Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE5.0.1.2 -- ANNEX 3
AXMEDIS Plug-Ins
Software Development Tool Kit, SDK

Version: 1.1
Date: 11/07/2007
Responsible: Peter Ebinger (FHGIGD), Ivan Bruno (DSI) (revised and approved by Coordinator)

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: Internal coded, related to DE5.1.3.1
Contractual Date of Delivery: M34
Actual Date of Delivery: 11/07/2007
Work-Package contributing to the Deliverable: WP4, WP5
Task contributing to the Deliverable: all tasks of WP4 and WP5
Nature of the Deliverable: report
Author(s): FHGIGD, DSI, UPC, UNIVLEEDS, DIPITA, EPFL

Abstract: This document describes the SDK usage for creating AXMEDIS plug-in. The AXMEDIS Tools allows including the functionality of external libraries through an advanced plug-in interface. Once produced an AXMEDIS plug-in it is automatically usable and accessible for a set of AXMEDIS applications: AXMEDIS Editor, Authoring tools, AXMEDIS Content Processing GRID and content processing node and scheduler, AXMEDIS players.

Keyword List: AXMEDIS Editor, AXMEDIS Plug-in format, SDK, GRID, Content Processing, adding capabilities, protection tools, software components, dynamic libraries

AXMEDIS Copyright Notice

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

7. **INFRINGEMENT**

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

8. **GOVERNING LAW AND JURISDICTION**

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

Please note that:

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

Table of Contents

1 Executive Summary and Report Scope

This document describes the SDK usage for creating AXMEDIS plugin.

The AXMEDIS Tools allows including the functionality of external libraries through an advanced plug-in interface.

Once produced an AXMEDIS plugin it is automatically usable and accessible for a set of AXMEDIS applications:

- AXMEDIS Editor, Authoring tools
- AXMEDIS Content Processing GRID and content processing node and scheduler:
- AXMEDIS players

Our Suggestion is to start reading the AXMEDIS for all documents (available at http://www.axmedis.org) before starting reading this document. A short summary is also reported in this document for completeness.

This document includes also two software packages:

- axmedis-de5-0-1-2-annex3-axmedis-plugin-sdk-v1-0-contenprocessing-plugin.zip
- axmedis-de5-0-1-2-annex3-axmedis-plugin-sdk-v1-0-protectiontool-plugin.zip

They are examples for developing Content Processing and Protection Tools AXMEDIS p-lug ins, respectively. They can be freely downloaded.

2 Introduction

AXMEDIS is a consortium and a solution based on the AXMEDIS project partially funded by the European Commission in IST FP6 and including about 35 partners and affiliated partners such as University of Florence, TISCALI, AFI, SEJER, ILABS, EUTELSAT, HP, Telecom Italia, Telecom Lituania, Telecom Estonia, EPFL, FHGIGD, ACIT, Technical University of Catalonia, University of Leeds, STRATEGICA, EXITECH, XIM, University of Reading, KTU, MBI, PKU, SDAE, SIAE, ANSC, ETRI, HEXAGLOBE, GESFOR, ALBENIZ, PENTEX, RIGEL, etc. AXMEDIS has produced innovative research results exploited in new tools and solutions that have been used for developing real trial and demonstrators for automated content production, protection and distribution.

Content:
▪Producers
▪Providers
▪Aggregators
▪Packager
▪Integrators
▪Promoters
▪Distributors
▪..
axmedis
Tools and Solutions
Models and technologies
Content authoring, processing, protection,
distribution, controlling and usage
➢ Supporting both B2B and B2C distribution
➢ Reducing costs for Content Production, processing, etc
➢ Enabling Multichannel Distribution
➢ Enabling Interoperable Distribution of content
➢ Enabling e-commerce of secure digital content

AXMEDIS goals

The AXMEDIS Framework is an open solution which builds on new technologies and tools to:

- reduce costs and increase efficiency for content production, protection (multiple and interoperable DRMs), management and multichannel distribution. It offers effective automation for

  - integrating your Content Management Systems (CMSs), with the distribution systems by automating the communication and maintenance of content and information between the two systems;
  - content gathering and ingestion processes from local and remote CMSs as well as file systems;
  - composition, supporting parallel processing, GRID technology, and optimisation techniques for content ingestion, production, protection and formatting;
  - managing the workflow processes at content-factory level and between different content-factories;
  - the overall process allowing content production on demand.
- support the whole value chain, including composition, packaging, integration, aggregation, synchronisation, formatting, adaptation, transcoding, indexing. Additional features include the integration of both protected and non-protected components within an object, definition of relationships with other resources, metadata integration and remapping/transcoding, protection, license production and verification;
- allow the convergence of the media and interoperability of content to enable multi-channel distributions. The framework supports content distribution:
  - on different channels such as satellite data broadcast, Internet, cellular/mobile network, wireless and traditional media supports such as CDs, DVDs;
  - via different communication technologies, particularly with Peer-to-Peer (P2P) for both B2B (Business-to-Business) and B2C (Business-to-Consumer) levels;
  - to different devices such as PC, PDA, interactive TV (i-TV), set-top box (STB), etc.;
  - with different transaction models on the same channels and content, and with flexibility.
- adopt new methods and tools for innovative, flexible and interoperable Digital Rights Management (DRM) in order to facilitate smooth transition from paper contracts to digital licenses. For examples:
  - the exploitation of MPEG-21 REL (Rights Expression Language) with specific extensions and enhancements;
  - the support of different business and transactions models and their integration;
  - the integration/interoperation of different DRM models such as MPEG-21 REL and ODRL OMA (Open Mobile Alliance).
- harmonise B2B and B2C areas for DRM, bringing the DRM model in the B2B area, supporting production and protection models in the whole value chain;
- increase content accessibility with a P2P platform at B2B level, which can integrate content management systems and workflows.

AXMEDIS implements the AXMEDIS Framework for all, especially small and large industries sharing a common interest in the exploitation of new technologies and solutions. The AXMEDIS Framework can be used to setup and built a large range of applications and services in the area of content production, protection and distribution, examples are also reported in this document. With the flexibility of AXMEDIS framework and dynamic plug-in technology, you can customize your applications and processes according to your needs. Moreover, accessing to the AXMEDIS Framework also means the full access to all the source code or client and server applications.

The AXMEDIS digital content and content components is an **open format** capable of integrating any kind of cross media format (video, images, animations, games, learning objects, multimedia, audiovisual, document, audio, etc.) in any digital format, any kind of metadata including identification, classification, categorization, indexing, descriptors, annotation, relationships and play activities and protection aspects.
The AXMEDIS format permits the combination of content components and their secure distribution in respect of the copyright laws, supporting a large variety of DRM rules and models according to concepts of

interoperability among DRMs (mainly, but not only, based on MPEG-21, with both binary and XML low level formats and OMA). AXMEDIS is open to any DRM models and solutions.
Within the AXMEDIS content any type of cross media content can be included from simple multimedia files to games or software components, for leisure and entertainment, infotainment, and also for managing protected governmental content, healthcare information, business or value information, etc. (http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Cross+Media+Content%3A+Examples)

- AXOM: http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=333

This document describes the **AXMEDIS open architecture and framework**:

- All the AXMEDIS specification is public and its specific use is royalty free. Any company or third party can use the document to create an AXMEDIS compatible solution.
- All AXMEDIS client tools are accessible and are free of charge.
    - Tools to be installed: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2885
    - Tools examples: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Tools+at+Work
    - Videos and tutorials: http://www.axmedis.org/tiki/tiki-index.php?page=Early+Demonstrations+and+Videos
- All the source code of AXMEDIS Framework is accessible if you get affiliated with AXMEDIS such as SIAE, ALBENIZ, RIGEL, PENTEX, HEXAGLOBE, FOCUSEEK, GESFOR, etc. The affiliation fee is low and affordable for all. If you are an institution the Affiliation if FREE of charge if you collaborate to improve the framework in somehow, and you will become an AXMEDIS sustainer and developers;
- The affiliation to AXMEDIS can also be obtained by contributing work or results to the community which improve and extend the AXMEDIS Framework, becoming an AXMEDIS sustainer and developers;.
- The AXMEDIS plug-in technology is public. The source code for creating new plug-ins is public without need to be affiliated.
    - Affiliation agreement http://www.axmedis.org/affiliation.php
    - http://www.axmedis.org/documenti/view_documenti.php?doc_id=1751
- In AXMEDIS the focus is on interoperability and openness of content model, including multi-channel distribution.
- In AXMEDIS the focus is on interoperability of DRM model, including multi-channel distribution.

More technical information on AXMEDIS architecture and framework and about how to access at the AXMEDIS framework getting affiliated to AXMEDIS are available on http://www.axmedis.org .

3 Business to Business areas

To facilitate the collaboration amongst actors in the business area, AXMEDIS is providing a wide set of tools and especially the AXEPTool, a P2P tool for B2B distribution of content and tools. The tools support DRM and the reporting of content usage. Information related to the exploitation of rights along the value chain including the final user is gathered and reported back to the concerned actors. These utilities are very useful to provide the evidence of the exploited rights in a transparent manner to collecting societies or other business partners.
You can join the P2P B2B network of AXMEDIS installing the AXEPTool from: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2960 On the P2P B2B network of AXMEDIS you can download example, video of tutorials, demonstrator, etc. You can publish on the P2P B2B network AXMEDIS and non AXMEDIS objects. ALL THE AXMEDIS partners are on the network and will see your work immediately, please start collaborating with us.

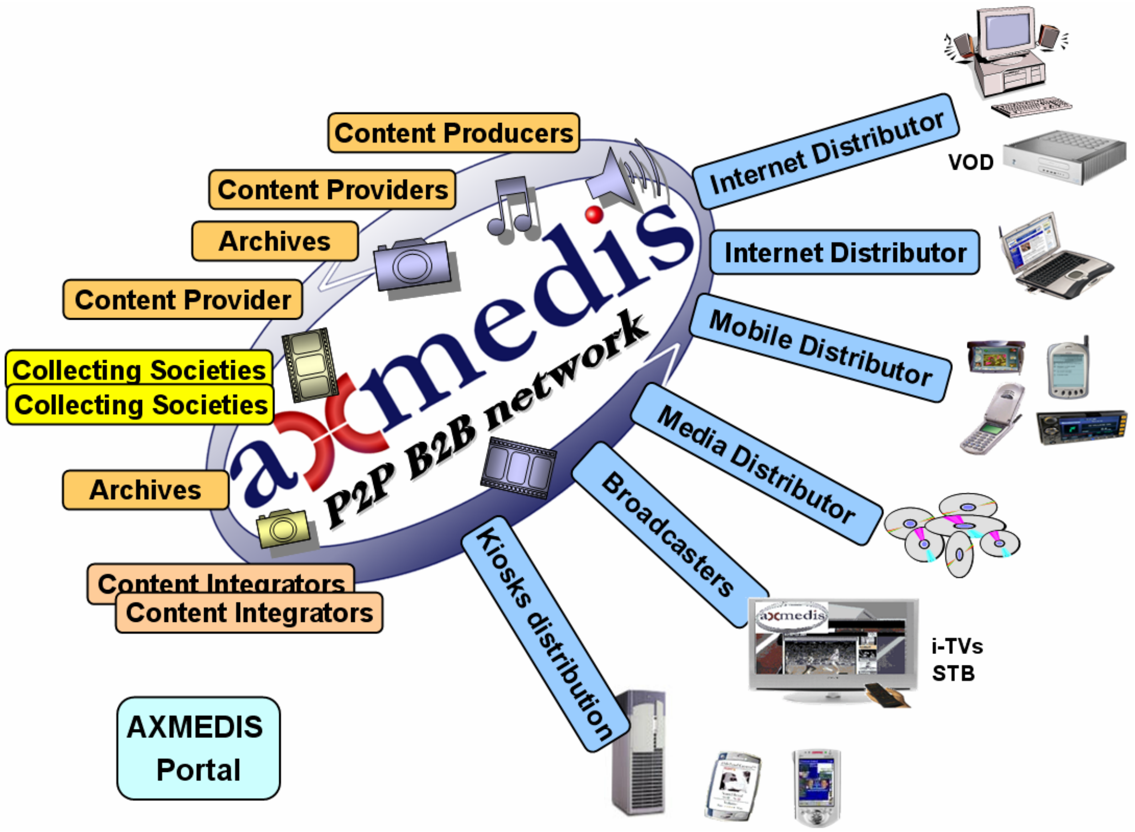


Fig.2 -- AXMEDIS Business to Business area with some distributors

The distribution side may present one or more single distribution paths for each type of content. In AXMEDIS, the content distributors can continue their preferred mechanisms for reaching the final users. The possible Channel Distributors have a large variety of capabilities, they are both of pull and push, and may include off-line and on-line connection from the client to the distributor.

4 AXMEDIS General Architecture

The diagram in figure 3 illustrates all the major areas of the AXMEDIS architecture. Each major component is described in relation to the flow of the content from its acquisition to distribution.

The major areas are the:

- **AXMEDIS Factory** for automatically: collecting content from legacy CMSs, producing the content, programming and scheduling the production process, processing metadata, composing and formatting content, collecting content information from content usage, producing licenses to harmonize the production with workflow applications in the factory and among geographically distributed factories, etc. The AXMEDIS Factory is scalable in the sense that it can satisfy the needs of small and large content producers, integrators, and distributors. The factory is supported by tools for automating the production process and to perform manual editing.
    - General download page: http://www.axmedis.org/documenti/documenti.php?area_id=1
    - Editors: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Editors
    - AXMEDIS Tools to be installed: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2885
    - Tools examples: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Tools+at+Work
    - Videos and tutorials: http://www.axmedis.org/tiki/tiki-index.php?page=Early+Demonstrations+and+Videos

- **AXMEDIS Distribution Area** for automating the content publication and acquisition in the business area allowing the interconnection of AXMEDIS Factories by means of the AXEPTool which is a secure and legal P2P tool. Among connected AXMEDIS Factories, it is also possible to make distributed queries to search for content, and to automatically publish and acquire/update content from/to the business partners, etc. The tools in this area also allow scheduling content distribution and publication towards external web services for example those of front end distribution servers.
    - AXEPTool: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2960
    - Programme and publication: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Programme+and+Publication
    - Example of AXMEDIS based distribution: http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=351
- **AXMEDIS Player** for content playing and execution on several different platforms, to built specific and customized content players, for distributing and sharing content among final users by means of secure P2P tools such as AXMEDIA P2P tool.
    - Tools to be installed: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2885
    - Players: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Player
    - AXMEDIA P2P tool: http://www.axmedis.org/documenti/view_documenti.php?doc_id=3068
    - http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=348
- **AXMEDIS Protection and Supervising tools** for registering users, certificating users, authenticating devices and tools, monitoring all the activities on the AXMEDIS content on AXMEDIS players and tools, processing licenses, managing black lists, and collecting and reporting the information about content usage and rights exploitation, etc.
    - Editors: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Editors

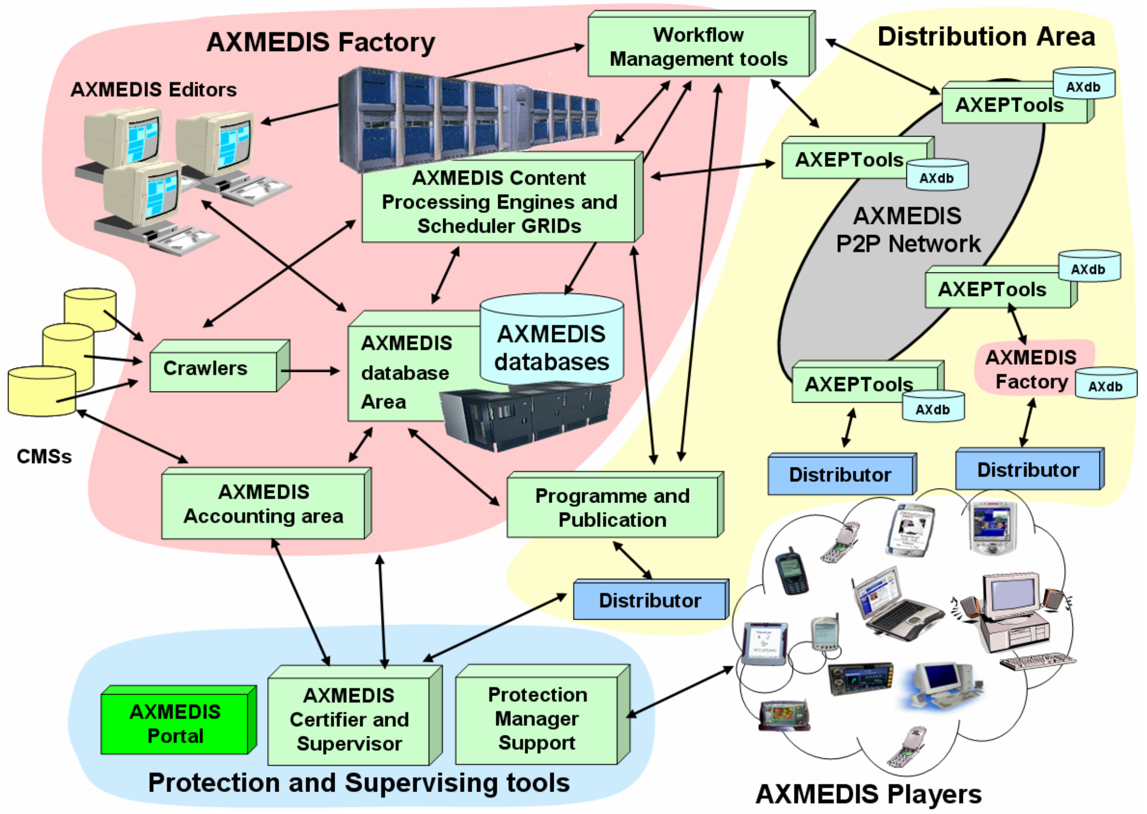


Fig.3 – AXMEDIS Architecture

5 AXMEDIS Factory

A content factory can be built on the basis of AXMEDIS tools in a scalable and flexible manner. For example, tuning is possible for GRID size, type of database, number of authoring tools, number and types of tools/algorithms and libraries for processing content, licenses, integration support based on Workflow or not. An AXMEDS content factory can be customized to satisfy the needs of small and large content producers, integrators, and distributors.

