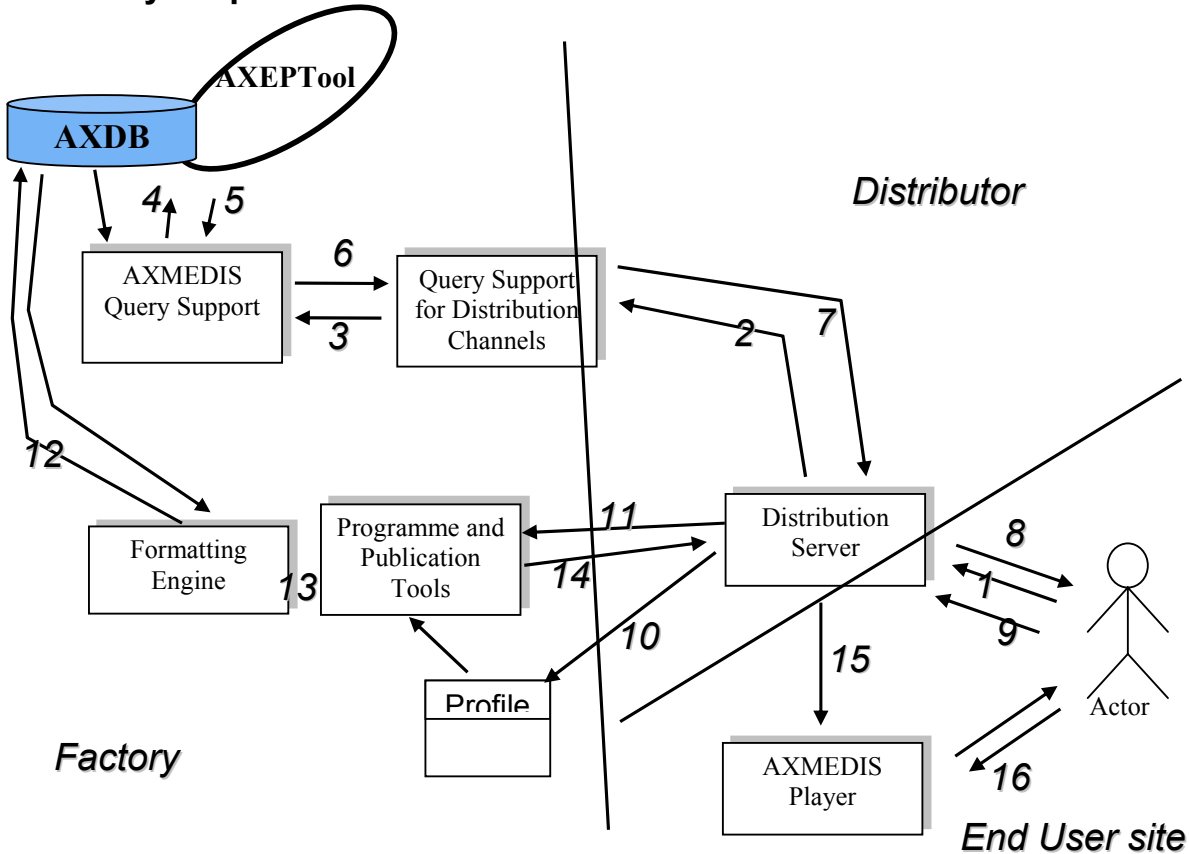
This module collect the results of the query from all the web services capable of returning information and give back a cumulative response to the Query support web service interface that in turn communicates to the GUI that can visualize the results.

This is also a QueryResultListener, since its architecture is compliant with that defined for the asynchronous return of results from a result source.

This module exports a web service interface for receiving the notification of the tool that will provide query results to the Query Support.

Depending on the operation mode (synchronous or asynchronous) the Query Results Intergration module will notify results to the Query Support WebService Interface that in turn will reorganize results taking them from the database described in section 4.7.

