Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE9.1.5
Integrated Prototype of CMS integration and feedback

Version: 1.1
Date: 22/07/2008 14:25:00
Responsible: EXITECH (ff@exitech.it)

Project Number: IST-2-511299
Project Title: AXMEDIS
Deliverable Type: Public
Visible to User Groups: Yes
Visible to Affiliated: Yes
Visible to Public: Yes

Deliverable Number: DE9.1.5
Contractual Date of Delivery: see annex I
Work-Package contributing to the Deliverable: WP 9.1.2 and sun WPs
Nature of the Deliverable: Prototype and Report
Author(s): EXITECH, DSI, SEJER, AFI, ANSC, ILABS, EUTELSAT, TISCALI, XIM, TEO, ELION, VRS, SDAE

Abstract:
This deliverable is related to the integrated prototype of WP9.1.2 that has been finalized apart from some residual optimization, some maintenance and future change not foreseen at the moment by producing a real integrated prototype cooperating with CAMART, AXDB and therefore also with AXCS.
This deliverable is an update of DE 9.1.4.
DE9.1.5 – Integrated Prototype of CMS integration and feedback

**Keyword List:**
Administrative data, AXCS, AXDB, CAMART

**Document responsible**
name: Fabrizio Fioravanti
Email address: ff@exitech.it
Affiliation acronym: EXITECH
# Table of Contents

1 EXECUTIVE SUMMARY AND REPORT SCOPE ................................................................. 6

2 INTRODUCTION ......................................................................................................... 7

3 USER REQUIREMENTS ............................................................................................. 8

3.1 ACCOUNTING MANAGER AND REPORTING TOOLS ............................................. 8

3.2 ADMINISTRATIVE INFORMATION INTEGRATOR .................................................... 9

4 USE CASES .............................................................................................................. 10

4.1 AXMEDIS REPORTING AND STATISTICS WEB SERVICE (DSI, EXITECH) ............. 10

4.1.1 Object usage reporting for accounting purposes (EXITECH, DSI) ..................... 10

4.1.2 Object usage reporting for statistics purposes (EXITECH, DSI) ....................... 10

4.2 ACCOUNTING MANAGER AND REPORTING TOOL (EXITECH) ......................... 10

4.2.1 List of all operations performed on an object ..................................................... 10

4.2.2 List of all operations performed by a user .......................................................... 11

4.2.3 Usage report about an object ............................................................................. 11

4.2.4 Usage report about a distributor ....................................................................... 11

4.2.5 Usage report about a provider .......................................................................... 12

4.2.6 List objects for which an administrative account can be requested ................ 12

4.2.7 Listing AXMEDIS clients of a distributor/channel ............................................ 12

4.2.8 Listing distributors ........................................................................................... 13

4.3 GENERAL USE CASES AND SCENARIOS ............................................................ 13

4.3.1 Mapping of Administrative Information ............................................................. 13

4.3.2 Distributor wants Administrative Information .................................................. 15

4.3.3 Creator or Collecting Society wants Administrative Information ..................... 15

4.3.4 Administrative Information Integrator ............................................................... 16

4.3.5 Administrative Information Integrator as seen from Collecting society perspective ............................................................................................................. 17

4.3.6 Relationships between CAMART and AII ......................................................... 20

5 SYSTEM SPECIFICATION ......................................................................................... 21

5.1 GENERAL ARCHITECTURE .................................................................................. 21

5.2 MODULE OR EXECUTABLE TOOL CORE ACCOUNTING MANAGER AND REPORTING TOOLS (CAMART) ................................................................. 21

5.2.1 General Description of the CAMART Module .................................................... 21

5.2.2 Core Accounting Manager and Reporting Tools interface toward the user ...... 22

5.2.3 CAMART interface with AXCS ......................................................................... 23

5.2.4 CAMART Module Design in terms of Classes ................................................... 23

5.2.5 CAMART Prototype description ........................................................................ 25

5.2.6 Technical and Installation information ............................................................... 26

5.2.7 Integration and compilation issues ..................................................................... 27

5.2.8 Configuration Parameters ................................................................................ 27

5.2.9 Errors reported and that may occur ................................................................. 28

5.3 ADMINISTRATIVE INFORMATION INTEGRATOR (AII) ..................................... 28

5.3.1 General Description of the Module .................................................................... 28

5.3.2 Administrative Information Integrator in polling mode .................................... 28

5.3.3 Administrative Information Integrator in push mode ..................................... 29

5.3.4 Administrative Information Integrator and CAMART integration ................... 32

5.3.5 Module Design in terms of Classes ................................................................... 34

5.3.6 AII Prototype User Interface description ......................................................... 34

5.3.7 AII Web Service Interface ................................................................................ 34

5.3.8 Technical and Installation information ............................................................... 41

5.3.9 Draft User Manual ............................................................................................ 41

5.3.10 Integration and compilation issues .................................................................... 41
6 ANALYSIS OF CMS OF PARTNERS (EXITECH, ALL) ........................................................................................................ 56

6.1 CMS DATA RETRIEVABLE FROM AXCS (DSI) ................................................................................................. 56
6.2 AFI CMS RELATED DATA (AFI) ........................................................................................................................... 57
   6.2.1 Role............................................................................................................................................................. 57
   6.2.2 Minimal set of data needed ......................................................................................................................... 57
   6.2.3 File format for data input .......................................................................................................................... 58
6.3 EUTELSAT CMS RELATED DATA (EUTELSAT) ...................................................................................................... 58
   6.3.1 Role............................................................................................................................................................. 58
   6.3.2 Minimal set of data needed ......................................................................................................................... 58
   6.3.3 File format for data input .......................................................................................................................... 59
6.4 ILABS CMS RELATED DATA (ILABS) .................................................................................................................... 59
   6.4.1 Role............................................................................................................................................................. 59
   6.4.2 Minimal set of data needed ......................................................................................................................... 59
   6.4.3 File format for data input .......................................................................................................................... 60
6.5 TISCALI CMS RELATED DATA (TISCALI) .............................................................................................................. 62
   6.5.1 Role............................................................................................................................................................. 62
   6.5.2 Minimal set of data needed ......................................................................................................................... 62
   6.5.3 File format for data input .......................................................................................................................... 63
6.6 XIM CMS RELATED DATA (XIM) ........................................................................................................................... 64
   6.6.1 Role............................................................................................................................................................. 64
   6.6.2 Role............................................................................................................................................................. 64
   6.6.3 Minimal set of data needed ......................................................................................................................... 64
   6.6.4 File format for data input .......................................................................................................................... 64
6.7 SEJER CMS RELATED DATA (SEJER) ..................................................................................................................... 64
   6.7.1 Role............................................................................................................................................................. 64
   6.7.2 Minimal set of data needed ......................................................................................................................... 64
   6.7.3 File format for data input .......................................................................................................................... 64
6.8 ANSC CMS RELATED DATA (ANSC) ....................................................................................................................... 65
   6.8.1 Role............................................................................................................................................................. 65
   6.8.2 Minimal set of data needed ......................................................................................................................... 65
   6.8.3 File format for data input .......................................................................................................................... 65
6.9 HP CMS RELATED DATA (HP) .............................................................................................................................. 66
6.10 BBC CMS RELATED DATA (BBC) ......................................................................................................................... 66
6.11 TEO CMS RELATED DATA (TEO) .......................................................................................................................... 66
   6.11.1 Role............................................................................................................................................................. 66
   6.11.2 Minimal set of data needed ......................................................................................................................... 66
   6.11.3 File format for data input .......................................................................................................................... 66
6.12 TI CMS RELATED DATA (TI) ............................................................................................................................... 67
6.13 SDAE CMS RELATED DATA (SDAE) .................................................................................................................... 67
   6.13.1 Role............................................................................................................................................................. 67
   6.13.2 Minimal set of data needed ......................................................................................................................... 68
   6.13.3 File format for data input .......................................................................................................................... 68
6.14 ELION CMS RELATED DATA (ELION) .................................................................................................................. 69
   6.14.1 Role............................................................................................................................................................. 69
   6.14.2 Minimal set of data needed ......................................................................................................................... 69
   6.14.3 File format for data input .......................................................................................................................... 70
6.15 VRS CMS RELATED DATA (VRS) .......................................................................................................................... 71
   6.15.1 Role............................................................................................................................................................. 71
   6.15.2 Minimal set of data needed ......................................................................................................................... 71
   6.15.3 File format for data input .......................................................................................................................... 71
DE9.1.5 – Integrated Prototype of CMS integration and feedback

7 ADMINISTRATIVE INFORMATION INTEGRATOR FORMAT DETAILED SPECIFICATION ........73

8 TABLE DESCRIPTION FOR DATABASE AXDB (CAMART AND AII RELATED TABLES) AND XML SCHEMAS ................................................................................................................................. 76

8.1 FORMAL DESCRIPTION OF FORMAT AII RECORD .................................................................................................................. 80
8.2 FORMAL DESCRIPTION OF FORMAT FOR STATISTICS RECORD .................................................................................................... 82

9 DEMOISTRATOR FACT SHEET .................................................................................................................................................. 85

9.1 MAIN PURPOSES OF THE DEMONSTRATOR ............................................................................................................................................. 85
9.2 REVIEW OF THE ARCHITECTURE INTEGRATION WITH AXMEDIS .......................................................................................... 86
9.3 DESCRIPTION OF THE EFFECTIVE INSTALLATION .................................................................................................................. 87
9.4 AXMEDIS TOOLS ........................................................................................................................................................................ 87
9.5 TARGET MARKET ................................................................................................................................................................. 87
9.6 DESCRIPTION OF THE BUSINESS MODEL ...................................................................................................................................... 87
9.7 DESCRIPTION OF CONTENT .......................................................................................................................................................... 87
9.8 FINAL USERS/CLIENTS .......................................................................................................................................................... 87
9.9 PARTNERS INVOLVED AND ROLES ........................................................................................................................................ 87

10 USAGE OF CAMART/CAMARTSTATS ON WP9 AND WP12 DEMONSTRATORS ........................................................................... 87

10.1 BBC, EUTELSAT, TELECOM .................................................................................................................................................. 87
10.2 ILABS ...................................................................................................................................................................................... 88
10.3 TISCALI .................................................................................................................................................................................. 88

11 BIBLIOGRAPHY ............................................................................................................................................................................. 89
1 Executive Summary and Report Scope

This deliverable is the accompanying document for the Prototype having the name common to the deliverable. This deliverable contains all the details related to the prototype from the user requirements and use case to the specification and manuals plus a forecast for validation phase. This deliverable is an update of DE 9.1.4 and therefore contains all the information of 9.1.4 updated and revised where needed.
2 Introduction
This deliverable is related to the mock up of WP9.1.2 that has been finalized apart from some residual optimization, some maintenance and future change not foreseen at the moment by producing a real integrated prototype cooperating with CAMART, AXDB and therefore also with AXCS.

This integrated prototype is able to perform the following tasks:
- Getting logs coming from AXCS web service
- Organizing such logs in the internal database of the factory
- Generating internal XML format in polling mode
- Generating the XML format provided by all of the partners according to the specification, and publishing at a predefined time frequency the resulting XML in an ftp directory
- Generating also in polling mode an XML format that is transformed according to the profiled XSLT.
- Generating top-bottom ten on demand for statistics.
- Authentication
- Availability of a web service for gathering statistics instead of the GUI only
- Availability of a Web service to remotely control AII completely in order to be able to automate log collection
3 User Requirements
The requirements that are reported here are taken from DE 2.1.1.2.1.

3.1 Accounting Manager and Reporting Tools
Accounting managing and reporting tool should collect information from AXMEDIS Certifier and Supervisor and Administrative Information Integrator and log these data in Account/public log database.

