Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE4.6.1.3
Content Distribution via Internet, updated version DE4.6.1.2

Version: 1.8
Date: 02/10/2007
Responsible: TISCALI (revised and closed 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: DE4.6.1.3
Contractual Date of Delivery: Month 36 (End of August 2007)
Actual Date of Delivery: 3/10/2007
Work-Package contributing to the Deliverable: WP4.6
Task contributing to the Deliverable: WP4.6
Nature of the Deliverable: Report and prototype
Author(s): TISCALI, DSI, FUPF/UPC, CRS4, EPFL

Abstract:
AXMEDIS will pursue an integrated solution to content distribution, providing flexibility and scalability to support any kind of content over any kind of network and configurable to support terrestrial, satellite and mobile transport protocols

Keyword List:
CMS, DRM integration, B2C distribution, AXMEDIS and Windows Media, AXMEDIS players, AXMEDIS PDA player, internet distribution, P2P and DRM.

AXMEDIS Copyright Notice

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

Market and end-users are pressing content industry to reduce prices. This is presently the only solution to setup viable and sustainable business activities with e-content. Production costs have to be drastically reduced while maintaining product quality. Content providers, aggregators and distributors need innovative instruments to increase efficiency. A solution is automating, accelerating and restructuring the production process to make it faster and cheaper. The goals will be reached by: (i) accelerating and reducing costs for content production with artificial intelligence algorithms for content composition, formatting and workflow, (ii) reducing distribution and aggregation costs, increasing accessibility, with a P2P platform at B2B level integrating content management systems and workflows, (iii) providing algorithms and tools for innovative and flexible Digital Rights Management, exploiting MPEG-21 and overcoming its limits, supporting several business and transactions models. AXMEDIS consortium (producers, aggregators, distributors and researcher) will create the AXMEDIS framework with innovative methods and tools to speed up and optimize content production and distribution, for *production-on-demand*. The content model and manipulation will exploit and expand MPEG-4, MPEG-7 and MPEG-21 and others real and de-facto standards. AXMEDIS will realize demonstrators, validated by means of real activities with end-user by leading distributor partners: (i) tools for content production and B2B distribution; (ii) content production and distribution for i-TV-PC, PC, kiosks, mobiles, PDAs. The most relevant result will be to transform the demonstrators into sustainable business models for products and services during the last project year. Additional demonstrators will be 2-3 associated projects launched as take up actions. The project will be supported by activities of training, management, assessment and evaluation, dissemination and demonstration at conference and fairs.

This deliverable is devoted to the description, analysis and report on issues concerning the content distribution via Internet done inside WP4.6.

This activity is by no means finished with the completion of this deliverable, but it has to be revised during the development of the project.

Main deliverables in WP4 are:

- DE4.1.1.1 – Content Modelling and managing (M13), report and prototype;
- DE4.1.1.2 – Content Modelling and managing, first update (M24);
- DE4.1.1.3 – Content Modelling and managing, final version, (M36), report and prototype;
- DE4.2.1.1 – Content indexing, monitoring and querying (M13), report and prototype;
- DE4.2.1.2 – Content indexing, monitoring and querying, first update (M24);
- DE4.2.1.3 – Content indexing, monitoring and querying, final version, (M36), report and prototype;
- DE4.3.1.1 – Content Composition and formatting (M13), report and prototype. It also includes the details about the integration of AXMEDIS with workflow management tools;
- DE4.3.1.2 – Content Composition and formatting, first update (M24);
- DE4.3.1.3 – Content Composition and formatting, final version, (M36), report and prototype.
- DE4.4.1.1 – Content sharing and production on P2P (M13), report and prototype;
- DE4.4.1.2 – Content sharing and production on P2P, first update (M24);
- DE4.4.1.3 – Content sharing and production on P2P, final version, (M36), report and prototype;
- DE4.5.1.1 – Content Protection and Supervision (M13), report and prototype;
- DE4.5.1.2 – Content Protection and Supervision, fist update (M24);
- DE4.5.1.3 – Content Protection and Supervision, final version, (M36), report and prototype;
- DE4.6.1.1 – Content Distribution via Internet (M13), report and prototype;
- DE4.6.1.2 – Content Distribution via Internet, first update (M24);
- DE4.6.1.3 – Content Distribution via Internet, final version, (M36), report and prototype;
- DE4.7.1.1 – Content Distribution toward mobiles (M13), report and prototype;
- DE4.7.1.2 – Content Distribution toward mobiles, first update (M24);