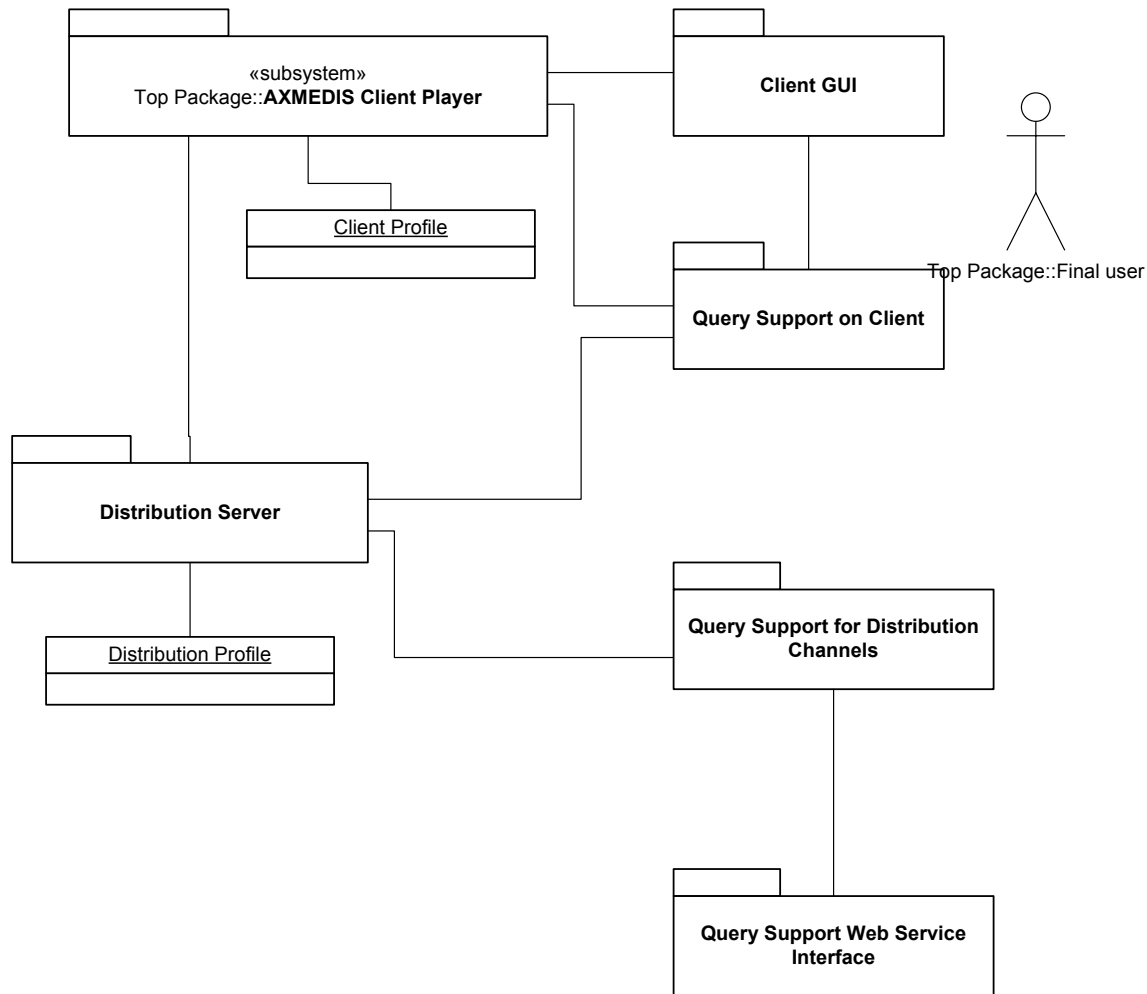
5 Query for production on Demand



The final user can enter a query in the AXMEDIS database using the Client GUI. The Query Support on the Client creates a query message in XML based on a simple query schema as defined in the next section.

1. This XML query request including the client profile is sent to the Distribution Server.
2. The Distribution Server adds his distribution profile and passes the query on to the Query Support for Distribution Channels.
3. The Query Support for Distribution Channels verifies and adjusts the query according to client and distribution profile. The simplified XML file is transferred in a AXMEDIS XML query as described in section 4.3.2 “Schema for an AXMEDIS query”. Therefore, some parameters are added and adjusted according to the client and the distribution profile. Then the query is sent to the Query Support Web Service Interface.
4. The Query Support Web Service Interface passes the request into the AXMEDIS database and/or AXEPT tool.
5. The query results are returned in an XML file according to the schema defined for AXMEDIS query results in section 4.3.3 “Schema for an AXMEDIS query result”.
6. The query results are passed on to the Query Support for Distribution Channels.
7. The Query Support for Distribution Channels passes the query results on to the Distribution Server.
8. The query results are returned to the Query Support for Clients and presented to the final user with the Client GUI.
9. The user selects a specific content using the Client GUI and the Query Support for Clients passes this selection on to the Distribution Server.

Afterwards the Distribution Server requests the selected content using the Programme and Publication Tools and returns the results to the client. The final user can now access the newly acquired content using the AXMEDIS client player.