11.5.1) DRM modelling, modelling licensing. Study Rights information models
11.5.2) Modelling database for licensing and transaction tracking
11.5.3) Communicating with the AXMEDIS Certifier and Supervisor to get specific information related to the transactions performed on the objects of a given content provider or aggregator.
11.5.4) Storing into the AXMEDIS database the transactions, matching who has done the action on what;
11.5.5) Listing clients of the provider, with the history of their transactions, etc.
11.5.6) Listing objects for which the user has authorization
11.5.7) Listing all distributors and higher level
11.5.8) Generate any kind of report among those for which data are present and accessible to the user who requested the report.
11.5.9) Report and statistic analysis data generation in order to empower content owner and distributors to improve, and possibly automate the process of building statistically based promotions and personalised offer at least as detailed hereafter:
   - Generate a list of the most used/acquired contents
   - Generate a list of the most used/acquired contents per user
   - Generate a list of the most used/acquired contents per distributor
   - Generate a list of the most used/acquired contents per category
   - Generate a list of the most used/acquired contents per user and category
   - Generate a list of the most used/acquired contents per distributor and category
   - Generate a list of the most used/acquired contents per distributor, user and category
   - Generate a list of the least used/acquired contents
   - Generate a list of the least used/acquired contents per user
   - Generate a list of the least used/acquired contents per distributor
   - Generate a list of the least used/acquired contents per category
   - Generate a list of the least used/acquired contents per user and category
   - Generate a list of the least used/acquired contents per distributor and category
   - Generate a list of the least used/acquired contents per distributor, user and category
11.5.10) Statistic analysis of the content usage, very useful for tuning the service and the structure of the ready to use proposed objects. Among these statistics it would be advisable to have the following ones: (should be addressed by external tools since statistical data are not stored in AXDB)
   - Content sorted by usage rate
   - Content used sorted by category
   - Content used sorted by category and usage rate
   - Top 10 used Content sorted by category
   - Top 20 used Content sorted by category
   - Bottom 10 used Content sorted by category
   - Bottom 20 used Content sorted by category
11.5.11) Compute statistics about the access, utilisation, distribution etc. of the AXMEDIS objects based on the event reports previously generated. Among these statistics it would be advisable to have the following ones: (should be addressed by external tools since statistical data are not stored in AXDB)
   - Content sorted by access rate
   - Content accessed sorted by category
   - Content accessed sorted by category and access rate
   - Top 10 accessed Content sorted by category
   - Top 20 accessed Content sorted by category
   - Bottom 10 accessed Content sorted by category
3.2 Administrative Information Integrator

This is a set of tools for making available the administrative information received from the AXMEDIS certifier and supervisor and collected into the AXMEDIS database (managed by the Accounting Managing and Reporting tool) into the database of the Content Providers in their administrative form. For example, to bring administrative information into XAURA, HP CMS, XX CMS, etc. For this purpose, in WP 9.1 several Administrative Information Integrators will be realised. The idea is to find a common basis among them and to customise the application according to the needs and protocols to interact with the different CMSs.

The Administrative Information Integrator shall:

3.2.1) interface with different CMS technology;
3.2.2) store administrative information into the Content Provider database.
3.2.3) communicate with the AXDB to get administrative information related to a specific Content Provider.
3.2.4) guarantee privacy of sensitive data via protection mechanisms
4 Use Cases

In this section the use cases reported in DE 2.1.2.2.2 are listed and at the end a general use case with the last addition in terms of functionalities.

4.1 AXMEDIS Reporting and Statistics Web Service (DSI, EXITECH)

4.1.1 Object usage reporting for accounting purposes (EXITECH, DSI)

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>Object usage reporting for accounting purposes</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants data about object usage for accounting purposes</td>
</tr>
<tr>
<td>Actors</td>
<td>Business users interested in accounting activities (e.g. Collecting Societies, Distributors, etc.), CAMART, AXCS</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
| Steps | 1 A business users interested in accounting activities sends a report request to the CAMART providing its own id, a timestamp and some other criteria  
2 The service checks and validates the received data  
3 The service collects information related to the requestor pertinent object usage from AXCS using the Reporting Web Service and sends it back to the requester |
| Post-conditions | The Actor has the report |
| Variations | None |
| Asynchronous actions | None |
| Design suggestions | None |
| Issues | CAMART is the only tool dealing with Reporting Web Service |

4.1.2 Object usage reporting for statistics purposes (EXITECH, DSI)

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>Object usage reporting for statistics purposes</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants data about object usage for statistics purposes</td>
</tr>
<tr>
<td>Actors</td>
<td>Business users interested in statistic activities (e.g. Collecting Societies, etc.), CAMART, AXCS</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
| Steps | 1 A business users interested in statistic activities sends a report request to the CAMART providing its own id, a timestamp and some other criteria  
2 The service checks and validates the received data  
3 The service collects anonymous information about object usage from AXCS using the Statistics Web Service and sends it back to the requester |
| Post-conditions | The Actor has the report |
| Variations | None |
| Asynchronous actions | None |
| Design suggestions | None |
| Issues | CAMART is the only tool dealing with Statistics Web Service |

4.2 Accounting Manager and Reporting Tool (EXITECH)

4.2.1 List of all operations performed on an object

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>List of all operations performed on an object</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants to know all operations performed on an object</td>
</tr>
<tr>
<td>Actors</td>
<td>Content provider, distributor, collecting society</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
### 4.2.2 List of all operations performed by a user

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>List of all operations performed by a user</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants to know all operations performed by a user</td>
</tr>
<tr>
<td>Actors</td>
<td>Distributor</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
| Steps | 1 An Actor sends the request for having the list of all operations performed by an user: the request contain the User ID, the Actor ID and the code of the operation to be performed, and time window related to the period  
 2 The service checks and validates the data received  
 3 The service collects and sends back Action-Logs related to the user |
| Post-conditions | The Distributor has the list |
| Variations | None |
| Asynchronous actions | None |
| Design suggestions | None |
| Issues | None |

### 4.2.3 Usage report about an object

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>Usage report about an object</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants to know usage statistic about an object</td>
</tr>
<tr>
<td>Actors</td>
<td>Distributor, Collecting society, content owner, content provider</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
| Steps | 1 An Actor sends the request for having the statistic about an object usage: the request contain the Object ID, the Actor ID and the code of the operation to be performed, and time window related to the period  
 2 The service checks and validates the data received  
 3 The service collects and sends back information related to the object usage |
| Post-conditions | The Actor obtains the statistic |
| Variations | None |
| Asynchronous actions | None |
| Design suggestions | None |
| Issues | None |

### 4.2.4 Usage report about a distributor

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>Usage report about a distributor</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants to know usage statistic about a distributor</td>
</tr>
<tr>
<td>Actors</td>
<td>Distributor, Collecting society, content owner, content provider</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
| Steps | 1 The Actor sends a request to the service: the request contains the Distributor ID, the Actor ID and the operation to be performed, and time window related to the period  
 2 The service checks and validates the data received  
 3 The service collects and sends back information related to the object usage |
| Post-conditions | The Actor obtains the statistic |
| Variations | None |
| Asynchronous actions | None |
| Design suggestions | None |
| Issues | None |
### 4.2.5 Usage report about a provider

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>Usage report about a provider</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants to know usage statistic about a provider</td>
</tr>
<tr>
<td>Actors</td>
<td>Distributor, Collecting society, content owner, content provider</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
| Steps | 1. The Actor sends a request to the service: the request contains the Provider ID, the Actor ID and the operation to be performed, and time window related to the period  
2. The service checks and validates the data received  
3. The service collects and sends back information related to the provider |
| Post-conditions | The Actor obtains the statistic |
| Variations | None |
| Asynchronous actions | None |
| Design suggestions | None |
| Issues | None |

### 4.2.6 List objects for which an administrative account can be requested

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>List objects which an administrative account can be requested</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants to obtain and consult the list of objects for which an administrative account can be requested, e.g. all the objects for which the Actor has the eligibility to obtain an administrative report</td>
</tr>
<tr>
<td>Actors</td>
<td>Content creators, distributors, collecting society, content providers</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
</tbody>
</table>
| Steps | 1. An Actor requests the list of objects for which an administrative account can be requested  
2. The reporting tool searches in the corresponding database all the AXMEDIS objects for which the actor is eligible to ask a report.  
3. The reporting tool lists all the results. |
| Post-conditions | None |
| Variations | None |
| Asynchronous actions | None |
| Design suggestions | None |
| Issues | None |

### 4.2.7 Listing AXMEDIS clients of a distributor/channel

<table>
<thead>
<tr>
<th>UCId</th>
<th>UC13.2.5.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>Listing all AXMEDIS clients</td>
</tr>
<tr>
<td>Description</td>
<td>An Actor wants to consult the AXMEDIS clients list that has been connected to the distributor or on a channel.</td>
</tr>
<tr>
<td>Actors</td>
<td>Distributors</td>
</tr>
<tr>
<td>Assumptions</td>
<td>None</td>
</tr>
<tr>
<td>Steps</td>
<td>1. The Actor asks for the list of AXMEDIS clients that have been connected to a distributor of a channel</td>
</tr>
</tbody>
</table>
2. The reporting tool searches in the corresponding database all the AXMEDIS clients satisfying point 1.
3. The reporting tool lists all the results.

Post-conditions: None
Variations: None
Asynchronous actions: None
Design suggestions: None
Issues: None

4.2.8 Listing distributors

UCId: UC13.2.5.8

Use case: Listing all distributors of AXMEDIS objects, those that have redistributed its objects.

Description: An Actor wants to consult the distributors of AXMEDIS objects.

Actors: Content creators, Distributors, End users, Content Providers.

Assumptions: None

Steps:
1. An actor asks for a list of distributors of AXMEDIS objects
2. The reporting tool searches in the corresponding database all the distributors of AXMEDIS objects
3. The reporting tool lists all the results.

Post-conditions: None
Variations: None
Asynchronous actions: None
Design suggestions: None
Issues: None

4.3 General Use Cases and scenarios

In this section the general use case from several different documents have been collected:

- Mapping of Administrative Information
- Distributor wants Administrative Information
- Creator or Collecting Society wants Administrative Information
- Administrative Information Integrator
- Administrative Information Integrator as seen from Collecting society perspective
- Relationships between CAMART and AII

4.3.1 Mapping of Administrative Information
1. End User requests to perform an action on an AXMEDIS Protected Object
2. AXMEDIS Player asks PMS to perform an Action (assuming client has been already certified)
3. PMS checks in the Licence DB if the Action is allowed (assuming OK)
4. PMS sends AXCS the action performed
5. AXCS returns the key to access the content (if necessary)
6. PMS grants the access to the content and possibly returns the key to the AXMEDIS Player
7. CAMART retrieves from AXCS the actions performed by all the End Users on objects distributed by the distributor
8. CAMART stores the transactions into the AXDB
9. Administrative Information Integrator gets transactions performed from the AXDB
10. Administrative information are mapped into the Distributor CMS
4.3.2 Distributor wants Administrative Information

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

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

4.3.4 Administrative Information Integrator
1. A Distributor wants to recover information on actions performed on the objects he has rights.
2. CAMART queries the reporting web service to obtain the Action-Logs
3. AXMEDIS Statistic or reporting tools queries AXCS
4. AXCS extracts the required Action-Logs and communicate them to the reporting tool
5. Accounting report is generated.
6. Accounting report is passed to the Administrative Information Integrator
7. Data are loaded into the Distributor CMS

4.3.5 Administrative Information Integrator as seen from Collecting society perspective

The following three scenarios describe how AXMEDIS provides a service to the Collecting Societies, supporting them in gathering reporting information on the use of protected objects so to enhance the management, ease the administration administration and enforce the rights they are vested in or represent.

Collecting societies administer a wide range of rights on behalf of copyright owners for a wide range of uses and users. They collect and distribute to right owners royalty income and equitable remuneration in relation to the exercise of these rights. In addition to these core functions, there are many other functions carried out by all or many Collecting societies such as enforcement, monitoring and auditing activities, particularly important in view of the increasingly uses of copyrighted content in the AXMEDIS context.