The **AXMEDIS Database Area** includes the AXMEDIS/MPEG-21 database model, supporting the storage and access to content via a large set of metadata for each AXMEDIS object grouped in what is called AXInfo. The AXMEDIS database model can be customized to your needs. The database also includes produced licenses for the objects, histories of performed actions on content, potentially available rights for each digital resource, models of contracts, etc. The AXInfo includes Dublin core plus descriptors and many other metadata for managing protection, lifecycle, etc. Any descriptors and metadata can be added in a flexible manner. Thus, different AXMEDIS factories may be based on different AXInfo and metadata, while automatic adapters can be defined and activated. The database area is based on a scalable database, a powerful AXMEDIS Database manager, and an effective **AXMEDIS Query Support** endowed with an easy to use user interface. The User may perform queries to search for objects and content located in the CMSs, in the local AXMEDIS database and in the virtual database comprised of the AXMEDIS content accessible/published via the P2P network of AXEPTools in the AXMEDIS B2B Network.

- AXDB: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Database+accounting+and+reporting+tools
- COPOP: http://www.axmedis.org/tiki/tiki-index.php?page=COPOP+Client+and+Server+and+Distribution+Portal
- Database: http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=341

The **AXMEDIS Content Processing Area** (AXCP Area) is based on a GRID solution for automating all the activities to be performed for the production, and processing of content. The major tools are the **AXCP GRID Node (Engine) and AXCP Scheduler,** which are respectively the single node (computer) of the GRID and the organizer of processes on the GRID Nodes. They implement a scalable solution to process from smaller collections to huge amount of content per period of days, minutes. The processing algorithms can be specified in terms of script code (in Spider Monkey) allowing the manipulation of complex AXMEDIS data types and simple digital resources and content in general, and for the direct access to the AXMEDIS database and processing queries with the help of the AXMEDIS Query Support. The solution allows the writing of any kind of content processing algorithms, to activate them automatically on some query result, and these can be put in execution as independent processes on a scalable GRID for massive production and processing of digital resources in respect of the DRM.

- AXCP: http://www.axmedis.org/tiki/tiki-index.php?page=Content+Processing
- AXCP http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=335
- Adaptation: http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=338
- Fingerprint: http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=339

The available data types, operators and accessible algorithms allow manipulation of any digital resource in a large number of formats. Algorithms can be defined for massive content composition (packaging, combination, etc.) and content layout formatting (synchronization, image and screen layout, from image sequence to video, etc.), content adaptation (change in resolution, subsampling, change in format, etc.), transcoding, coding, decoding, fingerprint extraction, estimation of descriptors, license adaptation and transcoding, license production and verification, etc.

The users of the **AXMEDIS Content Processing Area (AXCP)** can create Java Script rules for any kind of processing procedures and algorithms.

- AXCP: http://www.axmedis.org/tiki/tiki-index.php?page=Content+Processing
- Rules: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Content+Processing+Scripts

To manipulate/produce:

- Any digital resource:
    - such as images (more than 150 different formats), audio (more than 50 formats), video (more than 50 formats), documents (TXT, PS, HTML, PDF, RTF, DOC, etc.), multimedia (more than 20 formats including MPEG-4, HTML, LOM, etc.);
    - for their transcoding, adaptation, feature and descriptor extraction, recognition, certification, etc.;
    - with functionalities of many well known and powerful processing libraries such as: FFMPEG lib, LibSNDFile, TreTagger, DocFrac, GhostScript, XPDF, HTMLDOC, ImageMagik, MP4Box, Xerces, XALAN, CCPP, etc. (if you are interested in adding more libraries please contact AXMEDIS);
- Packages and their composition and formatting
    - AXMEDIS objects with AXInfo Metadata and indexing,
    - MPEG-21 Objects,
    - including digital resources, metadata (e.g., Dublin Core, etc.),
    - protection information, etc.
- Protected objects and resources, managing protection information:
    - by using MPEG-21 IPMP model, and format
    - by using algorithms for encryption/decryption, scramble, compression, key production, different sizes for keys, etc.
- Licenses on the basis of the business models chosen:
    - By stating grants, conditions, etc.,
    - by verifying license consistency with respect to the potentially available rights, with the license in production, etc.,
    - by using formalism of MPEG-21 REL, and with OMA ODRL – MPEG-21 REL transcoding
- Automatic content and information access
    - database accesses (ODBC, etc.) with direct facilities;
    - database access by means of crawling facilities to access to a larger set of possible database models. They may contain digital content, resources, files, metadata, administrative and licensing information, etc., and can be physically located in several different computer systems and based on several different database models: ODBC, MySQL, ORACLE, MS-SQL, Tamino Lobster XML, etc., or files systems. The access to this information is performed by means of Focuseek Crawler;
    - file system and operating system access;
    - http and ftp accesses;
    - AXMEDIS database access with query support, actualization of selections, active queries, etc.
- Device capabilities format and processing facilities, to take into account for adaptation and/or processing;
- User Profile format and processing facilities, to take into account for adaptation and/or processing;
- WSDL facilities for the activation of WEB services dynamically on the basis of their definition;
- XML facilities for the application of styles and general processing;
- SMIL facilities for the application of templates and styles and processing;
- etc.

The algorithms and procedures used in the AXCP Area can be expanded by using the AXMEDIS Plug In technology that allows customizing and easily expanding the processing capabilities of the AXMEDIS GRID. Algorithms for the extraction of fingerprint, descriptors, adaptation, content processing, DRM adaptation, metadata adaptation, are built as pluggable algorithms. ***Any other library, model and format and related algorithms for their manipulation can be plugged in the AXCP in a very easy manner.***

The AXMEDIS **Workflow Management tools** include a set of micro tools and interfaces which are pervasively connected to all the AXMEDIS tools and plug-ins to allow interfacing the whole content factory to Workflow tools such as Open Flow and BizTalk. The control is performed to define AXMEDIS factory workflow policies and to manage inter-factory workflows policies.

- AXWF: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Workflow+and+Workflow+manager
- http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=350

The **AXMEDIS Editor** is the authoring tool for manually producing AXMEDIS objects when needed and for supporting the designer to create the scripts for the AXCP that could be considered macros of the AXMEDIS Editor. http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=337
The editor is based on the AXMEDIS Object Model (AXOM) and on MPEG-21 model and format. http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=333
The editor includes modules and tools to manipulate and create MPEG-21 and AXMEDIS objects and related information and digital resources such as:

- resource hierarchy viewer and editor;
- visual and behavioral viewer and editor to show/manipulate visual and temporal aspects of related digital resources according to SMIL;
- metadata editor and viewer, to manipulate and view general XML metadata and specific AXInfo metadata;
- DRM viewer and editor to create and verify the DRM licenses;
    - From paper Contracts to Licenses
    - AXMEDIS License Definition and Usage
    - AXMEDIS License Translation and DRM interoperability
- Protection Information viewer and editor to specify, apply and browse protection aspects on the basis of the MPEG-21 IPMP format with extension of AXMEDIS;
- Plug-ins to use algorithms for content processing as those mentioned and used in the AXCP Area mentioned above;
- Plug-ins to allow the integration of AXMEDIS Editor within other editing and viewing applications such as: Video Editors, Image Editors, etc.;
- an interface to connect the AXMEDIS Editor with other external powerful editor tools;
- an interface with workflow (OpenFlow and BizTalk);
- Internal viewers and players for digital resources such as document, images, video, MPEG-4, and audio resources, etc., for more than 250 different file formats.

The **AXMEDIS Accounting Area** includes a set of tools which allow content producers, distributors or collecting societies to collect administrative information and report about their content in order to gather information about the list of rights that have been exploited on their AXMEDIS objects by the final users and by the business users. This information is collected into the AXMEDIS database for further analysis. The acquisition of accounting information is performed by collecting it from the AXMEDIS Certifier and Supervisor, AXCS. The local database and the AXCS provide support to make queries to obtain statistics data on content usage in the area, in the channel, for a type of content, for a period, etc. A specific tool allows extracting data from the AXMEDIS Database to migrate them towards the administrative side of the CMS, such as high level administrative information to prepare the bill at the content users, distributors, etc., to interface with customer relationships services.

- Action logs: http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=347

6 AXMEDIS Distribution Area and Players

The AXMEDIS tools for the distribution area allow automating the content publication and acquisition in the business area establishing also interconnection among different AXMEDIS Factories by means of the so called AXEPTools (AXMEDIS P2P Tool for B2B distribution) which is a secure and legal P2P tool. The tools in this area also allow scheduling content distribution and publication towards external web services for example those of front end distribution servers.

Each **AXEPTool** includes an instance of the AXMEDIS Database that allows distinguishing content in the AXMEDIS Factory and content published to be distributed and shared with other business partners. It also allows searching for content among business partners connected on the AXMEDIS Network. Typically the network allows sharing content among producers, integrators, distributors, publishers, archives, etc. Among the interconnected AXEPTools it is possible to make distributed queries to search for content, and to automatically publish and acquire/update content from/to the business partners, etc. The Potentially Available Rights and the contact information are the instruments to start the negotiation of content acquisition. This mechanism makes the B2B content distribution fasters, simpler and more secure. Each transaction and trial is monitored by the AXEPTool and by the AXCS. During queries, the content and the technical metadata are certified thus avoiding trivial problems of many P2P architectures.

- AXEPTool: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2960
- Snapshots http://www.axmedis.org/tiki/tiki-index.php?page=AXEPTool
- AXMEDIS P2P Tools: AXEPTool and AXMEDIA tool

The AXMEDIS **Programme and Publication (P&P) tools** include a set of tools for users to define AXMEDIS publication programmes to distribute content from the interconnecting AXMEDIS Databases to various user specified distribution channels. It utilizes the AXMEDIS Framework including the AXCP area for content adaptation, the AXMEDIS database and the AXMEDIS Query support. The P&P tools allow users to schedule distribution times, manage requests to use content production (AXCP) rules for content adaptation and specify configured distribution channels. AXCP rules requested using the AXP&P requests production/adaptation of content based on distribution profiles, production time costs and the delivering time of the content to be distributed. The P&P tools provide a front end and server in terms of Web services to deliver content ready for distribution.

- http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=349
- http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Programme+and+Publication

The **Distributors** represent any kind of content distribution services (see Fig.3): Internet, satellite and terrestrial broadcast, mobile, towards: PC, STB, I-TV, mobiles, PDAs, etc.
Distributors may be also interested in having in their plant some components of the AXMEDIS factory such as an instance of the AXMEDIS Database to make queries, license editor to produce licenses, AXCP for content adaptation and/or to protect content in massive manner, for content production on demand, etc. In order cases, they can delegate these actions to other parties or to external web services. Some examples about the usage of AXMEDIS tools to set up solutions for content distribution are reported in the following. In some cases, the Distributors may be interested in establishing a connection with the AXMEDIS Certifier and Supervisor for obtaining reports about the rights exploitation or for getting statistical information.

- Examples of AXMEDIS Framework and tool Exploitation
- Distribution on Internet the TISCALI demonstrator
- Distribution on PDAs and mobiles the ILABS demonstrator
- Distribution via Satellite data broadcast the EUTELSAT Demonstrator
- Content Sharing among Archives the ANSC case
- and many others

Some Distributors may be interested in exploiting AXMEDIS technology to set up a legal P2P service for content distribution. This solution can be realized by using the so called **AXMEDIA** tools. It is a P2P tool for distributing and sharing AXMEDIS content among end users, the distributor may insert AXMEDIS content in the network of peers and this may freely navigate among them but with the supervision and control of AXMEDIS protection and monitoring models.

The **AXMEDIS players** are based on the AXMEDIS object model and manager called AXOM (AXMEDIS Object Manager). They are capable of reading and playing/executing AXMEDIS objects according to the business models chosen and the license associated with the user/device. AXMEDIS player are available mainly for PC (as independent tools, as plug in for Internet Explorer and Mozilla browsers) and PDA with

Pocket PC 2003. The AXOM module can be integrated in any other content processing tool in order to manipulate AXMEDIS and/or MPEG-21 objects.

- Tools to be installed: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2885
- Players: http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Player
- http://www.axmedis.org/tiki/tiki-index.php?page_ref_id=348

AXMEDIS framework provides tool kits and libraries to create a large number of different players on different platforms, leaving free the customization of the user interface, skin, and much more; mainly MS-Windows, MAC and Linux, for PC, PDA and may be for mobiles and Set Top Boxes.

7 AXMEDIS Protection and Supervising Tools

AXMEDIS Protection and Supervision Tools provide support for registering and certifying users, providing unique IDs for the AXMEDIS objects, authenticating of devices and tools, processing licenses, managing black lists, continuous monitoring of the user activities on the AXMEDIS content on AXMEDIS players and tools on the basis of licenses, and collecting and reporting the information about content usage and rights exploitation, etc. The tools of this area are described in the following.

The **AXMEDIS Certifier and Supervisor, AXCS,** is the responsible of user registration, for device and tools authentication and certification, and for the registration and tracking of the activities performed on AXMEDIS objects on any AXMEDIS compliant tool. The AXCS provides protection information and share with the AXMEDIS Protection Manager Supports (PMS) the responsibility of managing the protection for distribution channels and domains. The AXCS also manages black lists of users, devices and tools to restrict their activities when irregularities are detected.

- AXMEDIS Certification and Supervision Control about the rights usage

The **AXMEDIS User Registration portal** is a service that can be used by content Distributors to make the registration of AXMEDIS final users (if they do not prefer to automatically register all their users). In any case, the User has to make a registration on AXMEDIS to obtain a certificate that could allow him/her to install tools and to cope with the related AXMEDIS licenses. Some Distributors may prefer to use a direct Web Service to automatically register their users in for AXMEDIS. Both solutions can be used.

The **AXMEDIS Object Registration service** to produce and assign a unique object ID to AXMEDIS objects. In the process, the most important metadata may be provided as well by allowing establishing relationship from standard and/or proprietary identification codes with those used in AXMEDIS. It is a service accessible by all tools capable of creating new AXMEDIS objects for any AXMEDIS Factory (such as: AXMEDIS Editor, AXCP GRID Node) and it is a WEB service directly connected to the AXMEDIS Certifier and Supervisor.