5.1 Query Support for Distribution Channels (FHGIGD)

Distribution channel can interact with AXMEDIS query support via the Query support Web service interface that is a standard technology for sharing information in Internet by using a standard protocol with a standard definition language for service interface.

The Query Support for Distribution Channels gets a client queries from the Distribution Server which include a client and a distribution profile. The Query Support for Distribution Channels verifies and adjusts such a query according to client and distribution profile.

Since queries are received according to a simplified XML schema, they have to be transferred in an AXMEDIS XML query as described in section 4.3.2 “Schema for an AXMEDIS query”. Therefore, some parameters are added and adjusted according to the client and the distribution profile. Then the queries are sent to the Query Support Web Service Interface.

Query results are returned to the Query Support for Distribution Channels in a XML file according to the schema defined for AXMEDIS query results in section 4.3.3 “Schema for an AXMEDIS query result”.. and passed on to the Distribution Server.

The simple XML query language used by the final client is structured as a sequence of pairs of

- field name and
- value.



Simple Query Schema

The field name can refer to a field in the AXINFO, e.g to one of the following Dublin Core fields:

- **Title:** A name given to the resource.
- **Subject and Keywords:** The topic of the content of the resource.
- **Description:** An account of the content of the resource.
- **Resource Type:** The nature or genre of the content of the resource.
- **Date:** A date associated with an event in the life cycle of the resource.
- **Format:** The physical or digital manifestation of the resource.

Module Profile		
Query Support for Distribution Channels		
Executable or Library(Support)	Executable	
Single Thread or Multithread	Multithread	
Language of Development	Java	
Responsible Name	Peter Ebinger	
Responsible Partner	FHGIGD	
Status (proposed/approved)	Proposed	
Platforms supported	All platforms that support Java	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
Query Support for Clients	Query Support Web Service Interface	SSL
Distribution Server	Distribution Server	
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.
No user interface		
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
XML parser	Xerces Java 2	Apache Software License
XSL stylesheet processors in Java	Xalan Java 2	http://xml.apache.org/dist/LICENSE.txt

GSOAP	GSOAP 2.7	Gsoap public license http://gsoap2.sourceforge.net/

5.2 Query Support for Clients (FHGIGD)

The query support for clients is located on the distributor side. It is either implemented on the distribution server or it is integrated into the client application.

The final user can enter a query in the AXMEDIS database using the Client GUI. The Query Support for the Client creates a query message in XML based on the query schema defined in the section above. A client profile is added to such a XML query and then it is sent to the Distribution Server.

When query results are returned to the Query Support For Clients they are presented to the final user with the Client GUI. The user can select a specific content using the Client GUI and the Query Support For Clients passes this selection on to the Distribution Server.

Query Support for PC using Web Interface

In the following is as an example the implementation of the Query Support for PC using Web Interface described.

The query support for PC using Web Interface is integrated in the distribution server. The final user can enter a query in the AXMEDIS database on the web server of the distributor using the client browser. The Query Support for the Client on the distribution server creates a query message in XML based on the query schema defined in the section above. A client profile is added to such a XML query and then it is sent to the Distribution Server.

When query results are returned to the Query Support for Clients on the distribution server they are presented to the final user on a web page. The user can select a specific content by clicking on the respective link and the Query Support For Clients passes this selection on to the P&P Engine via the Distribution Server.

Module Profile		
Query Support for PC using Web Interface		
Executable or Library(Support)	Executable	
Single Thread or Multithread	Single Thread	
Language of Development	Java	
Responsible Name	Peter Ebinger	
Responsible Partner	FHGIGD	
Status (proposed/approved)	Proposed	
Platforms supported	All platforms that support Java	
Interfaces with other tools:	Name of the communicating tools	Communication model and format (protected or not, etc.)
Query Support for Distribution Channels	Distribution Server	
Distribution Server		
Client Browser		
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.
Web Interface		
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
XML parser	Xerces Java 2	Apache Software License
XSL stylesheet processors in Java	Xalan Java 2	http://xml.apache.org/dist/LICENSE.txt

5.3 Client Profile (FHGIGD)

The client profile stores information about the hardware and software capabilities of the client device. Since there is a high number and big variety of devices that are connected to the Internet, there is a corresponding need to deliver content that is tailored to the capabilities of different devices. As part of a framework for content adaptation and contextualization, a general purpose profile format is required that can describe the capabilities of a client and preferences of its user. Composite Capability/Preference Profiles (CC/PP) is designed to be such a format and is therefore used in AXMEDIS to describe the Client Profile.

The “Structure and Vocabularies 1.0” of CC/PP can be found at <http://www.w3.org/TR/CCPP-struct-vocab/>.