It has to be underline that these scenarios only refer to the use of music whose exploitation rights are granted to the original publisher and the producer. Their content, once governed as an AXMEDIS object, are ready to be exploited within the AXEPTOOL and distributed accordingly to the DRM and license terms provided. All actions (uses) or events performed on these AXMEDIS objects are recorded in the AXCS of each user and then reported, along with other relevant data, to the super AXCS. The super AXCS tool will then interact with the AXMEDIS reporting tool and with the Administrative Information Integrator that will respectively report relevant information into the right owners’ database and into the database of the entitled collecting societies. It has to be underline that the link between AXMEDIS tools and the collecting societies should be implemented by taking into account new tools and network developed by collecting societies themselves such as the FastTrack project. This project aims at realizing a global interconnected network of databases on
musical and audiovisual works, rights owners, contracts and data on sound recording to support diary operations of the societies involved such as identification of works and distribution of royalties

Independently of the ways and methodologies rights are granted to users (e.g. compulsory license, individual license) the Administrative Information integrator tool should provide Collecting societies with data needed to check, verify and monitor the use of the AXMEDIS objects in conformity with the rights granted by the relevant license and with information necessary to identify right owners including identification standard codes already developed (such as the ISRC) as well as those under development. An AXMEDIS object will involve a multiplicity of rights owners (there could be many just in one musical work) and many different collecting societies (such as public performance rights societies, mechanical rights societies, producers rights societies, performers rights societies etc ). Its multi distribution channels will allow multiple reproductions, transmissions and retransmissions until it reaches the end user/consumer. Due to this issue and to the complexity of the different rules that govern the collection and distribution of royalties for the exploitation of multimedia contents and compounded objects in the digital environment, the control of the correct use of the rights granted and the consequent collection of the royalties due become a complex task and one of the main concern of right holders community.
### 4.3.6 Relationships between CAMART and AII

1. CAMART contact AXCS web service for having fresh logs
2. AXCS give back to CAMART the Logs
3. CAMART store logs on AXDB
4. Accountant configure AII to put export logs toward CMS
5. AII get logs from AXDB
6. AII export logs to CMS
7. Accountant can optionally query directly the CAMART for logs filtered by AXOID, AXUID, timestamp etc
8. In that case CAMART will get the log from AXDB
9. CAMART will generate a raw XML report of Logs
5 System Specification
In this section an update of what stated on the basis of the addition and improvement realized in the prototype in DE3-1-2-2-15 is reported.

5.1 General Architecture
In the following diagram the relationship among users and tools of the AXMEDIS platform is reported together with the media that can be used to access the information (Web Page or Web service).

5.2 Module or Executable Tool Core Accounting Manager and Reporting Tools (CAMART)

5.2.1 General Description of the CAMART Module
The role of Core Accounting manager and Reporting Tool (CAMART) is strictly bound with database for logs (provided by AXCS) since it has to collect information regarding the B2B activities and B2C actions. AXCS will not store forever its logs and therefore it is necessary for CAMART to gather time by time such logs and store locally in the AXMEDIS database. Such information will be collected on scheduled time interval and CAMART will act as a client of the AXCS Reporting Web Service.
AXMEDIS system is scalable and therefore we have to deal with the fact that some installation can have AXDB, AXCS and other supporting tools on different machines, while others can be less distributed due to a lesser need for speed or storage capacity. The core accounting manager is a sort of Client side of the bridge between the AXDB and the AXCS databases in order to allow AXCS to be independent by the database. The server side in the AXCS is the Web Service: AXMEDIS Reporting Web Service. The CAMART can be interpreted as a part of the AXMEDIS Database Interface, since is the part of the system that allows writing data related to Action-Logs into the AXMEDIS DB.
5.2.2 Core Accounting Manager and Reporting Tools interface toward the user

The actor working on the CAMART user interface can be any administrative and management user that has interest in making queries and browsing the information related to the usage of AXMEDIS objects. For example:

- A Distributor could be interested in seeing the list of Action Logs related to a given second distributor, integrator, etc.
- An Integrator could be interested in seeing the list of Action Logs related to a given AXMEDIS Object, etc.
- A Distributor could be interested in seeing the list of second level distributors that have exploited some specific AXMEDIS object, etc.
- A Distributor could be interested to see how many transactions have been registered on its AXMEDIS objects in the last two months.

These examples can be used by the Actor (account manager) to extract the information and move it into the Administrative Database of the CMS by means of the AXMEDIS tool called Administrative Information Integrator.

The general role of CAMART with respect to the end user that uses it is to provide a web interface for making reporting directly onto their systems. Reporting queries are always executed on the AXMEDIS Database Interface for collecting information that are contained in the local database that in turn had been gathered from the AXCS. In this way the actor sees only the data for which is authorized.

The user interface, is web based in order to cover the needs of interoperability, usability and maintainability that are expected from AXMEDIS in the large sense. The format that is provided to the user is generated according to the XML schema reported below in graphical shape:
5.2.3 CAMAR T interface with AXCS
CAMAR T needs to get fresh logs from AXCS and therefore it is a client of the web service interface that AXCS will expose to have access to log information. In this section it is defined the interface in order to specify in terms of WSDL this communication protocol that will be a synchronous polling from CAMAR T to AXCS. AXCS will have to implement two web services and CAMAR T will be a client for them, reading at predefined time interval the logs from AXCS and storing them in the local AXDB related to Logs for the reporting part and reading on demand for the statistical part. The reporting process is completely automatic and is managed by a CAMAR T daemon that will periodically extract from AXCS all the logs for which the user is entitled and store them in the AXDB.

The web services implemented by AXCS/SuperAXCS are described in details in DE 3.1.2.2.13 and therefore no need to explain them in this document.

5.2.4 CAMAR T Module Design in terms of Classes
A detailed view of the module in terms of classes is reported in the following pictures:
5.2.5 CAMART Prototype description

This module is comprised of a daemon that collects info from AXCS and stores logs locally on AXDB for future reuse from AII.

The interface of this daemon is very simple and is reported below:
After the authentication we have the daemon interface

The prototype now automatically starts when the service starts and can be stopped on demand.

This thread runs in background and collects locally the remote logs provided by AXCS, showing also information on the logs inserted in the last poll with the amount of logs that were already present.

5.2.6 Technical and Installation information
To deploy the Camart Service, a PC with Java2 1.5 Runtime environment and Apache Tomcat 5.5.20 is required. The service will be distributed as a WAR file to be deployed in %TOMCAT_ROOT%/webapps.
5.2.7 Integration and compilation issues
Since Java and Apache Tomcat technologies, combined with W3C standard protocol (HTTP) and mark-up language (XHTML), were used, there is not any known issue related to interoperability and integration in different context.

5.2.8 Configuration Parameters

<table>
<thead>
<tr>
<th>Config parameter</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>axdbUrl: URI of the axdb for the jdbc</td>
<td>jdbc:mysql://localhost/axdbv4?jdbcCompliantTrimination=false</td>
</tr>
<tr>
<td>axdbUser: user for the connection to the AXDB. Used only if axdbManagerClass =it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>axdbuser (usual user)</td>
</tr>
<tr>
<td>axdbPwd: password for the axdbUser for accessing database. Used only if axdbManagerClass =it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>mkzamk (usual pwd)</td>
</tr>
<tr>
<td>axdbMapping: file that contains the mapping for AXDB query engines (not used in this project)</td>
<td>C:\Inetpub\wwwroot\axmedis\metadata-mapper.xml (usually it is there)</td>
</tr>
<tr>
<td>axdbDataSource: connection pool to be used with Tomcat (usually not to be changed unless you change also context.xml). Used only if axdbManagerClass =it.exitech.axmedis.axdb.DataSourceAxdbManager</td>
<td>jdbc/listenerResource</td>
</tr>
<tr>
<td>axdbManagerClass: single ton class to be used for connecting to the AXDB. If it.exitech.axmedis.axdb.DataSourceAxdbManager is used please change also Camart.war/META-INF/context.xml</td>
<td>it.exitech.axmedis.axdb.DataSourceAxdbManager (recommended)</td>
</tr>
<tr>
<td>initialConnections: number of connections in the pool. Used only if axdbManagerClass =it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>10 (suggested value)</td>
</tr>
<tr>
<td>increment: number of connection incremented when the pool is exhausted. Used only if axdbManagerClass =it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>5 (suggested value)</td>
</tr>
<tr>
<td>Camart.war\WEB-INF\classes\axcs.properties</td>
<td>Camart.war\WEB-INF\context.xml</td>
</tr>
<tr>
<td>logpollingtimeoutmins: time interval in minutes between two downloads from AXCS webservice</td>
<td>2</td>
</tr>
<tr>
<td>loguser: user entitled to download logs</td>
<td>mario</td>
</tr>
<tr>
<td>logpassword: password of the user entitled to download logs</td>
<td>xxxxx</td>
</tr>
<tr>
<td>endpoint: endpoint for the statistic WS (not used in this project)</td>
<td><a href="http://localhost:8080/AXCSStatistics/services/Statistics">http://localhost:8080/AXCSStatistics/services/Statistics</a></td>
</tr>
<tr>
<td>axcslogendpoint: endpoint of the AXCS Reporting Web Service</td>
<td><a href="http://flauto.dsi.unifi.it:8080/AXCSReporting/services/Reporting">http://flauto.dsi.unifi.it:8080/AXCSReporting/services/Reporting</a></td>
</tr>
</tbody>
</table>
5.2.9 Errors reported and that may occur

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description and rationales</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountException</td>
<td>Happens if user has not the rights to access the service, or if the HTTP session was terminated by inactivity time-out. AIServlet should gently-fail upon this.</td>
</tr>
</tbody>
</table>

5.3 Administrative Information Integrator (AII)

5.3.1 General Description of the Module

Administrative Information integrator is a critical part of the AXMEDIS system since it is the real bridge between the AXMEDIS world and the world of company’s CMS and CRM for taking in account administrative and legal aspects (such reclaim for payment not done and so on).

This component has also a double face since it can operate in a dual manner: used for polling information from AXMEDIS system when needed by distributor for example, or used for pushing information in the CMS as soon as they are available for example in the case of collecting societies.

The AII can also completely remotely managed by a Web Service that offers all the functionalities guaranteed by the web application.

The operating mode is determined by accounting people during the installation/configuration of the system when it will be established whose fields have to be exported from the DB to the CMS and the frequency of exporting. When a frequency is set, the Administrative Information Integrator will work in push mode, pushing information in the CMS import area, otherwise it operates in polling mode by starting the update in the CMS by a link to a web page.

The principal specification arises, as always, from requirements. Administrative Information Integrator has to:

- **interface with different CMS technologies**: this means that administrative information integrator has to export its reports in a portable format such as an XML that will be defined in the detailed specification. The generated XML can then be parsed and transformed by the way of standard mechanisms such as XSL transformation;
- **store administrative information into the Content Provider database**: this operation will be possible in two ways, or by preparing a file to be put in the import area on the content provider database or by interfacing with the services offered by the CMS of the Content provider by standard mechanisms such as web services or other remote interface available to the Administrative Information Integrator;
- **communicate with the AXDB to get administrative information related to a specific Content Provider**: this is the minimum security requirement that have to be established in order to be sure to distribute information to entitled persons only.
- **guarantee privacy of sensitive data via protection mechanisms**: apart from what has already been stated, if a network connection is necessary to transfer data and the CMS of the Content Provider permits a connection on a secure channel, then the data will be sent encrypted.

5.3.2 Administrative Information Integrator in polling mode

This is the operational mode when an automatic update of the data in the CMS is not set. In this mode the Accountant connects to a web page and issue a simple command to the Administrative Information Integrator that visualizes in the web page or exports in the CMS Import area the information that have not been already exported according to the format defined during the installation and configuration process.

The process that happens in the Accounting Area when the Administrative Information Integrator is in polling mode can be modeled by the following diagram.
5.3.3 Administrative Information Integrator in push mode

All the times that during the configuration of the Administrative Information Integrator the timed insertion of administrative information has been selected, the Administrative Information Integrator is enabled to put in the import area of the CMS all the information that the CMS accountant has selected. The updating of the CMS should be also synchronized by a trigger sent by Core Accounting Manager and reporting tool.

In the following diagram the timed activation is reported, by which the Administrative Information Integrator and the CAMART operates asynchronously on a timed basis.
In the following diagram the timed activation is reported, by which the Administrative Information Integrator is triggered by the CAMART.
Core Accounting Manager and reporting tool

AXDB

AXCS Reporting Tool

Timed Activation

Request Action-Logs

return data

Map Action-Logs in DB

Ack

Activate All

Get Action-Logs

return data

Export data in CMS

Administrative Information Integrator

CMS
5.3.4 Administrative Information Integrator and CAMART integration

It is necessary to point out how CAMART and AII are related in order to have the right tool doing the right work. CAMART is the only interface toward the AXCS, while AII will read the Logs that are already in the AXDB; CAMART has the duty to put log on AXDB with a predefined scheduled period. CAMART will offer to the user an interface for getting logs that are on AXDB independently of the CMS exporting.