The AXMEDIS **Protection Manager Support**, **PMS,** collects the licenses and has the duty of processing chains of licenses on the basis of the requests received from AXMEDIS players, and all other AXMEDIS tools that include an AXOM to manipulate objects. The PMS allows the management of licenses and the sharing of these along other PMSs by means of a network of AXCSs. Each PMS can be associated with one or more different distribution channels or can be geographically distributed, e.g., to cover a geographic area. The definition of licenses and the management of information into PMS and AXCS allow to set up of a large variety of different distribution and transactions models, from client server to P2P, from satellite data broadcast toward i-TV to content distribution to cellular phones. The PMS is also provided in versions that allow managing Domains (the so called PMS Domain) such as those that can be set up for managing licenses for a school, at home, or a company, etc. Each PMS Server allows receiving the posting of new licenses by means of a Web Service, which can be used for automating their production from the Distributor Front End Sales server. Alternatively, the **AXMEDIS DRM Editor** (license editor) can be used in order to associate licenses and potential available rights (PAR) to the AXMEDIS objects during their creation in the AXMEDIS Editor.

The **AXMEDIS Portal** includes services for all the AXMEDIS users including those that support AXMEDIS and contribute to the construction and improvement of the AXMEDIS framework. It provides a set of services including the database of AXMEDIS documentation, the deployment of the AXMEDIS framework, the management of the mailing lists, etc. It provides tools updates and information to AXMEDIS partners. On the AXMEDIS portal, you can get the list of AXMEDIS compliant tools, devices, registered companies, test cases, documentation, libraries, etc. In addition, you can find references to the AXMEDIS services that allow the authentication, certification and continuous monitoring and control of any AXMEDIS tools.

- http://www.axmedis.org

8 AXMEDIS Framework

The AXMEDIS Framework is the set of information and tools that is at the basis of the above mentioned applications and solutions. Figure 4 shows a coarse view of the AXMEDIS Framework structure. It contains all the necessary tools to set up a large set of services and solutions in the area of content production, protection and distribution. The AXMEDIS Framework is an infrastructure on which several other models for content modeling, protection, production, DRM and distribution can be built in a very simple manner reusing the components and functionalities provided.

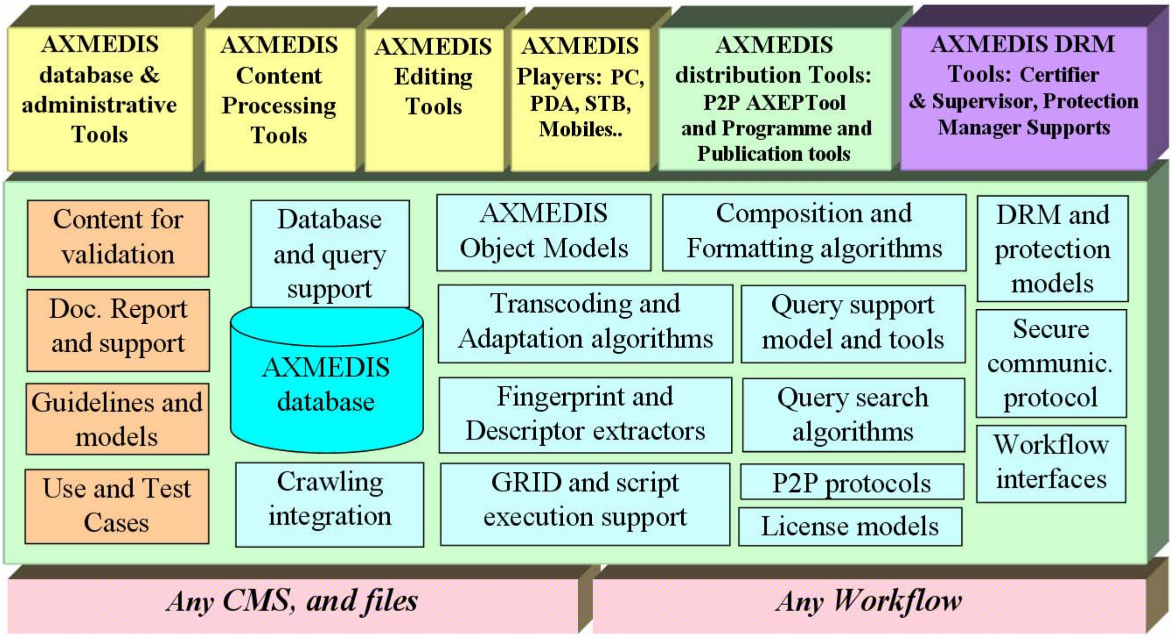

AXMEDIS database & administrative Tools
AXMEDIS Content Processing Tools
AXMEDIS Editing Tools
AXMEDIS Players: PC, PDA, STB, Mobiles..
AXMEDIS distribution Tools: P2P AXEPTool and Programme and Publication tools
AXMEDIS DRM Tools: Certifier & Supervisor, Protection Manager Supports
Content for validation
Doc. Report and support
Guidelines and models
Use and Test Cases
Database and query support
AXMEDIS database
Crawling integration
AXMEDIS Object Models
Composition and Formatting algorithms
Transcoding and Adaptation algorithms
Query support model and tools
Fingerprint and Descriptor extractors
Query search algorithms
GRID and script execution support
P2P protocols
License models
DRM and protection models
Secure communic. protocol
Workflow interfaces
Any CMS, and files
Any Workflow

Fig.4 -- AXMEDIS Framework structure, a simplified view

The general infrastructure gives a common ground on which other content based applications and tools can be built by small and large industries deciding to adopt the AXMEDIS framework for their business and by research institutions. In addition to the modules and tools described before, the most relevant parts of the **AXMEDIS Framework** are:

- requirements for many innovative solutions and their revisions,
- test cases and uses cases and their revisions,
- content for validations of tools and solutions, both single resources and metadata and demonstrative AXMEDIS objects. They range from collection of resources to full cross media content with templates and styles,
- general documentation of AXMEDIS tools and supports, including the:
    - whole specification of the AXMEDIS framework and the
    - detailed technical documentation of the source code,
    - user and installation manuals, technical references, etc.
- CVS/SVN tree with source codes of the several modules of the AXMEDIS framework, including libraries, applications and tools,
- examples of AXCP scripts modeling algorithms for content compositions and formatting, for transcoding and adaptation, for extraction of fingerprint and descriptors, content processing, license manipulation and verification, license adaptation, etc., for many different formats of digital resources and for any categories of them: audio, video, document, multimedia, images, animations, text, metadata, etc.,
- examples of content, from single content element, to collection, advertising integration, cross media, with SMIL HTML, etc.
- examples and models for licenses (MPEG-21 REL and OMA),

- example and models for protection information,
- examples of workflow usage and programming for controlling AXMEDIS Factories,
- examples of queries and selections for accessing to the database,
- tutorials about AXMEDIS framework exploitation and tool usage:
    - on general aspects and on state of the art of the sector,
    - on AXMEDIS framework, general aspects,
    - on automated and manual content production,
    - on automated and manual content protection and DRM (Digital Rights Management),
    - on AXMEDIS tools,
    - on multichannel distribution tools, etc.,
- guidelines for source code production for contributing to the AXMEDIS framework,
- guidelines on content production and distribution,
- guidelines for the production of AXMEDIS Plug-ins for AXCP GRID and AXMEDIS Editors,
- guidelines for the production of licenses on the basis of contracts,
- ready to use/install AXMEDIS tools such as: AXMEDIS Players, AXEPTool P2P, AXTraker, AXMEDIS Query Support, AXMEDIA P2P tool, AXMEDIS Editors, AXMEDIS Programme and Publication tools, AXMEDIS Content Processing Tools, AXCS (Certifier and Supervisor), AXMEDIS PMS.

A more detailed technical view of the AXMEDIS Framework is reported in the following figure.

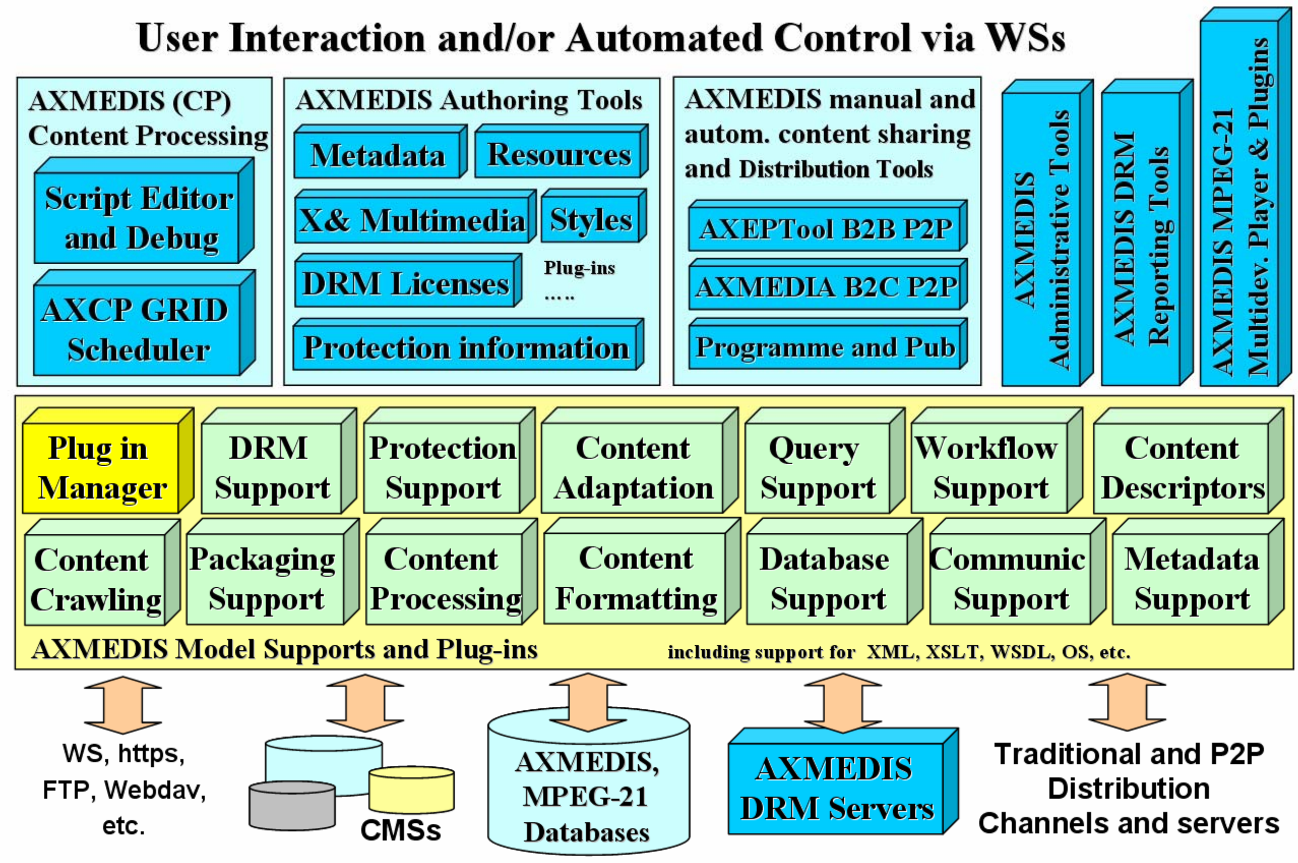


Fig.5 -- AXMEDIS Framework technical view

8.1 Accessing to the AXMEDIS Framework

The present status of the AXMEDIS Framework can be obtained from its coordinator or partners. Demonstrations of the AXMEDIS tools and of the whole AXMEDIS Framework are provided at AXMEDIS conferences and in other occasions listed on the AXMEDIS Portal. The AXMEDIS Framework can be accessed by all affiliated partners. The Affiliation to AXMEDIS is performed by subscribing an Affiliation Agreement with an AXMEDIS Contractor. The Affiliation Agreement and the list of Contractors are accessible on the AXMEDIS portal.

- http://www.axmedis.org/affiliation.php
- Affiliation Agreement: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1751

There are many **reasons to get affiliated to AXMEDIS,** which can be summarized as follows:

- Obtaining access to a royalty free and *open platform* that can be customized for your production., protection and distribution needs;
- *Reduction of costs* for content gathering, processing, production, protection and distribution;
- Adopting a standard model (MPEG-21, OMA) for content and licenses modeling and thus for inserting DRM in your business;
- Establishing contacts with other business partners interested in exploiting similar technology;
- Acquiring a greater control about content usage;
- Creating customized Cross media players for PC, PDA and mobiles, playing MPEG-21, MPEG-4, HTML, SMIL, etc., any format;
- Exploiting and trial of new business models;
- Joining the P2P B2B Network of AXMEDIS;
- Exploiting capabilities of secure legal P2P distribution;
- Setting up and creating a customized distribution channel interoperable with others;
- Setting up some new service (empowering your present solution) on the basis of AXMEDIS technology;
- Setting up a one-stop service for content protection and DRM set up;
- Allowing reporting to your business customers which rights are exploited on their content;
- Allowing the management of rights reporting for multimedia products;
- Allowing to use a solution that can be safer and more flexible with respect to state of the art;
- Saving money by using innovative technologies for content production and distribution, integrated environment;
- Accessing extremely innovative technology to give it a trial;
- Contributing to the AXMEDIS Framework gives you continuous access to the framework at reduced costs.

Research institutions and technology providers are interested in getting affiliated with AXMEDIS to:

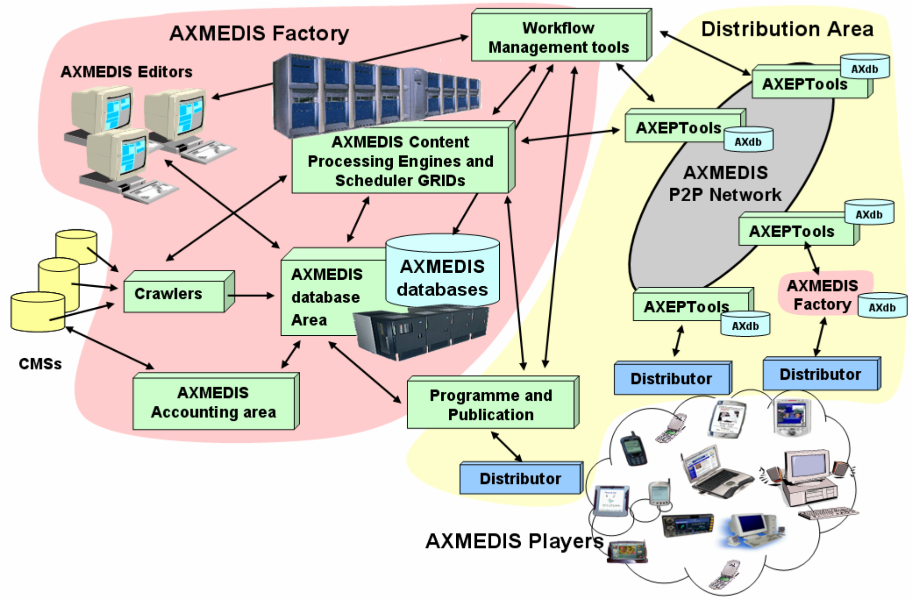
- make visible and promote their algorithms and tools for content processing and modeling that can be somehow integrated into the AXFW. These tools may be provided as demonstrators with limited capabilities;
- exploit the AXMEDIS Framework to make business with it for the reasons reported in the above list;
- add new content models and new DRM models and make them interoperable with MPEG-21 and others already in place on AXMEDIS;
- test new algorithms and tools with respect to the state of the art solutions, in a very easy and cheep manner;
- access at low cost the AXMEDIS framework by means of which several different configurations and solutions may be built to cover the needs of the value chain actors and tested with low effort;
- access tools based on MPEG-21 standard;
- collaborate with very relevant and well known research institution and companies of the areas;
- etc.

The present **status of the AXMEDIS Framework** can be obtained from the AXMEDIS project coordinator. Demonstrations of the AXMEDIS tools and framework are provided at AXMEDIS conferences and in other occasions listed on the AXMEDIS Portal. The AXMEDIS Framework can be accessed by Affiliated Partners. The Affiliation to AXMEDIS may be performed by subscribing an Affiliation Agreement with an AXMEDIS Contractor.