Three different categories of information exist:

- **Hardware capabilities** of the client device include the device class (e.g. PDA or mobile phone), the vendor, the device type, its processor, RAM, and general capabilities like graphics or sound capabilities.
- **Software capabilities** include the operating system of the client and its communication, encryption, compression, rendering (e.g. installed viewers or players) capabilities.
- **User preferences** are certain setting defined by the user or owner of the device. This allows the user to set certain parameters (capabilities) according to his needs.

E.g. due to the client settings defined by the user the profile has to be stored on the client device.

The CC/PP structure covers two main areas:

- CC/PP profile components
- CC/PP profile defaults

Profile Components

The general structure of a CC/PP client profile is a two-level tree: components and attributes, with provision for each component to reference an externally defined set of default attribute values.

A CC/PP profile contains *one or more components*, and each component contains *one or more attributes*. CC/PP profiles are constructed using RDF. The RDF data model represents CC/PP attributes as named properties linking a subject resource to an associated object resource or RDF literal value.

To describe client capabilities and preferences, the client being described is a resource whose features are described by labeled graph edges from that resource to corresponding object values. The graph edge labels

identify the client feature (CC/PP attribute) being described and the corresponding object values are the feature values.

RDF statement describing a client attribute:

[Client component resource] --attributeName--> (Attribute-value)

CC/PP attribute labels are represented by XML name values, which may include a namespace prefix. Attribute values may be of simple or structured data types.

Profile Defaults

Each component of a client profile may indicate a single separate resource that in turn indicates a subordinate collection of default attribute values. This collection of default values can be a separate RDF document that is named via a URI, or can appear in the same document as the client profile (though, in practice, there is probably little value in defaults in the same document). If an attribute in the collection of defaults is also present in the main part of the client profile, the non-default value takes precedence. The intent is that a hardware vendor or system supplier may provide default values that are common to a number of systems in a place easily accessible to an origin server, and then use the client profile to specify variations from the common profile. The owner of the product or system operator may be able to add or change options, such as additional memory, that add new capabilities or change the values of some original capabilities.

Example CC/PP profile for a mobile phone:

(Ericsson T68/R1 or SonyEricsson T68/R1, <http://wap.sonyericssonmobile.com/UAProf/T68R1.xml>)

```
<?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.wapforum.org/UAPROF/ccppschemata-20000405#">
  <rdf:Description ID="Profile">
    <prf:component>
      <rdf:Description ID="HardwarePlatform">
        <prf:ScreenSize>101x80</prf:ScreenSize>
        <prf:Model>T68R1</prf:Model>
        <prf:InputCharSet>
          <rdf:Bag>
            <rdf:li>ISO-8859-1</rdf:li>
            <rdf:li>US-ASCII</rdf:li>
            <rdf:li>UTF-8</rdf:li>
            <rdf:li>ISO-10646-UCS-2</rdf:li>
          </rdf:Bag>
        </prf:InputCharSet>
        <prf:ScreenSizeChar>15x6</prf:ScreenSizeChar>
        <prf:BitsPerPixel>8</prf:BitsPerPixel>
        <prf:ColorCapable>Yes</prf:ColorCapable>
        <prf:TextInputCapable>Yes</prf:TextInputCapable>
        <prf:ImageCapable>Yes</prf:ImageCapable>
        <prf:Keyboard>PhoneKeypad</prf:Keyboard>
        <prf:NumberOfSoftKeys>0</prf:NumberOfSoftKeys>
        <prf:Vendor>Ericsson Mobile Communications AB</prf:Vendor>
        <prf:OutputCharSet>
          <rdf:Bag>
            <rdf:li>ISO-8859-1</rdf:li>
            <rdf:li>US-ASCII</rdf:li>
            <rdf:li>UTF-8</rdf:li>
            <rdf:li>ISO-10646-UCS-2</rdf:li>
          </rdf:Bag>
        </prf:OutputCharSet>
        <prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
        <prf:StandardFontProportional>Yes</prf:StandardFontProportional>
        <prf:PixelsAspectRatio>1x1</prf:PixelsAspectRatio>
      </rdf:Description>
```

```

</prf:component>
<prf:component>
  <rdf:Description ID="SoftwarePlatform">
    <prf:AcceptDownloadableSoftware>No</prf:AcceptDownloadableSoftware>
  </rdf:Description>
</prf:component>
<prf:component>
  <rdf:Description ID="NetworkCharacteristics">
    <prf:SecuritySupport>WTLSclass1/2/3/signText</prf:SecuritySupport>
    <prf:SupportedBearers>
      <rdf:Bag>
        <rdf:li>TwoWaySMS</rdf:li>
        <rdf:li>CSD</rdf:li>
        <rdf:li>GPRS</rdf:li>
      </rdf:Bag>
    </prf:SupportedBearers>
  </rdf:Description>
</prf:component>
<prf:component>
  <rdf:Description ID="BrowserUA">
    <prf:BrowserName>Ericsson</prf:BrowserName>
    <prf:CcppAccept>
      <rdf:Bag>
        <rdf:li>application/vnd.wap.wmlc</rdf:li>
        <rdf:li>application/vnd.wap.wbxml</rdf:li>
        <rdf:li>application/vnd.wap.wmlscriptc</rdf:li>
        <rdf:li>application/vnd.wap.multipart.mixed</rdf:li>
        <rdf:li>text/x-vCard</rdf:li>
        <rdf:li>text/x-vCalendar</rdf:li>
        <rdf:li>text/x-vMel</rdf:li>
        <rdf:li>text/x-eMelody</rdf:li>
        <rdf:li>image/vnd.wap.wbmp</rdf:li>
        <rdf:li>image/gif</rdf:li>
        <rdf:li>image/jpeg</rdf:li>
        <rdf:li>application/vnd.wap.wtls-ca-certificate</rdf:li>
        <rdf:li>application/vnd.wap.sic</rdf:li>
        <rdf:li>application/vnd.wap.slc</rdf:li>
        <rdf:li>application/vnd.wap.coc</rdf:li>
        <rdf:li>application/vnd.wap.sia</rdf:li>
        <rdf:li>application/vnd.eri.thm</rdf:li>
      </rdf:Bag>
    </prf:CcppAccept>
    <prf:CcppAccept-Charset>
      <rdf:Bag>
        <rdf:li>US-ASCII</rdf:li>
        <rdf:li>ISO-8859-1</rdf:li>
        <rdf:li>UTF-8</rdf:li>
        <rdf:li>ISO-10646-UCS-2</rdf:li>
      </rdf:Bag>
    </prf:CcppAccept-Charset>
    <prf:CcppAccept-Encoding>
      <rdf:Bag>
        <rdf:li>base64</rdf:li>
      </rdf:Bag>
    </prf:CcppAccept-Encoding>
    <prf:FramesCapable>No</prf:FramesCapable>
    <prf:TablesCapable>Yes</prf:TablesCapable>
  </rdf:Description>
</prf:component>
<prf:component>
  <rdf:Description ID="WapCharacteristics">
    <prf:WapDeviceClass>C</prf:WapDeviceClass>
    <prf:WapPushMsgSize>3000</prf:WapPushMsgSize>
    <prf:WapVersion>1.2.1/June 2000</prf:WapVersion>
    <prf:WmlVersion>
      <rdf:Bag>

```



```

        <rdf:li>1.2.1/June 2000</rdf:li>
        <rdf:li>1.1</rdf:li>
      </rdf:Bag>
    </prf:WmlVersion>
    <prf:WmlDeckSize>3000</prf:WmlDeckSize>
    <prf:WmlScriptVersion>
      <rdf:Bag>
        <rdf:li>1.2.1/June 2000</rdf:li>
        <rdf:li>1.1</rdf:li>
      </rdf:Bag>
    </prf:WmlScriptVersion>
    <prf:WmlScriptLibraries>
      <rdf:Bag>
        <rdf:li>Lang</rdf:li>
        <rdf:li>Float</rdf:li>
        <rdf:li>String</rdf:li>
        <rdf:li>URL</rdf:li>
        <rdf:li>WMLBrowser</rdf:li>
        <rdf:li>Dialogs</rdf:li>
      </rdf:Bag>
    </prf:WmlScriptLibraries>
    <prf:WtailLibraries>
      <rdf:Bag>
        <rdf:li>WTA.Public.makeCall</rdf:li>
        <rdf:li>WTA.Public.sendDTMF</rdf:li>
        <rdf:li>WTA.Public.addPBEntry</rdf:li>
      </rdf:Bag>
    </prf:WtailLibraries>
  </rdf:Description>
</prf:component>
</rdf:Description>
</RDF>

```

5.4 Distribution Profile (FHGIGD)

The distribution profile stores general information about the distribution channel, e.g. bandwidth. Further information has to be available, including

- a default client profil: This is important if a client profile is not accessible, e.g. if content is bought for a specific device different from the requesting device.
- further parameters: Distributor specific settings, e.g. whether only the local AXMEDIS database should be searched and other distributor related preferences have to be stored in the distribution profile.