In this section an XML schema for the general format of the AII record file is defined. For each partner, on the basis of the requirements shown in DE 9.1.1, an XSL has been created to be applied to the XML in order to obtain a document in the user target format. This process can be repeated for each new partner of affiliated that joins AXMEDIS if the general XML file is not a supported format. In the XML file, the Business Information will not appear and the factory operation will be considered optional, since it is not known if they can be provided or not. The general schema is reported below in graphical form, and in Section 8 in the xsd format..
DE9.1.5 – Integrated Prototype of CMS integration and feedback

AXMEDIS project

CONFIDENTIAL

33
5.3.5 Module Design in terms of Classes

- AdminInformIntegrator: it is the main interface of the service, the business logic is implemented in AdminInformIntegratorImpl;
- AdminInformIntegratorImpl is the controller class of Administrative Information Integrator. It keeps track of the users and the various asynchronous Poller tasks;
- Poller is a class wrapping a polling task, which can be use synchronously or asynchronously. It works synchronously when Administrative Information Integrator is in polling mode, and asynchronously when the service is in pushing mode. In the last case the class is used as a TimerTask that polls data periodically and sends them back to controller class, which publish them at the exporting path selected by user;
- AxDBLogReader is the class in charge to read data from AXDB;
- User is the class that keeps track of all info about logged user, from user ID to configuration for exporting/formatting the AXDB results.

5.3.6 All Prototype User Interface description

Regarding the Administrative Information Integrator seen from the user perspective, the authentication screen is the first page that appears:
and the User after a login with authentication and authorization phase is bring to the following User Interface that allows to:

- Configure the AII for the logged user, giving him the chance to select a CMS style and the FTP URL where the logs will be transferred after each data pushing
- Configure the time intervals in minutes between two pushing of data and enabling or disabling pushing
- Poll raw data or formatted data in the case the user has already saved a configuration
The system will push data in the configured ftp URL at the predefined interval in the format specified in DE 3.1.2.2.15 after getting them from the remote AXCS. In the case of polling the data in XML (formatted or unformatted) will be presented on the screen allowing the user to manually save the file for his own purposes and allowing him to use it immediately. The system tracks the time at which a log has been requested in order to avoid multiple gathering of the same data.

If the user has not chosen the configuration the general format is displayed, if the Poll now button is pressed:
While after the configuration of the format will appear:
5.3.7 All Web Service Interface

All the AII operation shown above can be also performed by using a Web Service that allows a full control of the application behaviour allowing an automated configuration or polling without human interaction.

The WSDL is reported below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
xmlns:y="http://www.axmedis.org/adminInIn" xmlns:ns="http://www.axmedis.org/adminInIn/in-out.xsd"
targetNamespace="http://www.axmedis.org/adminInIn">
  <types>
    <xs:schema targetNamespace="http://www.axmedis.org/adminInIn/in-out.xsd">
      <xs:element name="checkUserParams">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="user"/>
            <xs:element type="xs:string" name="password"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element type="ns:userHashType" name="userHash"/>
      <xs:element name="setConfigurationParams">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="userHash"/>
            <xs:element name="setConfigurationParams"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </types>
</definitions>
```
<xs:element ref="ns:userHash/>
<xs:element name="configuration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="cmsStyle">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="internal"/>
            <xs:enumeration value="AFI"/>
            <xs:enumeration value="ANSC"/>
            <xs:enumeration value="ILABS"/>
            <xs:enumeration value="EUTELSAT"/>
            <xs:enumeration value="TISCALI"/>
            <xs:enumeration value="SEJER"/>
            <xs:enumeration value="XIM"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element type="xs:anyURI" minOccurs="0" name="exportingPath"/>
      <xs:element type="xs:boolean" minOccurs="0" name="backUp"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="setPushingModeParams">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
      <xs:element type="xs:boolean" name="enabled"/>
      <xs:element type="xs:int" name="period"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="pollParam">
  <xs:complexType>
    <xs:choice>
      <xs:element type="xs:string" name="csv"/>
      <xs:element type="xs:anyType" name="xml"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="getLastPushingTimeParam">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="ns:userHashType" name="pollParam"/>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getPushingTimeParam">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="ns:userHashType" name="pollParam"/>
      <xs:element type="xs:int" name="minutes"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="setConfigurationIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="ns:userHashType" name="pollParam"/>
      <xs:element type="xs:int" name="minutes"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="setConfigurationOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="report"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkUserOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns:userHash"/>
<message name="setConfigurationOut">
  <part name="ret" element="ns:operationSuccessful"/>
</message>

<message name="setPushingModeIn">
  <part name="param" element="ns:setPushingModeParams"/>
</message>

<message name="setPushingModeOut">
  <part name="ret" element="ns:operationSuccessful"/>
</message>

<message name="pollIn">
  <part name="param" element="ns:pollParam"/>
</message>

<message name="pollOut">
  <part name="ret" element="ns:report"/>
</message>

<message name="getLastPushingTimeIn">
  <part name="param" element="ns:getLastPushingTimeParam"/>
</message>

<message name="getLastPushingTimeOut">
  <part name="ret" element="ns:minutes"/>
</message>

<portType name="adminInInPortType">
  <operation name="checkUser">
    <input message="y:checkUserIn"/>
    <output message="y:checkUserOut"/>
  </operation>

  <operation name="setConfiguration">
    <input message="y:setConfigurationIn"/>
    <output message="y:setConfigurationOut"/>
  </operation>

  <operation name="setPushingMode">
    <input message="y:setPushingModeIn"/>
    <output message="y:setPushingModeOut"/>
  </operation>

  <operation name="poll">
    <input message="y:pollIn"/>
    <output message="y:pollOut"/>
  </operation>

  <operation name="getLastPushingTime">
    <input message="y:getLastPushingTimeIn"/>
    <output message="y:getLastPushingTimeOut"/>
  </operation>
</portType>

<binding name="adminInInBinding" type="y:adminInInPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>

  <operation name="checkUser">
    <input>      <soap:body use="literal"/>
    </input>
    <output>       <soap:body use="literal"/>
    </output>
  </operation>

  <operation name="setConfiguration">
    <input>      <soap:body use="literal"/>
    </input>
    <output>       <soap:body use="literal"/>
    </output>
  </operation>

  <operation name="setPushingMode">
    <input>      <soap:body use="literal"/>
    </input>
    <output>       <soap:body use="literal"/>
    </output>
  </operation>

  <operation name="poll">
    <soap:operation soapAction="http://www.axmedis.org/aii/poll" style="document"/>  
    <input>      <soap:body use="literal"/>
    </input>
    <output>       <soap:body use="literal"/>
    </output>
  </operation>

  <operation name="getLastPushingTime">
    <soap:operation soapAction="http://www.axmedis.org/aii/getLastPushingTime" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
The functionalities can be easily understood also by the means of the following pictures.

It should be noted that before doing any other operation, the checkUser operation has to be performed.

### 5.3.8 Technical and Installation information

To deploy the Administrative Information Integrator service a PC with Java 2 1.5 Runtime environment and Apache Tomcat 5.5.20 is required. The service will be distributed as a WAR file to be deployed in %TOMCAT_ROOT%/webapps.

### 5.3.9 Draft User Manual

The configuration interface appears after the login that allows to map the user with the chosen configuration. If the user do not select any particular configuration, data will be returned in the neutral format specified in this document, while if the user selects one of the predefined CMS styles the data will be formatted accordingly.

Logs can be extracted in the current time instant with the “Poll Now” button or scheduled every n minutes by the Push section.

The Export URL in the first section is the ftp url where the data formatted according to the CMS style have to be put each time the pushing action is activated.

### 5.3.10 Integration and compilation issues

Since Java and Apache Tomcat technologies, combined with W3C standard protocol (HTTP) and mark-up language (XHTML), were used, there is not any known issue related to interoperability and integration in different context.
<table>
<thead>
<tr>
<th>Config parameter</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>axdbUrl: URI of the axdb for the jdbc</td>
<td>jdbc:mysql://localhost/axdbv4?jdbcCompliantTruncation=false</td>
</tr>
<tr>
<td>axdbUser: user for the connection to the AXDB. Used only if axdbManagerClass=it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>axdbuser (usual user)</td>
</tr>
<tr>
<td>axdbPwd: password for the axdbUser for accessing database. Used only if axdbManagerClass=it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>mkzamk (usual pwd)</td>
</tr>
<tr>
<td>axdbMapping: file that contains the mapping for AXDB query engines (not used in this project)</td>
<td>C:\Inetpub\wwwroot\axmedis\metadata-mapper.xml (usually it is there)</td>
</tr>
<tr>
<td>axdbDataSource: connection pool to be used with Tomcat (usually not to be changed unless you change also context.xml). Used only if axdbManagerClass=it.exitech.axmedis.axdb.DataSourceAxdbManager</td>
<td>jdbc/listenerResource</td>
</tr>
<tr>
<td>axdbManagerClass: single ton class to be used for connecting to the AXDB. If it.exitech.axmedis.axdb.DataSourceAxdbManager is used please change also Camart.war/META-INF/context.xml</td>
<td>it.exitech.axmedis.axdb.DataSourceAxdbManager (recommended)</td>
</tr>
<tr>
<td>initialConnections: number of connections in the pool. Used only if axdbManagerClass=it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>10 (suggested value)</td>
</tr>
<tr>
<td>increment: number of connection incremented when the pool is exhausted. Used only if axdbManagerClass=it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>5 (suggested value)</td>
</tr>
</tbody>
</table>

### 5.3.11 Errors reported and that may occur

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description and rationales</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountException</td>
<td>Happens if user has not the rights to access the service, or if the HTTP session was terminated by inactivity time-out. AIIServlet should gently-fail upon this.</td>
</tr>
</tbody>
</table>
5.4 CAMART for Statistics

5.4.1 Introduction
The role of Core Accounting manager and Reporting Tool (CAMART) for Statistics is strictly bound with database for logs (provided by AXCS) since it has to gather information from AXMEDIS Certifier and Supervisor about Action Log and provide them to the user via web page or web service interface.

The statistics are also accessible also by a Web Service Interface whose WSDL is reported below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <types>
    <xs:schema targetNamespace="http://www.exitech.it/CamartStats.xsd" elementFormDefault="qualified" attributeFormDefault="unqualified">
      <!-- version 1.3 (25-Jul-2006) JWSDP compatible -->
      <xs:element name="CamartRecord">
        <xs:annotation>
          <xs:documentation>Root element for a Camart record</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AXCID" type="xs:string">
              <xs:annotation>
                <xs:documentation>log ID</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="executionTimestamp" type="xs:dateTime">
              <xs:annotation>
                <xs:documentation>timestamp of the execution of the Transaction</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="registrationTimestamp" type="xs:dateTime">
              <xs:annotation>
                <xs:documentation>Timestamp of the registration of the Transaction in AXCS</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="AXOID">
              <xs:annotation>
                <xs:documentation>ID of the object inside the axmedis system</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="40"/>
                  <xs:WhiteSpace value="collapse"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="AXDID">
              <xs:annotation>
                <xs:documentation>log ID</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
    </definitions>
```
AXMEDIS Object Distributor ID that has performed the transaction

Unique AXMEDIS Object Distributor ID that has performed the transaction

 Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Root element of internal Camart report

report

Factory Information: Unique standard ID that has performed the transaction

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field will be filled with the content of the field "type" of the "DCMI" table if present

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.
5.4.2 The prototype
The system is divided mainly in two parts, that are the CAMART for statistical analysis, that in its first prototype is quite raw in the statistics provided and in the operators that can be applied. The User interface is reported in the following. The first screen is a login screen:
After the login
Where it can be evidenced that after providing a user name and password and a search criteria in the Date Field, the system will ask for the records contained in the AXCS and after getting such information the record are filtered according to some criteria based on “equal” operator applied to the fields AXOID, AXCID, AXDID, AXDOM, AXWID.

Moreover it is possible to add charts that are summary of the statistics in the form of Top and Bottom ten that can be provided for AXOID, AXCID, AXDID, location and operation.