The AXMEDIS 2007 conference will be held in Barcelona in November 2007. The Call for papers is open until 15 May 2007. http://www.axmedis.org/axmedis2007/

9 AXMEDIS Content Production

The Content Production area of AXMEDIS is mainly focussed on what it is called AXMEDIS Factory automating: (i) the packaging containing the digital resources (the real content), (ii) adapting and transcoding, (iii) protecting content and producing corresponding Prot-Info, (iv) publishing and distributing the produced package, (v) producing licenses for the users, etc.



Content Object Production

- Existing contents (e.g., resources and metadata located in databases, CMSs, file systems) are crawled and collected using automated processes and rules (e.g., Crawler, rule editing);
- Crawled contents or new contents can be inserted into the AXMEDIS Database;
- Content can be automatically packaged in AXMEDIS model (which is MPEG-21 compliant);
- Content production and elements (metadata, resources, protection information and licenses) can be processed manually with authoring tools and editor or automatically with AXMEDIS Content Processing tools based on executable rules that can be hosted on single or massive GRID of computers;
- Authoring tools can be used to insert/revise metadata, define protection information, define DRM licenses models, to modify content or simply to view the objects;
- Various processing are offered (adaptation, transcoding, protection, etc.), either automatically or manually (using GUI editor);
- The various components and digital resources can be glued together by means of SMIL based templates and style that may be used to define the usage interface (format) of the whole object: karaoke, collection, menus, sliding presentation, buttons, live, animations, etc.;
- Results can be sorted in a database or on file systems or published in towards distributors or on the B2B P2P AXMEDIS network;
- They are now available on the AXMEDIS network, for further aggregation, distribution, etc., to be searched, modified, or shared;
- Queries and P2P allow retrieving content located in all the connected AXMEDIS Factories;
- The queries can be activated to automatically react at eventual changes in the sources, and thus to perform an automatic production/update;
- Queries can be performed on the basis of classification and identification metadata, but also on technical features, descriptors, licensing information (PAR), etc.;

- Once identified the objects or queries, they can be used as input parameter of processing rules for the GRID AXMEDIS Content Processing;
- All these activities can be governed by Workflow Management Tools for defining process production flow and information of the content factory and among different factories.

9.1 AXMEDIS Content Processing Capabilities

The AXMEDIS Content Processing Area has been designed to provide a set of digital content processing tools to aid content designers to create rules/script for **automating** content production and processing and in more details for:

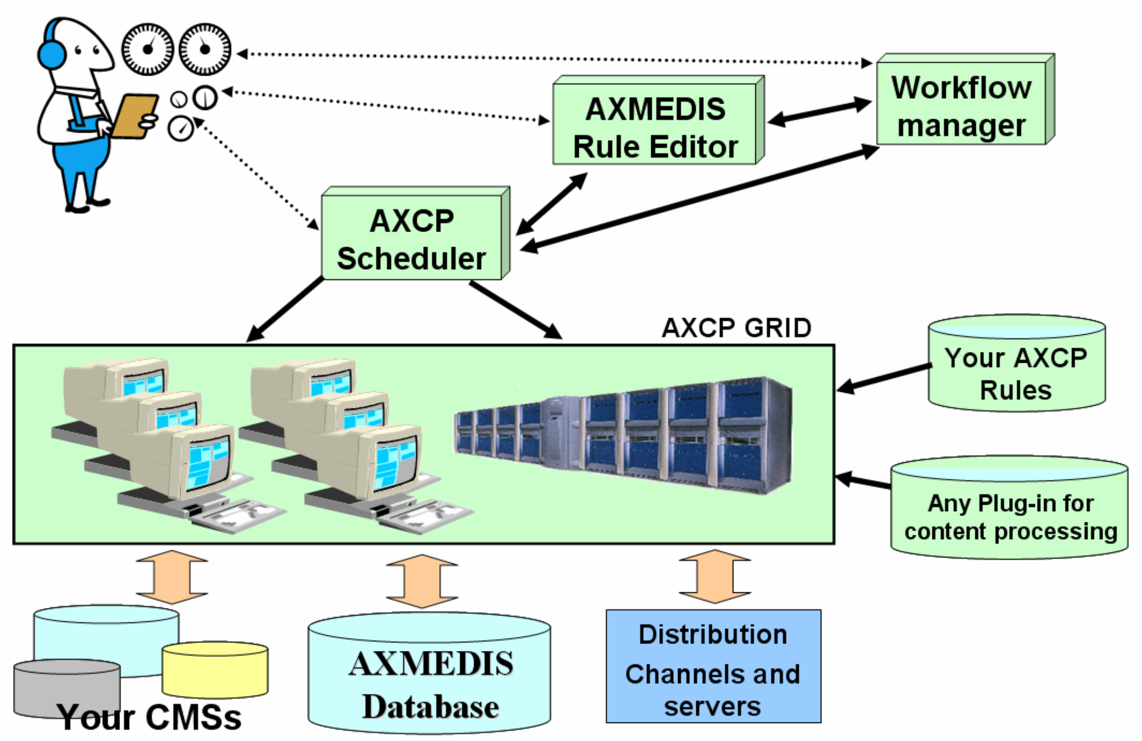
- **Content Ingestion and Gathering** from
    - Content Management Systems, from file system, or protocols;
    - processing resources and coupling them with metadata;
- **Content Retrieval** from
    - AXMEDIS database;
    - other AXMEDIS content Factories by means of the P2P tools, namely AXEPTool;
    - Content Management Systems, CSMs;
- **Content Storage** into
    - an AXMEDIS database;
    - ODBC based databases, other databases via Web Services, and other models will added later;
- **Content Processing** such as
    - digital resources adaptation, extraction of descriptors, transcoding, synchronisation, metadata processing, estimation of fingerprint, watermarking, indexing, content summarization, etc.;
    - metadata manipulation, mapping and adaptation;
- **Content Composition** for
    - creation of content components or objects as a combinations of raw assets such as Text, Images, Audio, Video (actual shot), Animation (synthetic), metadata, descriptors, licenses, multimedia objects such as MPEG-4, HTML, SCORM, macromedia tool file, animations, games, etc.;
    - creation of content as linear or hierarchical combination of content components;
- **Content Formatting**
    - gluing content elements together by means of SMIL based templates and applying style sheets to define the usage interface (format, layout) of the whole collection of content elements and the interested content usage paradigms (leaving open some parameters). For example, karaoke, collection browsing, selection menus, sliding presentation, stable background with a window with live video, animated text moving on an image, running text, etc.;
    - optimization of styling parameters left open or defining them manually to arrange for example: best fitting of images in the screen, optimizing the amount of text in the page, best time fitting, etc.;
- **Content Protection** such as:
    - protection of digital resources and full objects with their complex structure;
    - creation of Protection Information parameters, such as keys, or other features;
    - applying Protection Information model to AXMEDIS object, segmenting digital resources, slicing objects, applying encryption, scrambling, compression, and many other algorithms;
    - posting specific protection information for each given object to the AXMEDIS Certifier and Supervisor server;
- **Content Licensing** for
    - generating license from license model and additional information, storing licenses, posting them on license server automatically;
    - transcoding/translating licenses;

  - invocation of some verification algorithms about licenses and available rights to simulate the usage from the user site;
- **Content Publication and Distribution** towards
  - any distribution channel, producing programme and its schedule;
  - P2P network of other AXMEDIS Factories of content integrators, producers, and distributors.

9.2 AXMEDIS Content Processing Tools

In order to exploit the above list of features to process/manipulate content, resources, licenses, XML, SMIL, databases, protection information, etc., the AXMEDIS Content Processing Area is governed by a set of tools:

- **AXMEDIS Rule Editor**: to produce, debug, test and validate the executable AXCP Rules that can be:
  - written with a simple **AXCP language for content production** which is an extension of Java script;
  - created as macros from AXMEDIS Editor and authoring tools;
  - tested, debugged and validated on the AXMEDIS Rule Editor;
  - activated for content processing on any AXCP GRID Node or on a single computer;
  - used for B2B or B2C purpose;
  - used/parameterized for producing content on demand or to be integrated in your content factory;
  - activated from your Workflow Manager engine via web service;
  - activated by changes in remote objects and queries in the local database and on the P2P network.
- **AXCP GRID**: a set of general purpose or specialized computers to execute AXCP Rules governed by the AXCP Rule Scheduler;
- **AXCP GRID Node**: a single general purpose or specialized computer of the AXCP GRID;
- **AXCP Scheduler**: to schedule the AXCP rule of GRID nodes according to the content production and processing needs in terms of time and resources.



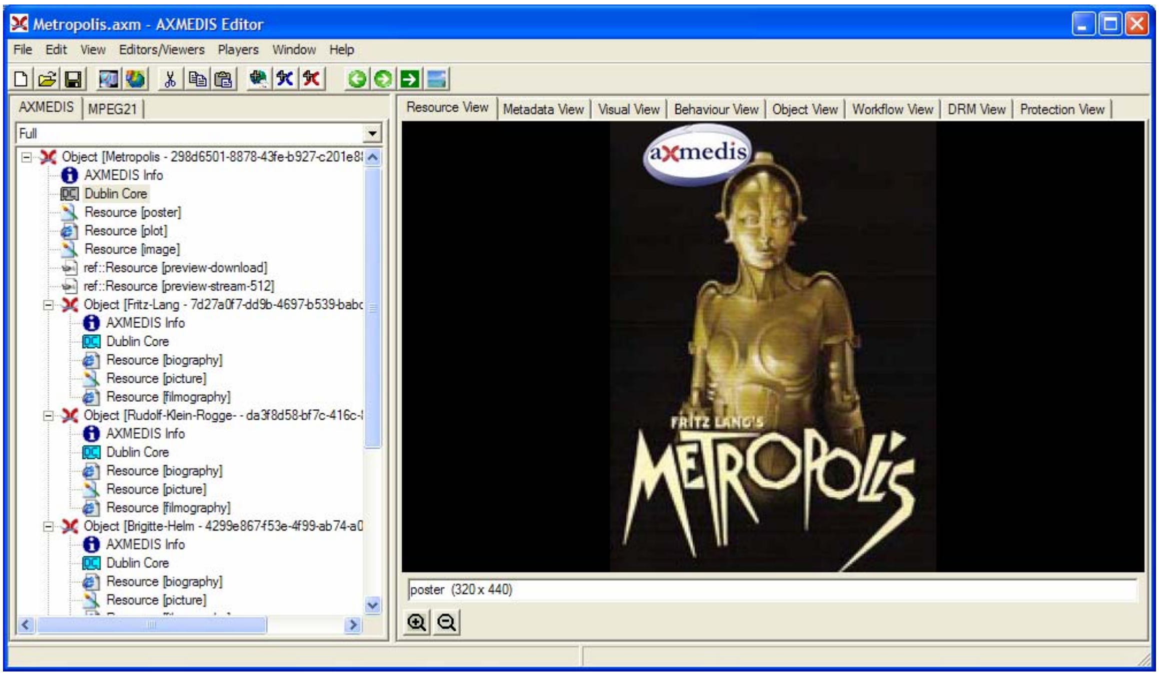
The processing capabilities and functionalities reported in the previous section and that can be exploited from AXCP Tools and Rules can be simply expanded by means of realizing or installing a set of additional plug-ins. The AXMEDIS Plug-in technology is open since:

- the specification of plug-in format is public;
- a plug-in tool kit (including examples and source code for creating those plug in) are public and accessible to all;
- a large part of the above mentioned features are provided by means of AXMEDIS Plug-ins. This demonstrate the solution flexibility;
- any user or third party company can create its own plug-in or include in a plug-in any open third party library including those open source.

9.3 AXMEDIS Editors, the authoring tools

AXMEDIS Editor is used for the manual production of AXMEDIS Objects, and allows creating and manipulating object features and their different aspects:

- ***Structure***, to add, remove, move digital resources and metadata inside the AXMEDIS Object (hierarchy editor and viewer of the object);
- ***Resource manipulation***, to use content processing plug-ins for generic and customizable resource manipulation (the same algorithms and tools used in the AXMEDIS Content Processing);
- ***Metadata editing***, allowing to edit/mapping the metadata related to digital resources and objects (metadata editor and viewer);
- ***Visual and behavioural editing,*** allowing to define content usage paradigm with SMIL organization of resources to present/layouting the digital content contained inside the AXMEDIS Object (visual and behavioural editor and viewer);
- ***Protection editing,*** allowing to specify the protection algorithms to be used for the AXMEDIS Object protection, and thus to define the Protection Information (Protection Editor and Viewer);
- ***DRM editing***, allowing to produce and verify licenses for end users and/or distributors of the AXMEDIS Object as well as the Potentially Available Rights, PAR, that could be acquired on objects shared in the P2P Network (DRM editor and viewer);
- ***Workflow***, allowing editing and viewing the status and the work to be done on the AXMEDIS Objects involved in the workflow process (workflow editor and viewer).



Moreover, **AXMEDIS Editor:**

- allows to perform queries to look for content to produce AXMEDIS Objects integrating other AXMEDIS objects coming from the internal AXMEDIS Database, the AXMEDIS P2P network or possibly from the factory CMS;

- allows download/upload AXMEDIS Objects from/to the AXMEDIS Database and file system;
- can be controlled by the AXMEDIS workflow system to integrate manual operations on AXMEDIS objects inside the production process formalized with the workflow system;
- can be used to inspect automatically generated objects for validation;
- can finalize the production of automatically produced objects;
- can produce rules/script to be used as templates for the AXMEDIS Content Processing environment.

9.4 Content Adaptation facilities

AXMEDIS objects can be created to be distributed over heterogeneous networks and towards different kind of terminals, client tools/devices. Moreover, the people who will ultimately consume and interact with the content may have different behavior and preferences, and the best formats to provide them the best experience on their terminal could be different. Consequently, digital items may need to be adapted to fit any particular usage environment. This is the goal of AXMEDIS content adaptation tools which aim at achieving interoperable transparent access to (distributed) advanced multimedia content by shielding Users from network and terminal installation, management and implementation issues.

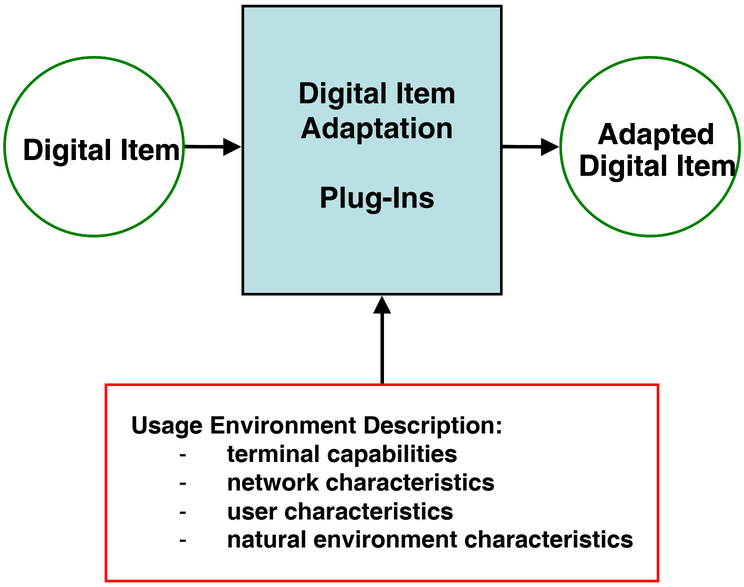
Adaptation may involve:

- transcoding of digital resources that means to change the format (for example, from TIF to GIF, from MPEG-4 to video, from a video to MPEG-4, from MPEG-4 video to a MPEG-2, from audio to symbolic music, from audio to MIDI, from audio in PCM to a ringtone format for mobiles, etc.), re-sampling, shrinking, stretching; In some cases, some features related to resolution, interactivity, are lost in the process in favor of having the content usable in another format. Typically the content is produced in a format that is transcoded scaling down its features and not the vice versa;
- manipulation of licenses, reduction of license scope, reduction of time or territorial information, transformation of format, translation of license (such as passing from ODRL to MPEG-21 REL), etc.;
- manipulation of metadata, metadata mapping, metadata reduction, translation of metadata, etc.