By adding two charts we obtain:
If a request without filtering is issued a web page like the following is obtained:
And by collapsing the results we obtain:

<table>
<thead>
<tr>
<th>Log ID</th>
<th>AXCID</th>
<th>Version</th>
<th>AXCID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSO_560ead4e-b3d8-339b-df26-9404e49e273c</td>
<td>OB1_1e7d914a444753764-a70c-5d840744d23</td>
<td>1.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_a22b04c3-1115-b6b7-0a488a5d3d3d3</td>
<td>OB1_887215ee-98f8-3afa-972b-8f2d9eda013a1</td>
<td>7.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_1e4eac20-5bac-3173-8d9e-06d6d72343a</td>
<td>OB1_9c6c7b30-1a30-1366-a248-862f23a37657a</td>
<td>4.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_e92a151f-baeb-3e6b-884b-1b7f11536109</td>
<td>OB1_1ac2d94-ae7c-3056-84a2-b926d2657ded</td>
<td>4.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_0db070496-71c2-3d77-3c1f-4095a266969e</td>
<td>OB1_3f4c7a7e-3e1b-3b32-8c0e-70617a4cd6ae</td>
<td>6.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_97060-05f9-3905-5b59-8000e6bed47</td>
<td>OB1_74661ab5-7d54-3121-857a-64a11d4356f</td>
<td>1.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_90129c-5a64-3e01-a501-4b91a5b93c8</td>
<td>OB1_b27a78e1-95cd-3042-b7b4-3574ae311f7</td>
<td>5.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_b6c2ca5d-5d5d-aa19-2370-3e9c-a9f0-2d87cb2dd5</td>
<td>OB1_4a1919-19370-3e9c-a9f0-2d87cb2dd5</td>
<td>4.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
<tr>
<td>CSO_85a7894c-7a54-379b-3f5-444a90765668</td>
<td>OB1_f1622f4d-d640-3021-8163-b1c3e3b0bde8</td>
<td>4.0</td>
<td>BUS_03719d29-e695-4db7-841c-078ae7db2f DIS_</td>
</tr>
</tbody>
</table>
The charts can be provided in the form of the opposite of the request in terms of bottom and top:
The filtering is now provided AXCS server side in order to minimize the load on the network, and the results can be filtered as in the example below, where a filter on both AXOID and AXCID are provided.
The results obtained are:
5.4.3 Draft User Manual

Regarding the interface for statistics, the user manual is very simple since we have 3 sections that are:

- **Dates**: in this first part the range of date/time we want to analyze must be imposed. If the final date is left empty it is assumed the current date/time
- **Filters**: this section allows to filter the data for a variety of parameters such as AXOID, AXCID, AXDID, location etc, in order to have for example statistics only for a certain location (say Italy and for a certain creator)
- **Charts**: The user can ask to the system to have high level statistics such as top ten and bottom ten based on different parameters

5.4.4 Integration and compilation issues

Since Java and Apache Tomcat technologies, combined with W3C standard protocol (HTTP) and mark-up language (XHTML), were used, there is not any known issue related to interoperability and integration in different context.

<table>
<thead>
<tr>
<th><strong>Config parameter</strong></th>
<th><strong>Possible values</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Camart.war:WEB-INF/classes/axdb.properties</td>
<td>jdbc:mysql://localhost/axdbv4?jdbcCompliantTruncation=false</td>
</tr>
<tr>
<td>axdbUser: URI of the axdb for the jdbc</td>
<td>axdbuser (usual user)</td>
</tr>
<tr>
<td>if axdbManagerClass</td>
<td>mkzamk (usual pwd)</td>
</tr>
<tr>
<td>=it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td></td>
</tr>
<tr>
<td>axdbPwd: password for the axdbUser for accessing database. Used only if axdbManagerClass</td>
<td>axdbUser: user for the connection to the AXDB. Used only if axdbManagerClass</td>
</tr>
<tr>
<td>=it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td></td>
</tr>
<tr>
<td>pageNum: number of pages to be fetched</td>
<td></td>
</tr>
<tr>
<td>axdbMapping: file that contains the mapping for AXDB query engines (not used in this project)</td>
<td>C:\inetpub\wwwroot\axmedis\metadata-mapper.xml (usually it is there)</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>axdbDataSource: connection pool to be used with Tomcat (usually not to be changed unless you change also context.xml). Used only if axdbManagerClass =it.exitech.axmedis.axdb.DataSourceAxdbManager</td>
<td>jdbc/listenerResource</td>
</tr>
<tr>
<td>axdbManagerClass: single ton class to be used for connecting to the AXDB. If it.exitech.axmedis.axdb.DataSourceAxdbManager is used please change also Camart.war/META-INF/context.xml</td>
<td>it.exitech.axmedis.axdb.DataSourceAxdbManager (recommended) it.exitech.axmedis.axdb.MysqlAxdbManager (for debug only)</td>
</tr>
<tr>
<td>initialConnections: number of connections in the pool. Used only if axdbManagerClass =it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>10 (suggested value)</td>
</tr>
<tr>
<td>increment: number of connection incremented when the pool is exhausted. Used only if axdbManagerClass =it.exitech.axmedis.axdb.MysqlAxdbManager</td>
<td>5 (suggested value)</td>
</tr>
</tbody>
</table>
| Camart.war\WEB-INF\context.xml file that controls the DataSource to be created in Tomcat. Please change the parameters according to your database configuration | <?xml version="1.0" encoding="UTF-8"?>
<Context docBase="Camart.war" path="/Camart">
<Resource auth="Container"
  driverClassName="org.gjt.mm.mysql.Driver"
  maxActive="8" maxIdle="4"
  name="jdbc/listenerResource" password="mkzamk"
  type="javax.sql.DataSource"
  url="jdbc:mysql://localhost/axdbv4?jdbcCompliantTru
  ncation=false" username="axdbuser"/>
</Context> |
| Camart.war\WEB-INF\classes\axcs.properties factoryuser: user entitled to download logs | mario |
| factorypassword: password of the user entitled to download logs | xxxxx |
| endpoint: endpoint for the statistic WS | http://localhost:8080/AXCSStatistics/services/Statistics |

### 5.4.5 Errors reported and that may occur

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description and rationales</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountException</td>
<td>Happens if user has not the rights to access the service, or if the HTTP session was terminated by inactivity time-out. AllServlet should gently-fail upon this.</td>
</tr>
</tbody>
</table>
6 Analysis of CMS of partners (EXITECH, ALL)

This section is dedicated to the analysis of the partners’ CMS in order to highlight the minimal set of data required by each partner on the basis of its profile (collecting society, distributor, content integrator, content producer, etc).

Each contributing partner has to provide or have provided:
- its role regarding the AII
- the minimal set of data needed by the Administrative part of the CMS
- the format of the interchange file generated by Administrative Information Integrator for the CMS, as described in the specification.

6.1 CMS data retrievable from AXCS (DSI)

Each Factory has its own database which stores pertinent data needed to make its own business.

AII will try to recover some factory’s related data from AXCS, if they are present, otherwise the fields that are impossible to be recovered will be left blank. In any case, not all data requested by the partner are stored in AXCS databases. In the following, data retrievable from AXCS are reported:

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>String</td>
<td>Use of the content (reading, printing, aggregating, editing.</td>
<td>It is impossible to know where an object will be physically used: the only information we have in ActionLog table is the “Location” field which contains information about the country related to the pertinent collecting society</td>
</tr>
<tr>
<td>AreaCode</td>
<td>string</td>
<td>Geographical area in which the object is going to be used.</td>
<td></td>
</tr>
<tr>
<td>AXDID</td>
<td>String</td>
<td>Unique AXMEDIS Object Distributor ID that has performed the transaction</td>
<td></td>
</tr>
<tr>
<td>AXOID</td>
<td>String</td>
<td>ID of the object inside the AXMEDIS system</td>
<td></td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>ID of the User that has performed the transaction</td>
<td></td>
</tr>
<tr>
<td>Device, Destination</td>
<td>string</td>
<td>Which use will be of the Axmedis Object (radio, television, kiosk, portable, others, etc)</td>
<td>the only information available is EstimatedHWFingerprint (in ActionLog table) Information about Device (pertinent Id) can be retrieved querying AXCS databases using EstimatedHWFingerprint value in the actionlog table in the AXCSAccounting database.</td>
</tr>
<tr>
<td>quantity, count,</td>
<td>String</td>
<td>Quantity of the individual transaction object acquired or rented</td>
<td>It is always 1 because in ActionLog table there is one registration for each partner object.</td>
</tr>
<tr>
<td>UtilizationsNumber</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**6.2 AFI CMS related data (AFI)**

### 6.2.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.

- Collecting Society
- Content Owner
- Content Creator

### 6.2.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction ID</td>
<td>string</td>
<td>Unique ID of the AXMEDIS transaction</td>
</tr>
<tr>
<td>AXOID</td>
<td>String</td>
<td>ID of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>ID of the User that has performed the transaction</td>
</tr>
<tr>
<td>Transaction Value</td>
<td>String</td>
<td>The overall value of the transaction</td>
</tr>
<tr>
<td>paymentForm</td>
<td>String</td>
<td>The code describing payment form selected. This code can have one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IC – Cash</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CC – Credit card</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CP – Coupon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PP – Pre paid card</td>
</tr>
<tr>
<td>Destination</td>
<td>String</td>
<td>Which use will be of the AXMEDIS Object (radio, television, kiosk, portable, others, etc)</td>
</tr>
<tr>
<td>TransactionDate</td>
<td>String</td>
<td>When the Object is been purchased</td>
</tr>
<tr>
<td>UtilizationsNumber</td>
<td>String</td>
<td>How many times the objects is going to be used concerning this transaction</td>
</tr>
<tr>
<td>AreaCode</td>
<td>string</td>
<td>Geographical area in which the object is going to be</td>
</tr>
</tbody>
</table>
6.2.3 File format for data input
The file format can be one of the following:

- CSV (comma separated values)

Structure of .TXT file for loading data into MySql DB
MySQL Table for importing data.
File .txt as described below. Data separated by [ ; ].

<table>
<thead>
<tr>
<th>ISRC</th>
<th>&gt;&gt;&gt;&gt;&gt;&gt;</th>
<th>INT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXOID</td>
<td></td>
<td>INT</td>
</tr>
<tr>
<td>AXUID</td>
<td></td>
<td>INT</td>
</tr>
<tr>
<td>Transaction ID</td>
<td></td>
<td>INT</td>
</tr>
<tr>
<td>PaymentForm</td>
<td>IC/CC/CP/PP &gt; CHAR(2)</td>
<td>INT</td>
</tr>
<tr>
<td>TransactionValue</td>
<td></td>
<td>INT</td>
</tr>
<tr>
<td>Destination</td>
<td>A list to be defined &gt; VARCHAR-</td>
<td>INT</td>
</tr>
<tr>
<td>UtilizationNumber</td>
<td></td>
<td>INT</td>
</tr>
<tr>
<td>AreaCode</td>
<td>VARCHAR</td>
<td></td>
</tr>
<tr>
<td>TransactionDate</td>
<td>INT-Unix Timestamp</td>
<td></td>
</tr>
</tbody>
</table>

6.3 EUTELSAT CMS related data (EUTELSAT)

6.3.1 Role
In the following the role of the partner regarding the administrative data to be collected is summarized.

- Content Distributor

6.3.2 Minimal set of data needed
In the following table report the minimal set of data necessary for your administrative purposes

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique descriptor of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>User that has performed the transaction</td>
</tr>
<tr>
<td>action_done</td>
<td>String</td>
<td>The code describing expected usage for the object. This code can have one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P – Purchase/sell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R – Rent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S – Subscription (only for services)</td>
</tr>
<tr>
<td>paymentForm</td>
<td>String</td>
<td>The code describing payment form selected. This code can have one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IC – Cash</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CC – Credit card</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CP – Coupon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PP – Pre paid card</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dxxx – Differed where accepted values for xxx are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D030 – Differed 30 dd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D060 – Differed 60 dd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D090 – Differed 90 dd</td>
</tr>
</tbody>
</table>
The received information will be stored in the following tables:

### 6.3.3 File format for data input

The file format can be one of the following:

- CSV (comma separated values) $\quad Y$

Structure of `.TXT` file for loading data into Postgresql DB.
File `.txt` as described below. Data separated by `[ ; ]`.

<table>
<thead>
<tr>
<th>ISRC</th>
<th>INT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXOID</td>
<td>INT</td>
</tr>
<tr>
<td>AXUID</td>
<td>INT</td>
</tr>
<tr>
<td>Transaction ID</td>
<td>INT</td>
</tr>
<tr>
<td>PaymentForm</td>
<td>IC/CC/CP/PP $\quad &gt; \quad$ CHAR(2)</td>
</tr>
<tr>
<td>TransactionValue</td>
<td>INT</td>
</tr>
<tr>
<td>TransactionTimestamp</td>
<td>INT-Unix Timestamp</td>
</tr>
<tr>
<td>ActionDoneType</td>
<td>VARCHAR</td>
</tr>
</tbody>
</table>

### 6.4 ILABS CMS related data (ILABS)

#### 6.4.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.

- Content Owner
- Content Distributor
- Content Creator
- Content Integrator

#### 6.4.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transactionId</td>
<td>String</td>
<td>Unique transaction ID</td>
</tr>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique descriptor of the object inside the AXMEDIS system</td>
</tr>
</tbody>
</table>
### ISRC
String
Unique standard ID for identifying the piece

### AXUID
String
User that has performed the transaction

### AXDID
String
Unique AXMEDIS Object Distributor ID that has performed the transaction

#### objectCode
String
The code describing object’s kind. This code can have one of the following values:
- RO – Raw object
- AO – Aggregated object
- CO – Course
- SO – Service
- OO – Other object

#### useCode
String
The code describing expected usage for the object. This code can have one of the following values:
- P – Purchase/sell
- R – Rent
- S – Subscription (only for services)

#### paymentForm
String
The code describing payment form selected. This code can have one of the following values:
- IC – Cash
- CC – Credit card
- CP – Coupon
- PP – Pre paid card
- Dxxx – Differed where accepted values for xxx are:
  - D030 – Differed 30 dd
  - D060 – Differed 60 dd
  - D090 – Differed 90 dd
  - D120 – Differed 120 dd

#### quantity
String
Quantity of the individual transaction object acquired or rented

#### transactionValue
String
The overall value of the transaction

#### transactionDate
String
Date of transaction occurrence

#### transactionTime
String
Time of transaction occurrence

#### transactionDiscount
String
Value of applied discount, 0 if none

### 6.4.3 File format for data input
The file format can be one of the following:
- XML file
ILABS Exchange file Data Format

<table>
<thead>
<tr>
<th>XML Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by Andrea Lorenzon (Giunti Interactive Labs S.r.l.) --&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:schema xmlns:xs=&quot;http://www.w3.org/2001/XMLSchema&quot; elementFormDefault=&quot;qualified&quot; attributeFormDefault=&quot;unqualified&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;transaction&quot; type=&quot;transactionType&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:complexType name=&quot;transactionType&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:sequence&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;transactionId&quot; type=&quot;xs:string&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;AXOID&quot; type=&quot;xs:string&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;ISRC&quot; type=&quot;xs:string&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;AXUID&quot; type=&quot;xs:string&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;AXDID&quot; type=&quot;xs:string&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;objectCode&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:token&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;RO&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;AO&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;CO&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;SO&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;OO&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:token&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;useCode&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:token&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element/&gt;</code></td>
</tr>
</tbody>
</table>
The main drawback of this solution consists in the fact that it has to be generated one file for each transaction, but as discussed with Fuschi this seems to be the solution more close to ILABS needs. In this case it is necessary to define a unique name for the file, that can be an UUID or something based on object ID plus timestamp.

### 6.5 TISCALI CMS related data (TISCALI)

#### 6.5.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.

- Content Distributor

#### 6.5.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transactionId</td>
<td>String</td>
<td>Unique ID of the AXMEDIS transaction</td>
</tr>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique ID of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>Unique ID of the user that has performed the piece</td>
</tr>
<tr>
<td>transactionDate</td>
<td>DateTime</td>
<td>Date and Time of the transaction</td>
</tr>
<tr>
<td>transactionValue</td>
<td>String</td>
<td>The overall value of the transaction</td>
</tr>
<tr>
<td>paymentMode</td>
<td>String</td>
<td>The code describing payment mode selected. This code can have one of the following values: CC - Credit card CP - Coupon</td>
</tr>
</tbody>
</table>
6.5.3 File format for data input
The file format can be one of the following:
- XML file

XML Schema of .XML file for loading data into Postgresql DB

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xs:complexType name="transactionType">
        <xs:sequence>
            <xs:element name="transactionId" type="xs:string"/>
            <xs:element name="AXOID" type="xs:string"/>
            <xs:element name="ISRC" type="xs:string"/>
            <xs:element name="AXUID" type="xs:string"/>
            <xs:element name="transactionDate" type="xs:dateTime"/>
            <xs:element name="transactionValue" type="xs:string"/>
            <xs:element name="paymentMode">
                <xs:simpleType>
                    <xs:restriction base="xs:token">
                        <xs:enumeration value="CC"/>
                        <xs:enumeration value="CP"/>
                        <xs:enumeration value="PP"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="transaction" type="transactionType"/>
</xs:schema>
```
6.6 XIM CMS related data (XIM)

6.6.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.
- Content Integrator (primary function)
- Content Creator
- Content Owner (plan to be selling some own content)

6.6.2 Role

6.6.3 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique descriptor of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>User that has performed the transaction</td>
</tr>
<tr>
<td>quantity</td>
<td>String</td>
<td>Quantity of the individual transaction object acquired or rented</td>
</tr>
<tr>
<td>transactionValue</td>
<td>String</td>
<td>The overall value of the transaction</td>
</tr>
<tr>
<td>transactionTimestamp</td>
<td>String</td>
<td>Timestamp of transaction occurrence</td>
</tr>
</tbody>
</table>

6.6.4 File format for data input

The file format can be one of the following:
- XML file: not necessary, unless this simplifies production of the data by being consistent with other partners
- CSV (comma separated values)
  - AXOID, ISRC, AXUID, quantity, transactionValue, transactionDate
  - tab delimiter

6.7 SEJER CMS related data (SEJER)

6.7.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.
- Content Owner
- Content Distributor
- Content Integrator

6.7.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique descriptor of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>User that has performed the transaction</td>
</tr>
<tr>
<td>Device</td>
<td>String</td>
<td>Type of device (computer, PDA, mobile)</td>
</tr>
<tr>
<td>Action</td>
<td>String</td>
<td>Use of the content (reading, printing, aggregating, editing,…)</td>
</tr>
<tr>
<td>TransactionTime</td>
<td>DateTime</td>
<td>Timestamp of transaction occurrence</td>
</tr>
<tr>
<td>LicenseId</td>
<td>String</td>
<td>The identifier of the license used to grant access to the object</td>
</tr>
</tbody>
</table>

6.7.3 File format for data input

The file format can be one of the following:
- CSV (comma separated values)
6.8 ANSC CMS related data (ANSC)

6.8.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.

- Content Owner
- Content Distributor
- Content Creator (soon)
- Content Integrator (potentially)

6.8.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique descriptor of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>User that has performed the transaction</td>
</tr>
<tr>
<td>Range date</td>
<td>date</td>
<td>Range of dates to which the report refers</td>
</tr>
<tr>
<td>Quantity</td>
<td>String</td>
<td>Quantity of the individual transaction object acquired or rented</td>
</tr>
<tr>
<td>transactionValue</td>
<td>String</td>
<td>The overall value of the transaction (all ANSC objects)</td>
</tr>
<tr>
<td>transactionTimestamp</td>
<td>String</td>
<td>Timestamp of transaction occurrence</td>
</tr>
<tr>
<td>transactionperAXOID</td>
<td>String</td>
<td>Number of transaction per single AXOID</td>
</tr>
<tr>
<td>objectCode</td>
<td>String</td>
<td>The code describing object’s kind. This code can have one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO – Raw object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AO – Aggregated object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CO – Course</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SO – Service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OO – Other object</td>
</tr>
<tr>
<td>useCode</td>
<td>String</td>
<td>The code describing expected usage for the object. This code can have one of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P – Purchase/sell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R – Rent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S – Subscription (only for services)</td>
</tr>
</tbody>
</table>

| paymentForm     | String    | The code describing payment form selected. This code can have one of the following |
|                 |           | values:                                                                        |
|                 |           | IC – Cash                                                                       |
|                 |           | CC – Credit card                                                                |
|                 |           | CP – Coupon                                                                     |
|                 |           | PP – Pre paid card                                                               |
|                 |           | Dxxx – Differed where accepted values for xxx are:                               |
|                 |           | D030 – Differed 30 dd                                                           |
|                 |           | D060 – Differed 60 dd                                                           |
|                 |           | D090 – Differed 90 dd                                                           |
|                 |           | D120 – Differed 120 dd                                                          |

6.8.3 File format for data input
Since the CMS of the ANSC which manages the framework of digital content is separated from the administration software we are not able now to ask for a specific format or fields of data beyond them already chosen by other content partners. In this case we think that a simple text file is the more flexible format or, alternatively, an XML file. CSV could be also an alternative since in the future we can find the way to convert it and import in the administrative procedure which at present is NOT a CMS. Our CMS does not comprehend administrative data, just contents.

The file format can be one of the following:
- CSV (comma separated values) : Field sequence as in the table before, and semicolon as separator

### 6.9 HP CMS related data (HP)
NO INFORMATION PROVIDED

### 6.10 BBC CMS related data (BBC)
NO INFORMATION PROVIDED

### 6.11 TEO CMS related data (TEO)

#### 6.11.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.
- Content Distributor

#### 6.11.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionId</td>
<td>String</td>
<td>Unique transaction ID</td>
</tr>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique descriptor of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>User that has performed the transaction</td>
</tr>
<tr>
<td>TransactionValue</td>
<td>String</td>
<td>The overall value of the transaction</td>
</tr>
<tr>
<td>TransactionTimestamp</td>
<td>DateTime</td>
<td>Timestamp of transaction occurrence</td>
</tr>
<tr>
<td>LicenseId</td>
<td>String</td>
<td>The identifier of the license used to grant access to the object</td>
</tr>
<tr>
<td>DeviceId</td>
<td>String</td>
<td>STB which consumed content fingerprint</td>
</tr>
</tbody>
</table>

#### 6.11.3 File format for data input

The file format can be one of the following:
- XML file
**6.12 TI CMS related data (TI)**

NO INFORMATION PROVIDED

**6.13 SDAE CMS related data (SDAE)**

**6.13.1 Role**

In the following the role of the partner regarding the administrative data to be collected is summarized.

- Content Integrator
6.13.2 Minimal set of data needed
In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTIFIER.ID</td>
<td>Integer</td>
<td>LCD commercial client Identifier</td>
</tr>
<tr>
<td>CONTENT.ID</td>
<td>Integer</td>
<td>LCD content identifier</td>
</tr>
<tr>
<td>DATE</td>
<td>Date</td>
<td>Date of use</td>
</tr>
<tr>
<td>MODALITY.ID</td>
<td>Integer</td>
<td>Use modality</td>
</tr>
<tr>
<td>PRICE_LCD</td>
<td>Float</td>
<td>LCD Price</td>
</tr>
<tr>
<td>TIME</td>
<td>Time</td>
<td>Time of use</td>
</tr>
<tr>
<td>COUNTRY.ID</td>
<td>string</td>
<td>ISO 2 code for country of use</td>
</tr>
</tbody>
</table>

6.13.3 File format for data input
The file format can be one of the following:

- XML file
6.14 ELION CMS related data (ELION)

6.14.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.

- Content Distributor

6.14.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
</table>

LCD (xml) Report format

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE LCD.USES (View Source for full doctype...)> 

- <LCD.USES>
  - <NOTIFIER.ID>6</NOTIFIER.ID>
  - <USE>
    - <CONTENT.ID>456</CONTENT.ID>
    - <DATE DAY="01" MONTH="01" YEAR="2004" /> 
    - <MODALITY.ID>1</MODALITY.ID>
  - <PRICE>
    - <PRICE_CLI>2.5</PRICE_CLI>
    - </PRICE>
  - <COUNTRY.ID>ES</COUNTRY.ID>
  - </USE>
  - <USE>
    - <CONTENT.ID>456</CONTENT.ID>
    - <DATE DAY="01" MONTH="01" YEAR="2004" /> 
    - <TIME HOURS="10" MINUTES="01" SECONDS="20" /> 
    - <MODALITY.ID>1</MODALITY.ID>
  - <PRICE>
    - <PRICE_LCD>2.5</PRICE_LCD>
    - </PRICE>
  - <PRICE_CLI>2.5</PRICE_CLI>
  - </PRICE>
  - <COUNTRY.ID>ES</COUNTRY.ID>
  - </USE>
  - </LCD.USES>
```
### 6.14.3 File format for data input

- XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:complexType name="transactionType">
    <xs:sequence>
      <xs:element name="TransactionId" type="xs:string"/>
      <xs:element name="AXOID" type="xs:string"/>
      <xs:element name="ISRC" type="xs:string"/>
      <xs:element name="AXUID" type="xs:string"/>
      <xs:element name="TransactionValue" type="xs:string"/>
      <xs:element name="TransactionTimestamp" type="xs:dateTime"/>
      <xs:element name="LicenseId" type="xs:string"/>
      <xs:element name="DeviceId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

**ELION Exchange file Data Format XSD Schema:**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:complexType name="transactionType">
    <xs:sequence>
      <xs:element name="TransactionId" type="xs:string"/>
      <xs:element name="AXOID" type="xs:string"/>
      <xs:element name="ISRC" type="xs:string"/>
      <xs:element name="AXUID" type="xs:string"/>
      <xs:element name="TransactionValue" type="xs:string"/>
      <xs:element name="TransactionTimestamp" type="xs:dateTime"/>
      <xs:element name="LicenseId" type="xs:string"/>
      <xs:element name="DeviceId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```
6.15 VRS CMS related data (VRS)

6.15.1 Role

In the following the role of the partner regarding the administrative data to be collected is summarized.