The **adaptation process** can be performed:

- during the content production (for digital resources, licenses and metadata) by exploiting functionalities accessible from the AXMEDIS Editors and/or for the AXMEDIS Content Processing tools
- directly on the player terminal/device (mainly for digital resources and metadata). In this case, ISO/IEC 21000 (MPEG-21) specified a set of normalized tools for the adaptation of digital content describing the usage environment of a digital item to command adaptation tools. Within the AXMEDIS players, MPEG-21 usage environment descriptions are used to drive the adaptation tools considering:
  - Terminal capabilities (codec, formats, input-output, etc., supported by the terminal),
  - Network characteristics (for example, the minimum guaranteed bandwidth of a network),
  - User characteristics (presentation preferences, auditory or visual impairment etc.),
  - Natural environment characteristics (for example, the illumination characteristics that may affect the perceived display of visual information).

The conceptual architecture of the adaptation engine is shown below: a digital item is subject to adaptation thanks to dedicated plug-ins to produce the adapted digital item; the adaptation performed by the plug-ins is parameterized according to MPEG-21 usage environment descriptions.

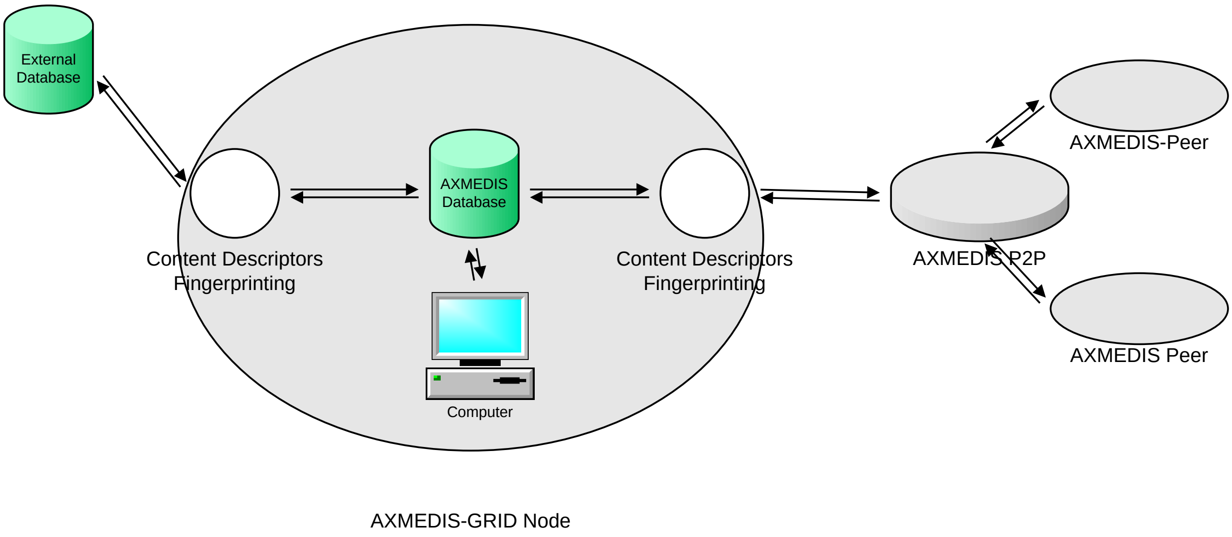


9.5 Content Fingerprint and Descriptors Extraction

Among the several content processing algorithms that can be applied in AXMEDIS, a large set can be classified as extractors or estimators of Fingerprint and/or Descriptors of content.

Content fingerprints and descriptors can be used in the AXMEDIS framework for different purposes:

- Classification and recognition of content and/or digital resources;
- Identification of a single piece of content/resource and thus for content certification;
- Authentication/verification of content integrity.



They can be of classified as:

- **High-level descriptors** to describe content with a set of high-level features independent on the format and content resolution. They are typically high level features immediately related with concepts understandable by humans, such as: rhythm and tonality for music, subject for text, etc. They can be very easily used as Descriptors by humans to make queries in the data bases, and thus to retrieve similar content;
- **Digital fingerprints** (or **perceptual hash values**). They can be compared with human fingerprints and may be used to identify a specific content and are robust against data transformation;

- **Low-level descriptors** to describe digital information at lower level and sometime these descriptors are not independent on the format and resolution, such as: energy for music, spectra for images, dynamic, duration, etc. Similar content may share similar low-level descriptors. In some cases, they are used at the basis to estimate high-level descriptors. They can be used for recognition of content as well;
- **Low-level fingerprints** to estimate a value from a specific digital resource, in many cases estimated with algorithms that do not take into account the content type, such as: cryptographic hash value that can be estimated for any digital file. They can be used for content and digital resource verification of consistency and authentication.

In AXMEDIS, Content Fingerprints and Descriptors can be:

- manually selected and estimated via the AXMEDIS Editors;
- automatically estimated and stored in the object metadata or other places by means of an AXMEDIS Content Processing Rule script. For example: during the acquisition/crawling of content from CMS, during content composition and/or formatting, during any content processing also included in the production on demand;
- estimated to verify integrity of content when the content is opened by an AXMEDIS Editor, or by an AXMEDIS Player, or processed by a AXMEDIS P2P tool such as AXEPTool or AXMEDIA, etc., that is any time that an AXMEDIS object is loaded into the AXOM core component of AXMEDIS Framework.

9.6 AXMEDIS Protection Tools

In AXMEDIS, the content protection can be performed automatically (in the AXCP Rules) or manually (with AXMEDIS Editor, Protection Editor) on any kind of content, from simple digital resources to complex AXMEDIS objects that may contain nested and hierarchies of other content and/or resources and metadata.

The content protection consists of:

- changing the digital coding of content for example by using algorithms such as encryption, compression, scrambling, etc. Any other protection tool can be added by using AXMEDIS plug-in technology;
- applying the algorithms mentioned at the previous point with some parameters (such as keys) and additional information (such as the algorithm type, the size of segments, the order in which the algorithms are applied, etc.). These data are the Protection Information that are need to unprotect the objects, and are delivered to the Players only when the needed authorisation is provided on the basis of the license. AXMEDIS Protection Information is an extension of MPEG-21 IPMP model.

Thus the protection model of AXMEDIS is based on the combination of protection techniques together with the use of DRM technology. In this sense, the Protection Information and the License are provided to the AXMEDIS players following different paths and in different moments. This allows setting up and managing a large number of solutions for content distribution based on different transaction and business models.

10 AXMEDIS plugin and External Libraries

The AXMEDIS Tools allows including the functionality of external libraries through an advanced plug-in interface. Through this plug-in interface different external libraries are already included. The included plug-ins can be grouped in different categories:

- **Content adaptation** tools and algorithms are the algorithms that allow automatic content processing. The input and the output content type are the same. E.g. typical applications are scenarios where content is adapted according to the needs of the receiver.
- **Content description** tools and algorithms automatically extract low or high level descriptors from content. AXMEDIS supports the overall range of the content descriptors through is general description.
- **Content fingerprinting** tools and algorithms are a subset of low level descriptors. Their purpose is the identification and authentication of content.
- **External processing** algorithms are external algorithms not belonging to the above group that are so far considered to be integrated within the AXMEDIS framework.

10.1 AXMEDIS Plug in purposes and usages

The AXMEDIS plug-in plug-in mechanism has the goal to enable an easy enrichment of the AXMEDIS framework with additional functionalities. In fact, using the plug-in technology, the AXOM, the basic library in the AXMEDIS framework, can be enriched with new functionalities in order to manipulate and protect AXMEDIS Objects. The two main application fields currently supported are the creation of new Protection Tools and Content Processing Functions.

Plug-ins are loaded, independently by their category, by the Plug-in Manager. It organizes them on the basis of their category allowing specific modules easily retrieving them. These modules (such as Content Processing and Protection Processor) provide specific functions to invoke functionalities provided by the loaded plug-ins.

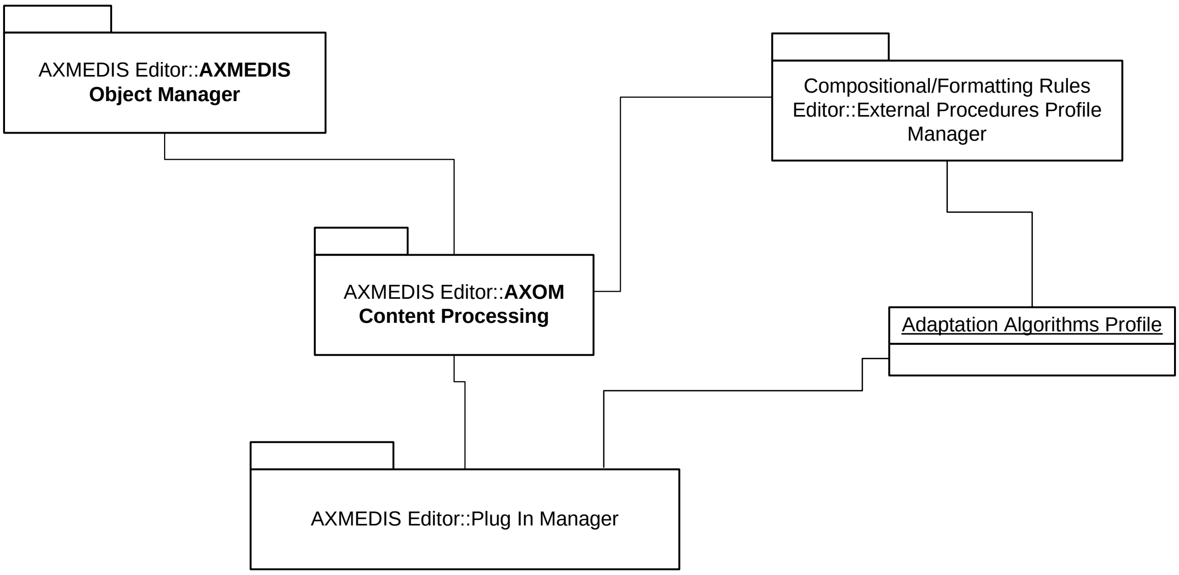
Within the AXMEDIS Tools different workflow processed and content processing operations are supported. To increase the flexibility and therefore the potential of the AXMEDIS framework, external libraries can be integrated into the different workflow processed and content processing operations.

10.2 AXMEDIS Plug-in Model

Within the AXMEDIS Framework, the AXMEDIS Plug-in Manager supports the installation/registration of plug-ins, loading of plug-ins. AXMEDIS Editor Plug-in Manager is able to manage different kinds of plug-ins, including:

- Data-manipulation plug-ins
- Metadata management and production
- Workflow plug-ins
- Protection plug-ins

As shown in the next figure, the AXMEDIS Plug-In Manager is accessed via the AXOM. Thus, the newly integrated functionalities are available within the different AXMEDIS Tools. For the management of the plug-ins, a description of the functionality has to be provided. This is the so-called algorithm profile.


AXMEDIS Editor::**AXMEDIS Object Manager**
Compositional/Formatting Rules Editor::External Procedures Profile Manager
AXMEDIS Editor::**AXOM Content Processing**
Adaptation Algorithms Profile
AXMEDIS Editor::Plug In Manager

The Plug-in Manager provides standard interface for the integration of the algorithms, which includes a plug-in profile. The plug-in profile describes the available functionality. Each plug-in is described by an XML file which contains the following information:

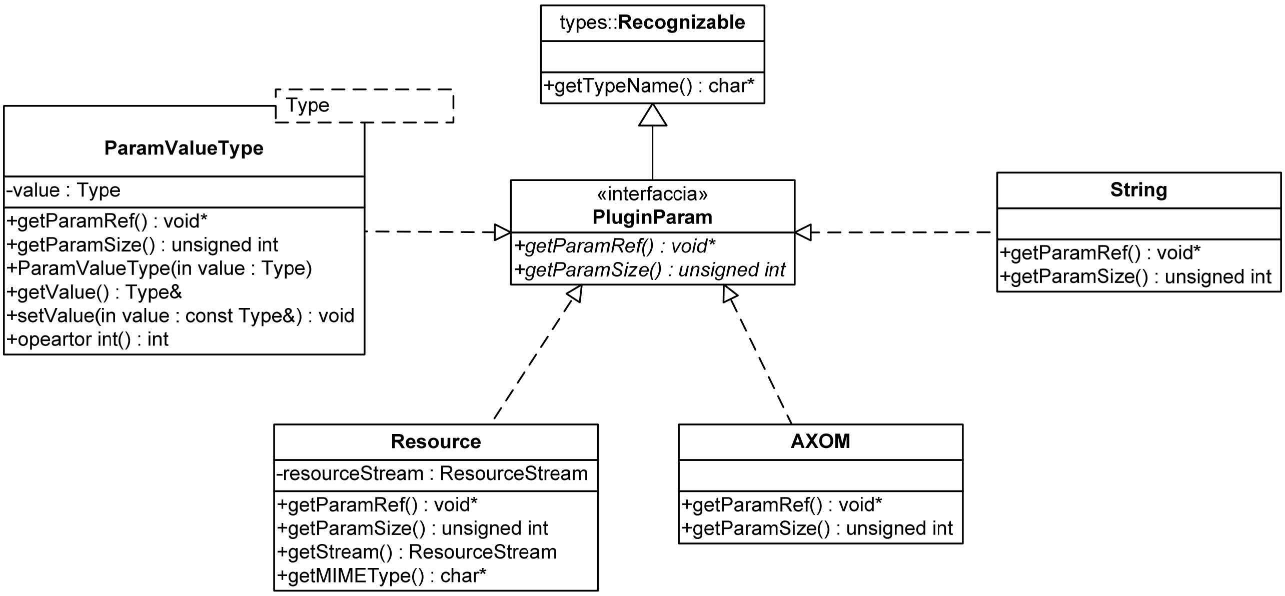
- the category of the plug-in, e.g. content processing;
- the unique identifier of the plug-ins, e.g. an URI like a XML namespace;
- the signature of the plug-in evaluated by an AXCS;
- data specific for the kind of plug-in (see below);
- the signature, estimated by the AXCS, of the whole XML file.

The profile provides a placeholder for custom information which can vary on the basis of plug-in category. The definition of general plug-in profile schema is reported in DE3.1.2.3.4. The category-dependant descriptions of the plug-ins are reported in DE3.1.2.3.4 (Content Processing) and in DE3.1.2.3.3 (Protection Tool).

10.3 AXMEDIS Plug in development

In order to develop a plug-in, a developer has simply to create dynamic libraries which respect given interfaces. These dynamic libraries are container of classes which implement a given interface (which interface has to be implemented depend on the category of the plug-in). A library simply acts as factory for the implemented classes. Once the dynamic library classes are generated the developer can easily create the profile of the plug-in on the basis of the implemented functionalities. The profile has to respect the schemas described in the specification.