- Content Creator: Y

6.15.2 Minimal set of data needed

In the following table report the minimal set of data necessary for your administrative purposes.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionId</td>
<td>String</td>
<td>Unique transaction ID</td>
</tr>
<tr>
<td>AXOID</td>
<td>String</td>
<td>Unique descriptor of the object inside the AXMEDIS system</td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>User that has performed the transaction</td>
</tr>
<tr>
<td>TransactionValue</td>
<td>String</td>
<td>The overall value of the transaction</td>
</tr>
<tr>
<td>TransactionTimestamp</td>
<td>Date/Time</td>
<td>Timestamp of transaction occurrence</td>
</tr>
<tr>
<td>LicenseId</td>
<td>String</td>
<td>The identifier of the license used to grant access to the object</td>
</tr>
<tr>
<td>DeviceId</td>
<td>String</td>
<td>STB which consumed content fingerprint</td>
</tr>
</tbody>
</table>

6.15.3 File format for data input

The file format can be one of the following:

- XML file

VRS Exchange file Data Format XSD Schema:

```xml
<?xml version="1.0" encoding="UTF-8"?>
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:complexType name="transactionType">
    <xs:sequence>
      <xs:element name="TransactionId" type="xs:string"/>
      <xs:element name="AXOID" type="xs:string"/>
      <xs:element name="ISRC" type="xs:string"/>
      <xs:element name="AXUID" type="xs:string"/>
      <xs:element name="TransactionValue" type="xs:string"/>
      <xs:element name="TransactionTimestamp" type="xs:dateTime"/>
      <xs:element name="LicenseId" type="xs:string"/>
      <xs:element name="DeviceId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="transaction" type="transactionType"/>
</xs:schema>
7 Administrative Information Integrator format detailed specification

The union of the information requested by partners can be summarized as in the following table with the addition of the last column where the domain to which the information is related are reported.

Domain are the following:
- AXCS: information that is stored and provided by AXCS
- Business: Information related to business that is stored outside AXMEDIS system
- Factory: Information that can be retrieved in the factory, if present
- AII: Information generated by the Administrative Information Integrator

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Data Description</th>
<th>Information domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportDate</td>
<td>String</td>
<td>Range of dates to which the report refers</td>
<td>AII</td>
</tr>
<tr>
<td>AXDID</td>
<td>String</td>
<td>Unique AXMEDIS Object Distributor ID that has performed the transaction</td>
<td>AXCS</td>
</tr>
<tr>
<td>logId</td>
<td>String</td>
<td>Unique ID of the AXMEDIS transaction</td>
<td>AXCS</td>
</tr>
<tr>
<td>AXLID</td>
<td>String</td>
<td>ID of the licence bounded to the transaction</td>
<td>AXCS</td>
</tr>
<tr>
<td>AXOID</td>
<td>String</td>
<td>ID of the object inside the AXMEDIS system</td>
<td>AXCS</td>
</tr>
<tr>
<td>AXUID</td>
<td>String</td>
<td>ID of the User that has performed the transaction</td>
<td>AXCS</td>
</tr>
<tr>
<td>device</td>
<td>String</td>
<td>Which use has been done for the AXMEDIS Object (radio, television, kiosk, portable, others, etc)</td>
<td>AXCS</td>
</tr>
<tr>
<td>executionTimestamp</td>
<td>String</td>
<td>Timestamp of the execution of the Transaction</td>
<td>AXCS</td>
</tr>
<tr>
<td>registrationTimestamp</td>
<td>String</td>
<td>Timestamp of the registration of the Transaction in AXCS</td>
<td>AXCS</td>
</tr>
<tr>
<td>location</td>
<td>String</td>
<td>Geographical area in which the object is used.</td>
<td>AXCS</td>
</tr>
<tr>
<td>operation</td>
<td>String</td>
<td>Use of the content (reading, printing, aggregating, editing).</td>
<td>AXCS</td>
</tr>
<tr>
<td>quantity</td>
<td>String</td>
<td>Quantity of the individual transaction object acquired or rented</td>
<td>AII, always 1</td>
</tr>
<tr>
<td>transactionValue</td>
<td>String</td>
<td>The overall value of the transaction</td>
<td>Business</td>
</tr>
<tr>
<td>userDetails</td>
<td>String</td>
<td>detail of the user</td>
<td>Business</td>
</tr>
<tr>
<td>paymentForm</td>
<td>String</td>
<td>The code describing payment form selected. This code can have one of the following values:</td>
<td>Business</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IC – Cash</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CC – Credit card</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CP – Coupon</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PP – Pre paid card</td>
<td></td>
</tr>
<tr>
<td>ISRC</td>
<td>String</td>
<td>Unique standard ID for identifying the piece</td>
<td>Factory, this field will be filled with the content of the field “identifier” of the “DCMI” table if present</td>
</tr>
<tr>
<td>objectCode</td>
<td>String</td>
<td>The code describing object’s kind: Raw object, Aggregated object, Course,</td>
<td>Factory, this field fill be filled with the content of the</td>
</tr>
</tbody>
</table>
AXCS and AII information will be provided always.
Factory information will be provided only if the information can be retrieved in the AXDB that means if the object with the given AXOID is inside the factory and is indexed in the AXDB, otherwise will be left blank.
Business Information cannot be provided at all

The mapping among the data requested by the partners and the previous table can be summarized in the following table where the information domain is reported and therefore it can be evidenced if the information will be provided or not.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Partner</th>
<th>Domain</th>
<th>Mapped AII Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>action_done</td>
<td>EUTELSAT, SEJER</td>
<td>AXCS</td>
<td>operation</td>
</tr>
<tr>
<td>AreaCode</td>
<td>AFI</td>
<td>AXCS</td>
<td>location</td>
</tr>
<tr>
<td>AXDID</td>
<td>ILABS</td>
<td>AXCS</td>
<td>AXDID</td>
</tr>
<tr>
<td>AXLID</td>
<td>SEJER</td>
<td>AXCS</td>
<td>AXLID</td>
</tr>
<tr>
<td>AXOID</td>
<td>AFI, EUTELSAT, ILABS, TISCALI, XIM, SEJER, ANSC</td>
<td>AXCS</td>
<td>AXOID</td>
</tr>
<tr>
<td>AXUID</td>
<td>AFI, EUTELSAT, ILABS, TISCALI, XIM, SEJER, ANSC</td>
<td>AXCS</td>
<td>AXUID</td>
</tr>
<tr>
<td>Destination</td>
<td>AFI</td>
<td>AXCS</td>
<td>device</td>
</tr>
<tr>
<td>Device</td>
<td>SEJER</td>
<td>AXCS</td>
<td>device</td>
</tr>
<tr>
<td>ISRC</td>
<td>AFI, EUTELSAT, ILABS, TISCALI, XIM, SEJER, ANSC</td>
<td>Factory</td>
<td>ISRC</td>
</tr>
<tr>
<td>objectCode</td>
<td>ILABS, ANSC</td>
<td>Factory</td>
<td>objectCode</td>
</tr>
<tr>
<td>paymentForm</td>
<td>AFI, EUTELSAT, ILABS, TISCALI, ANSC</td>
<td>Business</td>
<td>---</td>
</tr>
<tr>
<td>quantity</td>
<td>ILABS, XIM, ANSC</td>
<td>AII</td>
<td>1</td>
</tr>
<tr>
<td>Range</td>
<td>ANSC</td>
<td>AII</td>
<td>reportDate</td>
</tr>
<tr>
<td>transaction ID</td>
<td>AFI</td>
<td>AXCS</td>
<td>logId</td>
</tr>
<tr>
<td>transactionDate</td>
<td>AFI, ILABS, TISCALI</td>
<td>AXCS</td>
<td>executionTimestamp</td>
</tr>
<tr>
<td>transactionDiscount</td>
<td>ILABS</td>
<td>AXCS</td>
<td>---</td>
</tr>
<tr>
<td>transactionId</td>
<td>ILABS, TISCALI</td>
<td>AXCS</td>
<td>logId</td>
</tr>
<tr>
<td>transactionperAXOID</td>
<td>ANSC</td>
<td>AII</td>
<td>1</td>
</tr>
<tr>
<td>TransactionTime</td>
<td>ILABS</td>
<td>AXCS</td>
<td>executionTimestamp</td>
</tr>
<tr>
<td>transactionTimestamp</td>
<td>EUTELSAT, XIM, ANSC</td>
<td>AXCS</td>
<td>executionTimestamp</td>
</tr>
<tr>
<td>transactionValue</td>
<td>AFI, EUTELSAT, ILABS, TISCALI, XIM, ANSC</td>
<td>Business</td>
<td>---</td>
</tr>
<tr>
<td>useCode</td>
<td>ILABS, ANSC</td>
<td>AXCS</td>
<td>operation</td>
</tr>
<tr>
<td>UserDetail</td>
<td>AFI</td>
<td>Business</td>
<td>---</td>
</tr>
<tr>
<td>UtilizationsNumber</td>
<td>AFI</td>
<td>AII</td>
<td>1</td>
</tr>
</tbody>
</table>

As a summary, we can say that:
- Business information cannot be provided
- Factory information will be provided if present
- AXCS information will be stored in the AII DB
- AII information will be generated on the fly

A new version of the table related to the AXCS log needs to be generated starting from the data collected in this deliverable, and therefore the new implementation of such tables will be as in the next figure. Only indexes are subject to change.
As already stated in the introduction, the AII can be configured for each user in the company and therefore more than 1 profile for each company can be created. This allows different people to manage differently formatted information, at least in the so-called “polling” mode where the user can ask on demand the information needed.

This part needs the support of a database structure that can be modelled as showed below:
The DB structure suppose that the same style can be used by different people in the company and therefore put it in an external table, while in the main table together with the AXIUD are reported the main information needed to provide the file to the CMS/CRM system. These information are mainly:

- pushEnable shows if the push modality is enabled or less.
- downloadPath: the path where the user will download the file generated by the AII. This must contain also the credential to access such as ftp://user:passwd@server.foo.com/dir
- pushFrequency is the frequency of pushing operation in minutes
- lastUpdate contains the registrationTimestamp of the last log that has been downloaded by the user

### 8 Table description for database AXDB (CAMART and AII related tables) and XML schemas

In this section the AXDB table involved with CAMART and AII are described. The table for storing the information provided by AXCS Reporting Web service is reported below with the description of the fields.
**AXCS-Log**

**actionlog**

<table>
<thead>
<tr>
<th>ColumnName</th>
<th>DataType</th>
<th>PrimaryKey</th>
<th>NotNull</th>
<th>Flags</th>
<th>Default Value</th>
<th>Comment</th>
<th>AutoInc</th>
</tr>
</thead>
<tbody>
<tr>
<td>logId</td>
<td>BIGINT(20)</td>
<td>PK</td>
<td>NN</td>
<td></td>
<td></td>
<td></td>
<td>AI</td>
</tr>
<tr>
<td>axoid</td>
<td>VARCHAR(40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operationDetailId</td>
<td>INTEGER(11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>registrationTimestamp</td>
<td>BIGINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>executionTimestamp</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>axid</td>
<td>VARCHAR(40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>axcid</td>
<td>VARCHAR(40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>axwid</td>
<td>VARCHAR(40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>axuid</td>
<td>VARCHAR(40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>location</td>
<td>VARCHAR(255)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>device</td>
<td>VARCHAR(255)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**IndexName**

- PRIMARY
- actionlog_axoid
- actionlog_opId
- actionlog_registration
- actionlog_execution

**IndexType**

- PRIMARY
- Index

**Columns**

- logId
- axoid
- operationDetailId
- registrationTimestamp
- executionTimestamp
The table for storing the information needed by AII below with the description of the fields.

**Administrative Information Integrator**

### aiistyle

<table>
<thead>
<tr>
<th>ColumnName</th>
<th>DataType</th>
<th>PrimaryKey</th>
<th>NotNull</th>
<th>Comment</th>
<th>AutoInc</th>
</tr>
</thead>
<tbody>
<tr>
<td>styleId</td>
<td>BIGINT</td>
<td>PK</td>
<td>NN</td>
<td></td>
<td>AI</td>
</tr>
<tr>
<td>description</td>
<td>TEXT</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xslContent</td>
<td>TEXT</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IndexName: PRIMARY

IndexType: PRIMARY

Columns: styleId

### aiiconfiguration

<table>
<thead>
<tr>
<th>ColumnName</th>
<th>DataType</th>
<th>PrimaryKey</th>
<th>NotNull</th>
<th>Comment</th>
<th>AutoInc</th>
</tr>
</thead>
<tbody>
<tr>
<td>axuid</td>
<td>VARCHAR(40)</td>
<td>PK</td>
<td>NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>styleId</td>
<td>BIGINT</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pushEnabled</td>
<td>BOOL</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>downloadPath</td>
<td>VARCHAR(255)</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pushFrequency</td>
<td>BIGINT</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### UserGroup

#### users

<table>
<thead>
<tr>
<th>ColumnName</th>
<th>DataType</th>
<th>PrimaryKey</th>
<th>NotNull</th>
<th>Flags</th>
<th>DefaultValue</th>
<th>Comment</th>
<th>AutoInc</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXUID</td>
<td>VARCHAR(40)</td>
<td>PK</td>
<td>NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NickName</td>
<td>VARCHAR(64)</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passwd</td>
<td>VARCHAR(64)</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>VARCHAR(64)</td>
<td></td>
<td>NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>note</td>
<td>TEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GroupIDPk</td>
<td>VARCHAR(40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IndexName</th>