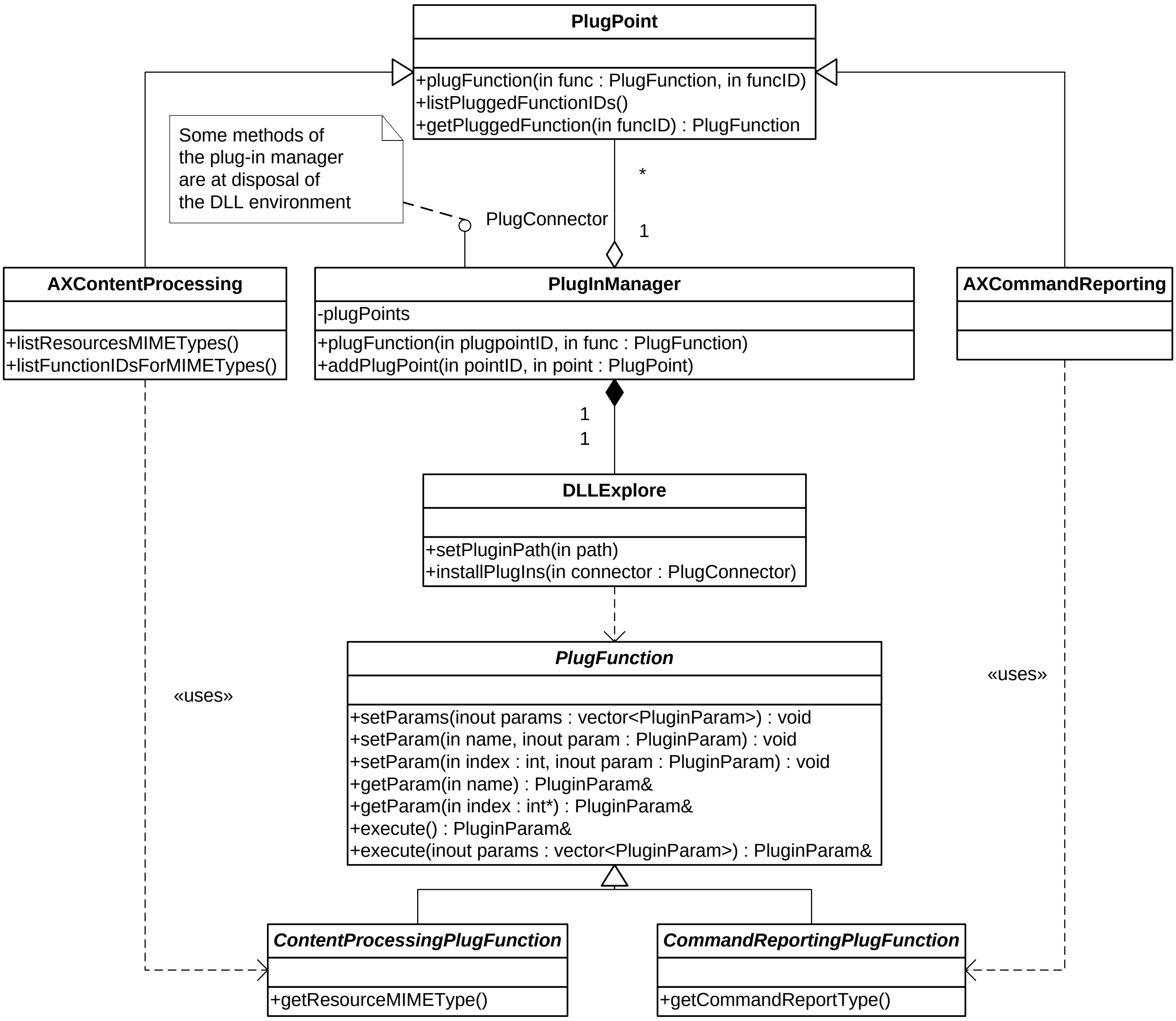
As stated in the description schema, for each parameter type a corresponding class is defined. This class has to be used within the programming language. All of these classes are derived from a common base class (**PluginParam**), which provides a basic interface for all possible parameter types. Moreover, PluginParam derives from Recognizable which is an interface that allows to perform type-related operations on the objects (e.g. something like reflection in Java and C#).



The associations among class and parameter types are depicted below:

- **ParamValueType** is a template class which will be used, through a set of *typedef* definition, to wrap the basic data type, that is: UINT16, INT16, UINT32, INT32, CHAR, BOOLEAN, FLOAT, DOUBLE.
  This class exposes a constructor and a cast operator which allow to freely use its instances in the same way as a base type.
- **String** represents a parameter of type STRING.
- **Resource** represents a RESOURCE. It exposes two meaningful functions: **getMIMEType**, **getStream**. The former allows knowing the MIME type of the resource contained in it, the latter returns a stream which allows retrieving the resource wherever it is physically located.
- **AXOM** represents a parameter of type AXOM.

The base class exposes two generic function **getParamRef** and **getParamSize** which return a reference to the real parameter (as void pointer) and the size (in bytes) of the parameter itself respectively.


PlugPoint
+plugFunction(in func : PlugFunction, in funcID)
+listPluggedFunctionIDs()
+getPluggedFunction(in funcID) : PlugFunction
Some methods of the plug-in manager are at disposal of the DLL environment
PlugConnector
*
1
AXContentProcessing
+listResourcesMIMETypes()
+listFunctionIDsForMIMETypes()
PlugInManager
-plugPoints
+plugFunction(in plugpointID, in func : PlugFunction)
+addPlugPoint(in pointID, in point : PlugPoint)
AXCommandReporting
1
1
DLLExplore
+setPluginPath(in path)
+installPlugIns(in connector : PlugConnector)
«uses»
PlugFunction
+setParams(inout params : vector<PluginParam>) : void
+setParam(in name, inout param : PluginParam) : void
+setParam(in index : int, inout param : PluginParam) : void
+getParam(in name) : PluginParam&
+getParam(in index : int*) : PluginParam&
+execute() : PluginParam&
+execute(inout params : vector<PluginParam>) : PluginParam&
«uses»
ContentProcessingPlugFunction
+getResourceMIMEType()
CommandReportingPlugFunction
+getCommandReportType()

In the above diagram the plug-in manager structure is depicted. That diagram provides base classes which allow the construction of more complicated architectures. In particular, fundamental elements of that approach are the following:

- **PluginManager** is the main class of the module which links the plug-ins (and their functionalities) to those components which will use them. It contains a list of **PlugPoint**, each of them is registered to a specific category of plug-in and function. Every time **PluginManager** loads a plug-in and its functions, it alerts the related **PlugPoint**.
- **DLLExplorer** is a utility class which is used by the **PluginManager** to discover all the plug-ins installed on a device. Every time it finds a dynamic library with its associated profile it invokes the **PluginManager**.
- **PlugFunction** is a base class for all those classes which have to be used as plug-in function. The **execute** method is pure virtual and it has to be defined in the derived classes, it is the real functional part of those classes.
- **PlugPoint** is a base class for all those classes which want to use plug-in functions. As depicted in the figure, PlugPoint is the base class from which the content processing module derives from (as well as the command and reporting module).

Plug-in certification functionalities are based on the services provided by the Protection Processor which needs to access the profile of the plug-ins.

10.4 AXMEDIS Plug in Content Processing and Protection Tools

10.4.1 Content Processing

Content Processing Plug-ins have to export the following functions:

```
extern "C" declspec(dllexport) AxCPFunction* GetPluginFunction(std::string funcName)
extern "C" void declspec(dllexport) releasePluginFunction(AxCPFunction* function)
```

`GetPluginFunction` allows the creation of instances of content processing functions by providing the function name (`funcName`).
`releasePluginFunction` releases the given `function` instance. This function has been introduced for the function instances that are created by the dynamic library and they have to be deleted by it.
`AxCPFunction` is the base class for content processing function classes. It provides some default methods for setting parameters and requires the implemention of the execution method of the function itself. Therefore the only method which has to be implemented in order to develop a Content Processing Function is `executed`.

An example of content processing plug-in can be found in the following folders in the repository:
https://cvs.axmedis.org/newrepos/Framework/source/adaptation/image/
https://cvs.axmedis.org/newrepos/Framework/bin/adaptation/image/
The first folder contains the source code of the dynamic library. The second one contains the XML profile of the plug-in. This and another example will be described in more detail in the following sections.

10.4.2 Protection Tools

Protection Tools used by the Protection Processor are distributed as AXMEDIS Plug-ins. Each plug-in contains a set of protection tools. Moreover, the manifest of the plug-in contains a specific description which reports the identifiers of the provided tools and their main features.
In particular, an AXMEDIS Plug-in containing some protection tools has to export the following functions:

```
extern "C" declspec(dllexport) Alg* createIPMPTool(const std::string& toolID, bool encoding)
extern "C" declspec(dllexport) void releaseIPMPTool(Alg* tool)
```

`createIPMPTool` allows creating Protection Tool instances by providing the tool identifier (`toolID`). When the `encoding` parameter is `true`, the function has to return the encoding version of the tool itself. Otherwise, the function has to return the decoding part of the required tool. That is, a tool identifier identifies the couple of encoding and decoding algorithm.
`realeaseIPMPTool` releases the given `tool` instance. This function has been introduced for the tool instances are created by the dynamic library and they have to be deleted by it.
`Alg` is an abstract class which is the base class of all Protection Tools. That is, each Protection Tool has to expose the interface provided by `Alg`. In particular, a given Protection Tool has to implement its own specific `BlockProcess` function which, given a block of bytes, has to produce the clear-text version of this block w.r.t. the implemented algorithm. In this way, different Protection Tools can be treated in a common manner.

https://cvs.axmedis.org/newrepos/Framework/source/encdecsup-plugin
https://cvs.axmedis.org/newrepos/Framework/bin/encdecsup-plugin

The first folder contains the source code of the dynamic library. The second one contains the XML profile of the plug-in. This and another example will be described in more detail in the following sections.

10.5 AXMEDIS Plug-in Examples

This section describes the development of plug-ins for the integration of existing tools. Besides calling the functionality that is available in external libraries, system functionalities (executables) can be called and additionally system-call function that are available in C/C++ can be used.

10.5.1 AXMEDIS Plug-in Examples: Developing a Content Processing Plug-In

10.5.1.1 Description and Use

This example, which describes the integration of a content processing plug-in, is based on the integration of functionalities that are available within the ImageMagick image processing library. The corresponding source code can be found in the following folders in the repository:

https://cvs.axmedis.org/newrepos/Framework/source/adaptation/image/
https://cvs.axmedis.org/newrepos/Framework/bin/adaptation/image/

The first folder contains the source code of the dynamic library. The second one contains the XML profile of the plug-in.

`AxCPFunction` is the base class for all content processing function classes. A specific class has to be defined for each processing functionality, which is derived from `AxCPFunction`.

***10.5.1.2** Implementation Details*

As described in the following, the constructor allocates memory for the input parameters. This is needed for the mapping of (content processing function) parameters to the corresponding class parameters. .

The function `execute()` contains the content processing functionality or calls the corresponding functionality.

The following extract of the plug-in source code shows the functionality for the function that manipulates the contrast of an image by calling the corresponding ImageMagick functionality.

```
//################################## Contrast #################################

class Contrast: public AxCPFunction
{
public:
	Contrast()
	{
		mapParameters["InputResource"] = new AxCPParameterResource(NULL,NULL,"");
		mapParameters["AMOUNT"] = new AxCPParameterINT32();
		mapParameters["OutputResource"] = new AxCPParameterResource(NULL,NULL,"");
	}
	AxCPParameter* execute()
	{
		AxCPParameter * result = new AxCPParameterSTRING();
		AxCPParameterResource* parInputResource
			= dynamic_cast<AxCPParameterResource*>(mapParameters["InputResource"]);
		AxCPParameterResource* parOutputResource
			= dynamic_cast<AxCPParameterResource*>(mapParameters["OutputResource"]);
		AxCPParameterINT32* paramAmount
			= dynamic_cast<AxCPParameterINT32*>(mapParameters["AMOUNT"]);
		std::istream& inputStream = parInputResource->getIstream();
		std::ostream& outputStream = parOutputResource->getOstream();
		unsigned int amount = paramAmount->getValue();
		std::string mimetype=parInputResource->getMimeType();

		AxImage image;
		if(image.loadFrom(inputStream,mimetype))
```

```
                {
                        image.Contrast(amount);

                        image.writeTo(outputStream,mimetype);

                        parOutputResource->setMimeType(mimetype);

                        result->setStrValue("SUCCESS");
                }
                else
                        result->setStrValue("ERROR - cannot load file");
                return result;
        }

};
```

As described in the previous section the following DLL - functions have to be implemented:

```
extern "C" declspec(dllexport) AxCPFunction* GetPluginFunction(std::string funcName)
extern "C" declspec(dllexport) void releasePluginFunction(AxCPFunction* function)
```

The function `GetPluginFunction` creates the required objects for the content processing functionality, The string parameter `funcName` identifies the selected algorithm (if multiple algorithms are available).

The following code extract exemplifies the implementation with the functions `Conversion`, `Import`, `Resize`, `Contrast`, and `Test`. It starts with the function header and some other functionalities.

```
extern "C" __declspec(dllexport) AxCPFunction * GetPluginFunction(string functionName)
 {
        AxCPFunction* Inst;
        if (functionName=="Conversion")
                Inst = new Conversion();
        else if(functionName=="Import")
                Inst = new Import();
        else if(functionName=="Resize")
                Inst = new Resize();
```

The following code extract is responsible for the management of the contrast function.

```
        else if(functionName=="Contrast")
                Inst = new Contrast();
```

The plug-in must as well verify that a correct parameter for the requested functionality is given.

```
        else if(functionName=="Test")
                Inst = new Test();
        else
                throw std::runtime_error(" REQUESTED FUNCTION IS NOT INTO PLUGIN ");

        return Inst;
    }
```

For completeness, the corresponding objects have to be freed. This is done by calling the function `releasePluginFunction`.

```
extern "C" __declspec(dllexport) void releasePluginFunction(AxCPFunction * pluginFunction)
 {
        delete pluginFunction;
 }
```

To finalize the plug-in development, an XML-description of the available functionality has to be described.

The corresponding XML file for the above described functionality and an additional dummy function is given below. It starts with the general header for each content processing plug-ing.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2005 rel. 3 U (http://www.altova.com)-->
<pin:Plugin xmlns:fd="http://www.axmedis.org/plugin-function-schema"
xmlns:pin="http://www.axmedis.org/plugin-schema"  xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.axmedis.org/plugin-function-schema plugin-function-schema.xsd
http://www.axmedis.org/plugin-schema plugin-schema.xsd">
        <pin:GeneralDescriptor>
                <pin:Category>ContentProcessing</pin:Category>
                <pin:Identifier>ImageProcessing</pin:Identifier>
                <pin:Library>Adaptation</pin:Library>
                <pin:Version>1.001</pin:Version>
                <pin:Vendor>Axmedis</pin:Vendor>
                <pin:MainLibrary>imageprocessingplugin.dll</pin:MainLibrary>
                <pin:Description>Plugin for image processing</pin:Description>
        </pin:GeneralDescriptor>
        <pin:ComponentsSignature>
                <dsig:SignedInfo>
                        <dsig:CanonicalizationMethod Algorithm=""></dsig:CanonicalizationMethod>
                        <dsig:SignatureMethod Algorithm=""></dsig:SignatureMethod>
                        <dsig:Reference>
                                <dsig:DigestMethod Algorithm=""></dsig:DigestMethod>
                                <dsig:DigestValue>AA==</dsig:DigestValue>
                        </dsig:Reference>
                </dsig:SignedInfo>
                <dsig:SignatureValue
xmlns="http://www.w3.org/2000/09/xmldsig#">AA==</dsig:SignatureValue>
        </pin:ComponentsSignature>
        <fd:FunctionList>
```

Below, the description of the available functionality is given.