<th>IndexType</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMARY</td>
<td>PRIMARY</td>
<td>axuid</td>
</tr>
<tr>
<td>Users_FKIndex1</td>
<td>Index</td>
<td>GroupIDPk</td>
</tr>
</tbody>
</table>
8.1 Formal description of format All Record

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <!-- version 1.1 (2-Mar-2006) -->
  <xs:element name="AIIRecord">
    <xs:annotation>
      <xs:documentation>Root element for an Administrative Information Intergrator record</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="logId" type="xs:string">
          <xs:annotation>
            <xs:documentation>Unique ID of the Axmedis transaction</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="executionTimestamp" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>timestamp of the execution of the Transaction</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="registrationTimestamp" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>Timestamp of the registration of the Transaction in AXCS</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="AXOID" type="xs:string">
          <xs:annotation>
            <xs:documentation>ID of the object inside the axmedis system</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="40"/>
              <xs:whiteSpace value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="AXUID" type="xs:string">
          <xs:annotation>
            <xs:documentation>ID of the User that has performed the transaction</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="40"/>
              <xs:whiteSpace value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="AXDID" type="xs:string">
          <xs:annotation>
            <xs:documentation>Unique AXMEDIS Object Distributor ID that has performed the transaction</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="40"/>
              <xs:whiteSpace value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
DE9.1.5 – Integrated Prototype of CMS integration and feedback

transaction

The documentation element represents the ID of the licence bounded to the transaction.

device

The documentation element represents which use has been done for the Axmedis Object (radio, television, kiosk, portable, others, etc).

operation

The documentation element represents the use of the content (reading, printing, aggregating, editing).

location

The documentation element represents the geographical area in which the object is used.

ISRC

The documentation element represents the factory information: unique standard ID for identifying the piece.

objectCode

The documentation element represents the factory information: this field will be filled with the content of the field “type” of the “DCMI” table if present.

reportDate

The documentation element represents the range of dates to which the report refers.
8.2 Formal description of format for statistics record

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- version 1.0 (30-Mar-2006) -->
  <xs:element name="CamartRecord">
    <xs:annotation>
      <xs:documentation>Root element for a Camart record</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="AXCSID" type="xs:string">
          <xs:annotation>
            <xs:documentation>log ID</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="executionTimestamp" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>timestamp of the execution of the Transaction</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="registrationTimestamp" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>Timestamp of the registration of the Transaction in AXCS</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="AXOID">
          <xs:annotation>
            <xs:documentation>ID of the object inside the axmedis system</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="40"/>
              <xs:whiteSpace value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="AXDID">
          <xs:annotation>
            <xs:documentation>ID of the object inside the axmedis system</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="40"/>
              <xs:whiteSpace value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
Unique AXMEDIS Object Distributor ID that has performed the transaction

Creator Id

Use of the content (reading, printing, aggregating, editing).

Geographical area in which the object is used.

Factory Information: Unique standard ID for identifying the piece

Factory Information: this field fill be filled with the content of the field “type” of the “DCMI” table if present

ID of the object inside the axmedis system
9 Demostrator Fact Sheet

9.1 Main purposes of the demonstrator

Main purposes of Core Accounting manager and Reporting Tool (CAMART) is strictly bound with database for logs (provided by AXCS) since it has to collect information regarding the B2B activities and B2C actions. AXCS will not store forever its logs and therefore it is necessary for CAMART to gather time by time such logs and store locally in the AXMEDIS database. Such information will be collected on scheduled time interval and CAMART will act as a client of the AXCS Reporting Web Service.

AXMEDIS system is scalable and therefore we have to deal with the fact that some installation can have AXDB, AXCS and other supporting tools on different machines, while others can be less distributed due to a lesser need for speed or storage capacity.

The core accounting manager is a sort of Client side of the bridge between the AXDB and the AXCS databases in order to allow AXCS to be independent by the database. The server side in the AXCS is the Web Service: AXMEDIS Reporting Web Service. The CAMART can be interpreted as a part of the AXMEDIS Database Interface, since is the part of the system that allows writing data related to Action-Logs into the AXMEDIS DB.

Administrative Information integrator is a critical part of the AXMEDIS system since it is the real bridge between the AXMEDIS world and the world of company’s CMS and CRM for taking in account administrative and legal aspects (such reclaim for payment not done and so on).

Main purposes of this component is to operate in a dual manner: used for polling information from AXMEDIS system when needed by distributor for example, or used for pushing information in the CMS as soon as they are available for example in the case of collecting societies.

The AII can also completely remotely managed by a Web Service that offers all the functionalities guaranteed by the web application.

The operating mode is determined by accounting people during the installation/configuration of the system when it will be established whose fields have to be exported from the DB to the CMS and the frequency of exporting. When a frequency is set, the Administrative Information Integrator will work in push mode, pushing information in the CMS import area, otherwise it operates in polling mode by starting the update in the CMS by a link to a web page.
The role of Core Accounting manager and Reporting Tool (CAMART) for Statistics is strictly bound with database for logs (provided by AXCS) since it has to gather information from AXMEDIS Certifier and Supervisor about Action Log and provide them to the user via web page or web service interface.

By using this demonstrator you can:
- Get logs coming from AXCS web service
- Organize such logs in the internal database of the factory
- Generate internal XML format in polling mode
- Generate the XML format provided by one of the partners according to the specification and publishing at a predefined time frequency the resulting XML in an ftp directory or file path
- Generate also in polling mode an XML format that is transformed according to the profiled XSLT.
- Generate top-bottom ten on demand for statistics.
- Use a web service for gathering statistics instead of the GUI only
- Use a Web service to remotely control AII completely in order to be able to automate log collection

### 9.2 Review of the architecture integration with AXMEDIS

CAMART and related tool are strictly bounded with some AXMEDIS components: the AXCS and the AXDB.

The relationships among CAMART and the underlying AXMEDIS components are depicted in the following picture.

All the tools can be used by physical person (Accountant in the picture) by a simple Web Interface or by an automated tool (Machine in the picture) by a web service interface.

This allows as easy integration in your factory allowing you to select the best approach for event logs access on the basis of the way in which you are currently working.
9.3 Description of the effective installation
The demonstrator is a set of 3 web application distributed as WAR that can be deployed directly (after the initial configuration) on top of Tomcat >5.5.20. They must have a connection with AXMEDIS database that can be in the same server or in a different one. It is needed therefore for a full installation to have at least a server with Mysql 4.x and Tomcat 5.5.y where y is greater or equal to 20. This server must be installed with an AXDB but it is not necessary that you install AXDB related services and web services, the database itself is enough. After that you can deploy on Tomcat the web application of Camart, CamartStats and AII.

9.4 AXMEDIS tools
The AXMEDIS tools used by this demonstrator are:
- AXMEDIS Database for storing and retrieving logs and configurations
- AXCS for retrieving logs and statistics by the means of the exposed Reporting and Statistic WS

9.5 Target Market
This demonstrator is strictly integrated with AXMEDIS and therefore we have decided to publish the source code inside the AXMEDIS framework in order to bind it with the platform. The target market is the AXMEDIS factory and therefore it is necessary that its deployment and usage are connected with the installation of the framework. Moreover since several future AXMEDIS user can use different CMS technologies, the possibility to perform training on the logs format customization or the writing of log converter will arise.

9.6 Description of the business model
 Installed with the AXMEDIS framework when a factory is set-up comprised with the AXMEDIS framework fee or as an additional fee to be paid.

9.7 Description of content
Involved contents are AXCS logs and administrative data, but no multimedia content are involved.

9.8 Final Users/ Clients
These tools are not for final user, they are tool for people operating in the factory.

9.9 Partners involved and roles
DSI: for providing AXCS

10 Usage of Camart/CamartStats on WP9 and WP12 demonstrators

10.1 BBC, EUTELSAT, TELECOM

<table>
<thead>
<tr>
<th>Have you used CAMART in your demonstrator</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>How easy was CAMART integration? (from 1 to 5, where 1 means very easy, 5 very difficult)</td>
<td></td>
</tr>
<tr>
<td>How many records are present in the actionlog table of the AXDB linked to CAMART</td>
<td></td>
</tr>
<tr>
<td>Have you used CAMARTSTATS in your demonstrator</td>
<td>NO</td>
</tr>
<tr>
<td>How easy was CAMARTSTATS integration? (from 1 to 5, where 1 means very easy, 5 very difficult)</td>
<td></td>
</tr>
<tr>
<td>How many record do you obtain from CAMARTSTATS in your most complete query</td>
<td></td>
</tr>
</tbody>
</table>

Not Used
10.2 ILABS

<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have you used CAMART in your demonstrator</td>
<td>YES</td>
</tr>
<tr>
<td>How easy was CAMART integration? (from 1 to 5, where 1 means very easy, 5 very difficult)</td>
<td>2</td>
</tr>
<tr>
<td>How many records are present in the actionlog table of the AXDB linked to CAMART</td>
<td>294</td>
</tr>
<tr>
<td>Have you used CAMARTSTATS in your demonstrator</td>
<td>YES</td>
</tr>
<tr>
<td>How easy was CAMARTSTATS integration? (from 1 to 5, where 1 means very easy, 5 very difficult)</td>
<td>2</td>
</tr>
<tr>
<td>How many record do you obtain from CAMARTSTATS in your most complete query</td>
<td>8677</td>
</tr>
</tbody>
</table>

10.3 TISCALI

<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have you used CAMART in your demonstrator</td>
<td>NO</td>
</tr>
<tr>
<td>How easy was CAMART integration? (from 1 to 5, where 1 means very easy, 5 very difficult)</td>
<td></td>
</tr>
<tr>
<td>How many records are present in the actionlog table of the AXDB linked to CAMART</td>
<td></td>
</tr>
<tr>
<td>Have you used CAMARTSTATS in your demonstrator</td>
<td>YES</td>
</tr>
<tr>
<td>How easy was CAMARTSTATS integration? (from 1 to 5, where 1 means very easy, 5 very difficult)</td>
<td>1</td>
</tr>
<tr>
<td>How many record do you obtain from CAMARTSTATS in your most complete query</td>
<td>&gt; 100</td>
</tr>
</tbody>
</table>

10.4 OTHER PARTNER

No answer has been received from other partners
11 Bibliography

DE 9.1.1 Specification of CMS integration and feedback
DE 9.1.2 Mock-up for CMS Integration and Feedback
DE 9.1.3 First Prototype for CMS Integration and Feedback
DE 2.1.1.2.1 User Requirement First Update
DE 2.1.1.2.2 Use case First Update
DE3-1-2-2-15 Specification of AXMEDIS Accounting and Reporting