```xml
                <fd:Function>
                        <fd:Name>Contrast</fd:Name>
                        <fd:Version>1.0</fd:Version>
                        <fd:FunctionDescription>Change image contrast</fd:FunctionDescription>
                        <fd:ParameterList>
                                <fd:Param>
                                        <fd:Name>InputResource</fd:Name>
                                        <fd:ParamType>RESOURCE</fd:ParamType>
                                        <fd:In/>
                                        <fd:Mandatory/>
                                        <fd:ParamDescription>The Resource to be
manipulated</fd:ParamDescription>
                                        <fd:Constraints>
                                                <fd:Resource>
                                                        <fd:ResourceType>image</fd:ResourceType>
                                                        <fd:ResourceFormat>jpeg</fd:ResourceFormat>
                                                        <fd:ResourceFormat>gif</fd:ResourceFormat>
                                                        <fd:ResourceFormat>png</fd:ResourceFormat>
                                                </fd:Resource>
                                        </fd:Constraints>
                                </fd:Param>
                                <fd:Param>
                                        <fd:Name>AMOUNT</fd:Name>
                                        <fd:ParamType>INT32</fd:ParamType>
                                        <fd:In/>
                                        <fd:Mandatory/>
                                        <fd:ParamDescription>The contrast amount</fd:ParamDescription>
                                </fd:Param>
                                <fd:Param>
                                        <fd:Name>OutputResource</fd:Name>
                                        <fd:ParamType>RESOURCE</fd:ParamType>
                                        <fd:Out/>
                                        <fd:Mandatory/>
                                        <fd:ParamDescription>Where the manipulated resource will be
stored</fd:ParamDescription>
                                </fd:Param>
                        </fd:ParameterList>
                        <fd:Result>
                                <fd:Name>Result</fd:Name>
                                <fd:ResultType>STRING</fd:ResultType>
                                <fd:ResultDescription>The result of conversion, SUCCESS if ok, ERROR 
followed by a message in case of error</fd:ResultDescription>
                        </fd:Result>
                </fd:Function>
```

The missing plug-in XML - description for the final description is shown below:

```
</fd:FunctionList>
        <pin:Signature>
                <dsig:SignedInfo>
                        <dsig:CanonicalizationMethod Algorithm=""></dsig:CanonicalizationMethod>
                        <dsig:SignatureMethod Algorithm=""></dsig:SignatureMethod>
                        <dsig:Reference>
                                <dsig:DigestMethod Algorithm=""></dsig:DigestMethod>
                                <dsig:DigestValue>AA==</dsig:DigestValue>
                        </dsig:Reference>
                </dsig:SignedInfo>
                <dsig:SignatureValue>AA==</dsig:SignatureValue>
        </pin:Signature>
</pin:Plugin>
```

10.5.1.3 Project Dependencies

The provided ZIP-file for the content processing plug in contain all AXMEDIS-files for a successful standalone compilation of the plug in. Additionally, some extra libraries are necessary:

- XERCES C++ Parser (tested with version 2.6.0)
  - http://xml.apache.org/xerces-c/
- wxWidgets (tested with version 2.4.2)
  - http://www.wxwidgets.org/

Furthermore, the plug in has dependencies to the following AXMEDIS components:

- axom
- common
- contentprocessing

10.5.2 AXMEDIS Plug-in Examples: Developing a Protection Tool Plug-In

10.5.2.1 Description and Use

This example for the integration of a protection tool plug-in is based on OpenSSL. An example of a protection tool plug-in can be found in the following folder in the repository:

https://cvs.axmedis.org/newrepos/Framework/source/encdecsup-plugin
https://cvs.axmedis.org/newrepos/Framework/bin/encdecsup-plugin

The first folder contains the source code of the dynamic library. The second one contains the XML profile of the plug-in. This tool has the following ID number: "urn:axmedis:ipmp:tool:id:0011"

To invoke this plugin, there are two parameters:

- Symmetric, will be `1` to symmetric encrypting, `0` to asymmetric encrypting
- Password, with the password to be used.

The constructor of the plugin will be called with a `0` or a `1` a a parameter, depending whether the required operation is encrypting or decrypting.

10.5.2.2 Implementation Details

As described above, `Alg` is an abstract class which is the base class of all Protection Tools. Therefore, all protection functionality has to be derived from the base class `Alg.` In the following example, the class `EncryptionDecryptionPlugin` integrates OpenSSL functionality to encrypt and decrypt streams of data. It is derived from `Alg` and the `EncDecSup` AXMEDIS Module

```
class EncryptionDecryptionPlugin : public Alg, public EncryptionDecryption
{
public:
        EncryptionDecryptionPlugin(int mode);
        ~EncryptionDecryptionPlugin();
        virtual unsigned long int BlockProcess(unsigned long int);
        virtual void init();
private:
        void Symmetric();
        void Asymmetric();
        int m_mode; // 0=encrypt, 1=decrypt
        class Algorithm *m_cipher;
        axeds::byteType *m_cKey;
        Data m_password;
        int m_passwordLength;
};
```

The `BlockProcess` function has to contain the encryption/decryption functionality. The encryption can be symmetric or asymmetric, and the corresponding private method will be called according to the parameter “symmetric”.

```
unsigned long int EncryptionDecryptionPlugin::BlockProcess(unsigned long int a)
{
    m_mode=a;
    AxParameterSTRING *symmetric=(AxParameterSTRING *) getParameter("symmetric");
    string ssymmetric=symmetric->getValue();
    if (ssymmetric=="true")
        Symmetric();
    else if (ssymmetric=="false")
        Asymmetric();
    state = DONE;
    return 512;
}
```

Parameters (such as the key) have to be set in the constructor. They will be read within the function “init”, as shown in the following. This should be highly instructive:

```
void EncryptionDecryptionPlugin::init()
{
    //1. WE RETRIEVE THE PASSWORD FROM THE PARAMETERS MAP (see parent class Alg)
    //what happens if there is no password?
    AxParameterSTRING *passwd=(AxParameterSTRING *)getParameter("passwd");
    std::string sPasswd=passwd->getValue();
    AxParameterSTRING *symmetric=(AxParameterSTRING *)getParameter("symmetric");
    std::string sSymmetric=symmetric->getValue();

    //2. WE INSTANTIATE THE ALGORITHM
    m_passwordLength=sPasswd.size();
    m_cipher = new Algorithm( axeds::AES_128_CBC_ALGORITHM);
    m_cKey = new axeds::byteType[m_cipher->getSizeKey()];

    //3. WE FILL OUR PASSWORD WITH WHAT WAS READ
    axeds::byteType *arr = new axeds::byteType[m_passwordLength];
    memcpy(arr,sPasswd.c_str(),m_passwordLength);
    m_password.setData(arr,m_passwordLength);

    //IN THIS POINT, THE ALGORITHM, IS GOING TO BE RUN VIA "BlockProcess"
};
```

As described in the previous section the following DLL-functions have to be implemented:

```
extern "C" Alg* createIPMPTool(const std::string& toolID, bool encoding)
extern "C" void releaseIPMPTool(Alg* tool)
```

The function `createIPMPTool` creates the required objects for encryption and decryption of the content. These objects are created in the constructor of the class `EncryptionDecryptionPlugin`.

```
extern "C" __declspec(dllexport) Alg* createIPMPTool(const std::string& toolID,bool encoding)
{
        if(toolID==std::string(ENCDECSUPID))
        {
                return new EncryptionDecryptionPlugin(encoding ? 0 : 1);
        }
        return NULL;
}
```

For completeness, the corresponding objects have to be freed. This is done by calling the function `releaseIPMPTool`.

```
extern "C" __declspec(dllexport) void releaseIPMPTool(Alg* tool)
{
    delete tool;
}
```

To finalize the plug-in development, an XML-description of the available functionality has to be described. This is shown in the following.

```
<?xml version="1.0" encoding="utf-8"?>
<pin:Plugin xmlns:pp="http://www.axmedis.org/ipmp-tool-schema"
xmlns:pin="http://www.axmedis.org/plugin-schema" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-
IPMPINFO-NS" xmlns:prm="http://www.axmedis.org/parameter"
xsi:schemaLocation="http://www.axmedis.org/ipmp-tool-schema ipmp-tool-schema.xsd
http://www.axmedis.org/plugin-schema plugin-schema.xsd urn:mpeg:mpeg21:2004:01-IPMPINFO-NS
ipmpinfo.xsd">
        <pin:GeneralDescriptor>
                <pin:Category>IPMPTool</pin:Category>
                <pin:Identifier>encdecsup-plugin</pin:Identifier>
                <pin:Library>protection processor plugin 11</pin:Library>
                <pin:Version>0.0.1</pin:Version>
                <pin:Vendor>Axmedis</pin:Vendor>
                <pin:MainLibrary>encdecsup-plugin.dll</pin:MainLibrary>
                <pin:Description>Provides some ipmp tools</pin:Description>
        </pin:GeneralDescriptor>
        <pin:ComponentsSignature>
                <dsig:SignedInfo>
                        <dsig:CanonicalizationMethod Algorithm=""/>
                        <dsig:SignatureMethod Algorithm=""/>
                        <dsig:Reference>
                                <dsig:DigestMethod Algorithm=""/>
                                <dsig:DigestValue>AA==</dsig:DigestValue>
                        </dsig:Reference>
                </dsig:SignedInfo>
                <dsig:SignatureValue
xmlns="http://www.w3.org/2000/09/xmldsig#">AA==</dsig:SignatureValue>
        </pin:ComponentsSignature>
        <pp:IPMPToolList>
                <pp:Tool pp:type="authoring">
                        <ipmpinfo:IPMPToolID>urn:axmedis:ipmp:tool:id:0001</ipmpinfo:IPMPToolID>
                        <pp:Name>encdecsup</pp:Name>
                        <pp:Description>Encrypts and decrypts blocks of bytes</pp:Description>
                </pp:Tool>
        </pp:IPMPToolList>
        <pin:Signature>
                <dsig:SignedInfo>
```

```
                        <dsig:CanonicalizationMethod Algorithm=""/>
                        <dsig:SignatureMethod Algorithm=""/>
                        <dsig:Reference>
                                <dsig:DigestMethod Algorithm=""/>
                                <dsig:DigestValue>AA==</dsig:DigestValue>
                        </dsig:Reference>
                </dsig:SignedInfo>
                <dsig:SignatureValue>AA==</dsig:SignatureValue>
        </pin:Signature>
</pin:Plugin>
```

10.5.2.3 Project Dependencies

The provided ZIP-file for the protection tool plug in contain all AXMEDIS-files for a successful standalone compilation of the plug in. Additionally, some extra libraries are necessary:

- OpenSSL (tested with version 0.9.8a)
    - http://www.openssl.org/
    - http://www.slproweb.com/products/Win32OpenSSL.html
- XERCES C++ Parser (tested with version 2.6.0)
    - http://xml.apache.org/xerces-c/

Furthermore, the plug in has dependencies to the following AXMEDIS components:

- axom
- common
- encdecsup
- keygen
- pluginmanager

10.6 AXMEDIS Plug in Content Processing and Protection Tools

This section provides a short overview of the existing functionality that is integrated via the plug-in interface. Technical details can be found in DE-3-1-2-3-7 Specification of AXMEDIS - External Processing Algorithms.

10.6.1 Adaptation Tools and Algorithms for Text

Document adaptation tools provide functions which can convert a text document file modifying its format.

10.6.2 Adaptation Tools and Algorithms for Video

The adaptation tools and algorithms for video provide the basic functions that are needed to convert video files. The module for video adaptation is based on the FFMPEG library (http://ffmpeg.sourceforge.net/index.php).

FFmpeg is “a set of free computer programs that can record, convert and stream digital audio and video. It includes libavcodec, a leading audio/video codec library. FFmpeg is developed under Linux, but it can compile under most operating systems, including Windows ... Notable is that most FFmpeg developers are part of either the MPlayer, xine or VideoLAN project as well.” (see http://en.wikipedia.org/wiki/FFmpeg)

10.6.3 Adaptation Tools and Algorithms for Images

This module gives the possibility to use algorithms and tools for adaptation of images. The main adaptation functions needed by the AXMEDIS Framework could be summarised in:

- Scaling

- Resolution improvement/reduction
- Colour to Greyscale conversions
- Format transcoding
- Composition with other images
- Adding widgets and graphic motifs
- Text drawing
- Image decomposition

These functions are implemented by defining specific algorithms or using graphic libraries. An example of library is given by the *ImageMagick Library*.

10.6.4 Adaptation Tools and Algorithms for Audio

The FFmpeg Audio Transcoding function can be used to convert an audio file into a different format and/or codec (please refer to section 4.4.10 for a complete description of formats and codecs supported in decoding and encoding). Apart from the bit rate reduction depending on the selected codec, one can further reduce the size of the resulting audio file by changing its sample rate and its number of audio channels. Moreover one can select only a specific portion of the input file to produce the resulting output file by specifying starting and ending points in the input file.

The libsndfile Audio Transcoding function can be used to convert an audio file into a different format and/or codec (please refer to section 4.4.10 for a complete description of formats and codecs supported in decoding and encoding). Apart from the bit rate reduction depending on the selected codec, one can further reduce the size of the resulting audio file by changing its sample rate and its number of audio channels. Moreover one can select only a specific portion of the input file to produce the resulting output file by specifying starting and ending points in the input file.

10.6.5 Adaptation Tools and Algorithms for Multimedia

Text and images have long been the main resources for Web content. Yet, as new formats emerge, rich multimedia presentations are making their entrance into the Web world and are being increasingly used for newscasts, educational material, entertainment etc. At the same time, the 3rd Generation Partnership Project and the 3rd Generation Partnership Project 2 seek to provide uniform delivery of rich multimedia over newly evolved, broadband mobile networks (3rd generation networks) to the latest enabled cell phones.

By rich media, we refer here to a broad range of digital media choreographing audio, video, text, graphics and synthetic animations in real time. Another key feature of such media is interactivity. Rich media may indeed respond directly to user interactions allowing for new ways of consuming media. For example, a pre-recorded webcast may be coupled with a synchronized slide show that allows user interaction with the mouse.

In the recent years, a large number of new multimedia formats have been created independently from each other including among others Quicktime, RealVideo, Advanced Streaming Format, Shockwave, SMIL and MPEG-4 BIFS. AXMEDIS Multimedia Adaptation tools focus on the last two since they are non-uproprietary and support a large variety of usage scenario.

To adapt multimedia files two goals are considered: direct transcoding of a multimedia file towards another multimedia file format and adaptation of the simpler media files embedded into the multimedia file. The later goal asks for means to extract simple media files from multimedia files and means to reintegrate adapted media files into the richer multimedia file. More specifically, the multimedia adaptation tool should provide:

- Transcoding functions: SMIL to MP4, MP4 to ISMA compliant MP4, MP4 to 3GP compliant MP4, SWF to MP4.
- Extraction functions: extraction of media files from richer media files (for example, extracting a video file from a multimedia file to apply specific video adaptation functions).

- Embedding functions: embedding of media files into richer media files (for example, adding an MP3 file into a MP4 file which already contains a video track and a subtitle track).

10.6.6 Adaptation Tools and Algorithms for Metadata

The MetaDataMapper library is used by the Metadata Mapper Editor and the metadata adaptation tools including the jsmetadatamapper libraries for scripting adaptation in the AXMEDIS Content Processing area (AXCP).The adaptation tools and algorithms for metadata creates mapping information as Extensible Stylesheet Language Transformation (XSLT) between source and target XML metadata documents and converts a source XML metadata document to a target metadata according to existing mapping information.

When used by the Metadata Mapper Editor, the MetadataMapper library builds an XSLT document from the mapping set in the GUI where the user connects XML elements from the source metadata to the destination metadata.

The Metadata Mapper Library uses the Apache Xerces-C parser to parse the input XML documents and build an XSLT document containing mapping information. Xalan (an XSL processor) is used to transform metadata documents according to an XSLT document created in the Metadata Mapper Editor.

The main adaptation functions produced by this module can be summarised as:

- Connecting XML elements to generate an XSLT
- Loading and Saving a generated XSLT
- Transforming XML metadata by applying the generated XSLT to adapt the XML elements

10.6.7 Adaptation Tools and Algorithms for DRM information

DRM adaptation involves the adaptation of the related licenses, as derived AXMEDIS objects or digital resources within the AXMEDIS objects can be seen as new creations with regard to original ones. Therefore, new licenses must be created during the adaptation process, always respecting the terms and conditions fixed in the original license or licenses for the adapted AXMEDIS objects or contents within these objects.

Inside AXMEDIS project, DRM information (mainly licenses and PARs) associated to AXMEDIS Objects is expressed in XML language by using MPEG-21 REL. Nevertheless, this is not the only rights expression language supported, as we are also able to create licenses in OMA DRM REL language v2.0. Licenses expressed in this language are particularly useful for mobile devices.

In order to adapt this information to different rights expression languages, also based in XML or to adapt a license to be more compact in order to use it into portable devices (for instance, mobile phones or PDAs), we make use of existing libraries for manipulating XML documents.

The main adaptation function produced by this module can be summarised in:

- Compacting licenses for their use in portable devices
- Translating licenses from one rights expression language to another
- Automatic generation of a license when an adaptation over the content it applies is done

For XML DRM rules transcoding, the *Xerces Libraries* can be used to parse a given piece of XML data.

It is currently available the translation between the two rights expression language above mentioned, MPEG-21 REL and OMA DRM REL v2.0 in both directions. At the present moment, not all the elements present in one license expressed in one of these languages can be translated to the other. Nevertheless, we are working in the addition of new profiles defined in MPEG-21 REL for directly supporting OMA DRM REL. The user requesting a translation will receive the translated license as well as the non-translatable elements between languages. Then, he will be able to decide if the translated license is acceptable or not.

10.6.8 Adaptation Tools and Algorithms for Ringtones

Ringtone Adaptation refers to the adaptation of ringtones of popular formats to enhance usability and manage the variable delivery to cater for heterogeneous client devices and user requirements on-demand. It uses external libraries like FFMPEG and LIBSNDFILE to convert the ring tones from one format to another and to resample it based on the client device. Following functionalities are provided by this plug-in:

- **Conversion of formats**
    a. Convert a ringtone from one format to another. The plug-in provides support for formats such as x-wav, x-aiff, x-ms-wma, basic, mpeg, mid.
    b. Convert to MP3 format: the input formats supported are x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)
    c. Convert to WAV format: the input formats supported are x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

- **Resampling**
    a. Resample a ringtone: change frequency, bitrate, sampling rate etc.

- **Information about an input ringtone**
    a. The plug-in can be used to get information such as mime type, number of channels, bit rate, sampling rate etc. about an input ringtone file.

- **Clip a ringtone for reduction to a specified duration**
    a. The plug-in can be used to clip an input ringtone file to a desired duration e.g. 30 seconds.

10.6.9 Descriptor extractor as Fingerprint for Text files

Automatic keyword extraction aims to provide the user with a set of descriptors which represent the contents of the document, and which are further exploited to perform advanced searches within the textual repository. Extracted keywords can also be used to identify higher levels of descriptors, such as the document domain. The procedure is divided into three main steps:

1) **Comparative frequency analysis: mono-term keywords extraction:** This process is performed through an integration of resources and standard algorithms, by means of comparison between the document and the referring universe, represented by a general corpus. The output is a list of nouns in the document, ordered by TF.IDF scores

2) **Semantic analysis: mono-term keywords semantic disambiguation and domain detection:** Since nouns are potentially ambiguous with respect with their semantics, a word sense disambiguation (WSD) procedure is run over the output of the previous step. This would allow, in principle, the translatability of keywords in different languages. WordNet Domains database is also exploited to determine the “area of discussion” to which each keyword belongs, so providing other keys for content identification.

3) **Internal analysis of lexical associations: multi-term keywords detection:** The extracted keywords are further refined from the point of view of the accuracy, of the content identification, and of the value of the descriptors, referring to language properties of word association: high frequency collocations within the text are considered more definite and highly representative of its content.

10.6.10 Descriptor extractor as Fingerprint for Audio files

With electronic music distribution (EMD), music catalogues have become huge. The biggest online services now propose around 2 million tracks while personal users have the possibility to carry thousand of songs on their portable music player. In fact, the amount of digital music is now urging for reliable and fast tools for content analysis and description, to be used for searches, content queries and interactive access. In the context of the AXMEDIS project, a number of algorithms for audio content analysis and description have been developed and implemented to ease audio content retrieval and browsing into collections.

10.6.11 Descriptor extractor as Fingerprint for Video Files

Within the AXMEDIS project no research on the content descriptors for video is performed. Instead, the extensibility of the AXMEDIS plug-in interface is exploited and validated by integrating existing state-of-the-art algorithms.

The MPEG-7 eXperimental Model (XM) is the MPEG-7 Reference Software and is available for downloading at http://www.lis.e-technik.tu-muenchen.de/research/bv/topics/mmdb/mpeg7.html and via CVS.[1] The MPEG-7 XM Reference software includes state-of-the-art content description algorithms for audio-visual content.

Some of the available functionality was selected based on the identified user requirements and integrated into the AXMEDIS framework as a plug-in.

10.6.12 Fingerprint Estimation for Text files

The text fingerprint plug-in output will be a string in which the value is stored. Moreover the document comparison functionality also implemented in the plug-in will give as a result a normalized floating point value representing the degree of similarity between two given input documents.

The text fingerprints plug-in aim is twofold, it provides a way of calculating a fingerprint value of the documents provided as input, moreover it provides functionalities for similarity estimation between two documents without making prior assumptions on the language. The fingerprint algorithm hashes the ASCII representation of the input file and gives as result a string in which the fingerprint value is stored (full or selective hash values based on the analyzed document structure).

For similarity comparison a plug-in specific function is provided. This functionality could be exploited by several use cases including: identification of content from a sub part or when the different formats comparison is not straightforward, plagiarism detection and so on. The algorithm allows for robust multilevel comparison of documents taking into account document structure and leveraging the plagiarist behaviour, which is modeled as a combination of 3 basic actions: insertion, deletion, substitution. We recognize that this behavior may occur at various level of the document structure: the plagiarist may insert, delete or substitute a word, period or a paragraph. The procedure consists in two main steps: document structure extraction and plagiarism function calculation. We propose a recursive plagiarism evaluation function to be evaluated at each level of the document structure which is based on the Levenshtein edit distance.

For what concerns fingerprint it has to be said that in MPEG-7 this kind of meta-information (for verification of the text documents) is not consider so far. Within AXMEDIS it has to be identified, how cryptographic hash values can be integrated best into MPEG-7 and the relationship with MPEG-21.

10.6.13 Fingerprint Estimation for Audio files

The Audio fingerprint plug-in is a tool that extracts an audio fingerprint of a given audio stream within a multimedia file. The audio stream can be embedded either in a normal audio file (mpg, wav, wma, etc…) or within a video file (mpeg, wmv, avi, etc…).

The Fingerprint extractors for audio files automatically calculate a digest describing its main characteristics in way suitable for automatic verification of AXMEDIS objects. Thus the descriptor is a low level description according to the previous definition.

[1] Access to the CVS is only possible through registration at the national standardization body.

Audio files are streams. This input stream is segmented. That means that for each segment a "sub-fingerprint" is calculated. Depending on the length of the input sequence, typically a request to the database does not result in a single fingerprint but in an array of fingerprints.

10.6.14 Fingerprint Estimation for Video files

Video files are related to audio as video also is time dependent: They are streams. The input stream is segmented. For each segment a "sub-fingerprint" is calculated. Depending on the length of the input sequence, typically a request to the database does not result in a single fingerprint but in an array of fingerprints.

10.6.15 Fingerprint Estimation for Metadata

For the verification of the objects metadata only cryptographic hash functions are feasible. This hash function is applied to all kinds of meta-data of an AXMEDIS object.

10.6.16 Fingerprint Estimation for Generic Files

For the verification of the objects metadata only cryptographic hash functions are feasible. This hash function is applied to all kinds of meta-data of an AXMEDIS object.

10.6.17 External Protection Libraries

The information age has seen the development of electronic pathways that carry vast amounts of valuable commercial content between individuals and companies. Unfortunately the unprecedented levels of access provided by systems like the Internet also expose this data to breaches of confidentiality, disruption of service, and copyrights infringements. For this reason, in the content distribution field, many applications use more and more DRM (Digital rights Managements) solutions.

This means that content use/manage applications need security module implemented within. Unfortunately the security systems required to protect data are generally extremely difficult to design and implement, and even when available tend to require considerable understanding of the underlying principles in order to be used. This has lead to a proliferation of "snake oil" products that offer only illusionary security, or to organizations holding back from deploying online information systems because the means to secure them are not readily available, or because they employed weak, easily broken security that was unacceptable to users.

The cryptlib security library provides a complete set of cryptographic algorithms that fit the Axmedis needs.

11 References and links

11.1 AXMEDIS Tutorials

- General Tutorial and Overview (December 2006, Leeds, UK)
  - PPT: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2659
  - Video on part 1 - torrent
  - Video on part 2 - torrent
  - Video on part 3 - torrent
  - Video on part 4 - torrent
  - Video on part 5 - torrent
  - Video on part 6 - torrent
- Content Production Tutorial http://www.axmedis.org/documenti/view_documenti.php?doc_id=2667
- Content Distribution Tutorial http://www.axmedis.org/documenti/view_documenti.php?doc_id=2666
- Content Processing Tutorial http://www.axmedis.org/documenti/view_documenti.php?doc_id=2655
- Workflow Tutorial http://www.axmedis.org/documenti/view_documenti.php?doc_id=2652

11.2 AXMEDIS tools for free download

- General download page: http://www.axmedis.org/documenti/documenti.php?area_id=1
- AXMEDIS tools (AXTOOLS) which include http://www.axmedis.org/documenti/view_documenti.php?doc_id=2885
  - AXMEDIS players described in the next point;
  - AXMEDIS Editor;
  - AXMEDIS Content Processing GRID tools: AXCP Scheduler, AXCP Rule Editor, AXCP GRID node, etc.;
  - DRM editor: License Editor, examples of licenses;
  - AXMEDIS plug ins for video, audio, document, images, processing, etc.
  - Examples of AXMEDIS objects in several type; http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Cross+Media+Content%3A+Examples
  - Examples of AXCP scripts for content processing of several kind; http://www.axmedis.org/tiki/tiki-index.php?page=AXMEDIS+Content+Processing+Scripts
  - etc.
- AXMEDIS players: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2884 including (all these players are included into the above mentioned package):
  - PC players,
  - PC player with Skin,
  - Bordas and Nathan Player for PC,
  - AXMEDIS Active X player for PC
  - Content Examples etc.
- AXEPTools: P2P client tool for establishing connection with the AXMEDIS P2P B2B network as Business User: http://www.axmedis.org/documenti/view_documenti.php?doc_id=3102
- AXMEDIA: P2P client tool for establishing connection with the AXMEDIS P2P B2B network as final users: http://www.axmedis.org/documenti/view_documenti.php?doc_id=3103
- AXMEDIS PDA player for Windows Mobiles 5: It is capable to play AXMEDIS objects based on SMIL, HTML, video, audio, MPEG-4 files, etc. AXMEDIS PDA player for AXMEDIS MPEG-21 content including resources with presentations layer based on MPEG-4, HTML and SMIL Unzip the file, copy the CAB file and execute it on the PDA
  - http://www.axmedis.org/documenti/view_documenti.php?doc_id=3107

11.3 AXMEDIS Framework Specification

- AXMEDIS Framework General aspects, Editor and Model http://www.axmedis.org/documenti/view_documenti.php?doc_id=1891
- AXMEDIS Command Manager http://www.axmedis.org/documenti/view_documenti.php?doc_id=2686
- AXMEDIS Object Manager and Protection Processor: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1972
- AXMEDIS Editor and Viewers: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2213
- AXMEDIS External Editors, Viewers and Players: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2211
- AXMEDIS Content Processing Area: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1958
- AXMEDIS External Processing Algorithms: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2017
- AXMEDIS CMS Crawling capabilities: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1907
- AXMEDIS Database and query support: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1932
- AXMEDIS AXEPTool and AXMedia Tools: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2718
- AXMEDIS Programme and Publication Tools: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1885

- AXMEDIS Workflow Tools: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1883
- AXMEDIS Certifier and Supervisor and networks of AXCS http://www.axmedis.org/documenti/view_documenti.php?doc_id=1952
- AXMEDIS Protection Support http://www.axmedis.org/documenti/view_documenti.php?doc_id=1882
- AXMEDIS Accounting and Reporting : http://www.axmedis.org/documenti/view_documenti.php?doc_id=1887
- Definitions Terms tables links http://www.axmedis.org/documenti/view_documenti.php?doc_id=1388

AXMEDIS reports on basic enabling technologies

- Content Model and Managing, MPEG-21, authoring, etc. http://www.axmedis.org/documenti/view_documenti.php?doc_id=2324
- Content indexing and querying: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2436
- Content processing, Composition and Formatting http://www.axmedis.org/documenti/view_documenti.php?doc_id=2374
- Content sharing and Production on P2P: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2454
- Content Protection and Supervision http://www.axmedis.org/documenti/view_documenti.php?doc_id=2371
- Content Distribution via Internet http://www.axmedis.org/documenti/view_documenti.php?doc_id=2451
- Content Distribution via Mobile http://www.axmedis.org/documenti/view_documenti.php?doc_id=2442
- Content Distribution via Satellite data broadcast http://www.axmedis.org/documenti/view_documenti.php?doc_id=2313
- Usability issues http://www.axmedis.org/documenti/view_documenti.php?doc_id=2339
- AXMEDIS vs DMP MPEG21 Analysis http://www.axmedis.org/documenti/view_documenti.php?doc_id=1063
- AXMEDIS Framework Infrastructure, guidelines and some tools http://www.axmedis.org/documenti/view_documenti.php?doc_id=1391
- AXMEDIS Framework Validation and integration http://www.axmedis.org/documenti/view_documenti.php?doc_id=2445

Basic knowledge reports

- User requirements http://www.axmedis.org/documenti/view_documenti.php?doc_id=1712
- Use Cases http://www.axmedis.org/documenti/view_documenti.php?doc_id=1824
- Test Case http://www.axmedis.org/documenti/view_documenti.php?doc_id=2023

Content Modeling and Test Cases

- Content Aspect Specification http://www.axmedis.org/documenti/view_documenti.php?doc_id=1389
- Content Aspect Specification Appendix http://www.axmedis.org/documenti/view_documenti.php?doc_id=1670
- Content for Test Cases and Validation http://www.axmedis.org/documenti/view_documenti.php?doc_id=1393
- Content Selection Guidelines http://www.axmedis.org/documenti/view_documenti.php?doc_id=1390
- Multilingual Guidelines and Technical Solutions http://www.axmedis.org/documenti/view_documenti.php?doc_id=1427
- AXMEDIS Editorial Format Guidelines and basic examples http://www.axmedis.org/documenti/view_documenti.php?doc_id=1394

11.4 AXMEDIS Framework Demonstrators, Cases, Trials, for distribution etc.

- requirements and use cases of AXMEDIS ELTEO of the content distribution for DVB-T to STB of Telecom Lithuania, and content distribution of Telecom Estonia http://www.axmedis.org/documenti/view_documenti.php?doc_id=2978
- requirements and use cases of the 4HOME take up, demonstrators of BBC, TI, SDAE, including domains, AXMEDIS for broadcasting, and OMA integration and distribution http://www.axmedis.org/documenti/view_documenti.php?doc_id=2976
- Specification final version of Take up AXMEDIS ELTEO for Video on demand, STB, IPTV solutions based on AXMEDIS technology: http://www.axmedis.org/documenti/view_documenti.php?doc_id=3096
- Integrated CMS integration aspects: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2853
- Integrated prototype: automated content production and formatting: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2939
- Integrated Distribution on demand via Internet http://www.axmedis.org/documenti/view_documenti.php?doc_id=2951
- Integrated distribution via satellite data broadcast: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2954
- Integrated distribution towards mobiles: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2945
- Integrated Distribution towards PDA via Kiosks: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2944
- Content Posting Portal, Content Posting for Final User publication, SIAE Trial presentation:
  - http://www.axmedis.org/documenti/view_documenti.php?doc_id=2922
  - http://www.axmedis.org/documenti/view_documenti.php?doc_id=2923
- VARIAZIONI project portal: http://www.variazioniproject.org/

11.5 Brochures and press cutting (a part)

- AXMEDIS Project Brochure http://www.axmedis.org/documenti/view_documenti.php?doc_id=2712
- Annual Public Report (2006) http://www.axmedis.org/documenti/view_documenti.php?doc_id=2471
- Annual Public Report (2005) http://www.axmedis.org/documenti/view_documenti.php?doc_id=1439
- AXMEDIS Project Synopsis http://www.axmedis.org/documenti/view_documenti.php?doc_id=1668
- Digital Media in Italy presentation http://www.axmedis.org/documenti/view_documenti.php?doc_id=1669

11.6 Other references

- ISO/IEC, ISO/IEC FDIS 21000-5 - Rights Expression Language. ISO/IEC JTC1/SC 29/WG 11/N5839. July 2003.
- ISO/IEC, ISO/IEC FDIS 21000-6 - Rights Data Dictionary. ISO/IEC JTC 1/SC 29/WG 11/N5842. July 2003.
- Iannella, R.: Open Digital Right Language (ODRL) Version 1.1. http://odrl.net/1.1/ODRL-11.pdf . August 2002.
- Open Mobile Alliance (OMA), http://www.openmobilealliance.com/
- OMA DRM Rights Expression Language version 2 (OMA DRM REL v.2), http://www.openmobilealliance.com/
- ISO/IEC, Study of ISO/IEC FCD 21000-4 IPMP Components. ISO/IEC JTC 1/SC 29/WG 11/N7426. July 2005.

Further Information

Please contact the AXMEDIS coordinator (Paolo Nesi: Nesi@dsi.unifi.it) if you are interested in having more information to create your own AXMEDIS compliant plug-ins for your algorithms.