- DE4.7.1.3 – Content Distribution toward mobiles, final version, (M36), report and prototype;
- DE4.8.1.1 – Content Distribution via satellite data broadcast, the push optimisation and the on demand problem (M13), report and prototype;
- DE4.8.1.2 – Content Distribution via satellite data broadcast, the push optimisation and the on demand problem, first update (M24);
- DE4.8.1.3 – Content Distribution via satellite data broadcast (for both PC and STBs), final version, (M36), report and prototype;
- DE4.9.1.1 – The Usability issues for the AXMEDIS production tools (M13), report;
- DE4.9.1.2 – The Usability issues for the AXMEDIS production tools, first update (M24);
- DE4.9.1.3 – The Usability issues for the AXMEDIS production tools, final version, (M36), report.

The main activities that have supported the production of this deliverable are related to:

WP4.6 – Content Distribution via Internet - This deliverable is related to the analysis of distributors and their Content Management modules in order to integrate advanced and business sensible solutions such as application and content staging, reporting, usage tracking, personalization, scheduling, workflow management, advertising etc. The aim is supporting all main standard and commercial content formats, repositories and applications that are handled by the AXMEDIS Project or by its partners systems. Extend software components to natively support all components of the AXMEDIS project, in particular the P2P services. Extend distribution features to provide the appropriate framework for the fulfillment of goals set in the development of the fully integrated distribution tool with AXMEDIS solutions and tools. Solutions for the production and delivery of on demand content on PDA and PC.

Details: analyze distributors and their Content Management modules required to integrate advanced and business sensible applications such as application and content staging, reporting, usage tracking, personalization, scheduling, workflow management, advertising etc.; identification of critical issues and of strategies for addressing these issues; Check aspects of content distribution via Internet from different points of view: DRM (problems, solutions, control), protocols (security, quality of service), access devices, interoperability with other systems, etc.; identification of the first versions of the java classes and application of them to the test cases provided to assess their functionalities as signature and integrity verification code; Design how to support all main standard and commercial content formats, repositories and applications that are supported by the AXMEDIS Project or by its partners systems; Evaluation of the use of standards in this distribution channel: MPEG, others; Extend software components to natively support all components of the AXMEDIS project, in particular the P2P services for clients, according to a B2C model.

1.1 Modified sections

This section summarizes main changes of this deliverable respect to the previous DE4.6.1.2 “Content distribution via Internet, first update).

In section 4, the state of the art of DRM integration and interoperability has been enriched adding further references to other DRM systems (sections: 4.1.10, 4.1.11, 4.1.12, 4.1.13, 4.1.14). Information regarding users and tools registration, certification, verification and object usage has been updated (section 4.2.3.3 and 4.2.3.6). Other updated subsections are those related to the realized tasks (4.3) and the integration of OMA tools (4.5.2).

In section 6.3, the subsection 6.3.3 has been added regarding the usability and performance issues of the AXMEDIA Tool.

In section 7.5 (and following subsections) the description of the AXMEDIS PDA player has been added, including information about the technology, the user interface and the issues related to the integration of the player into the AXMEDIS framework.

2 Introduction

AXMEDIS will pursue an integrated solution to content distribution, providing flexibility and scalability to support any kind of content over any kind of network, and configurable to support terrestrial, satellite and mobile transport protocols. The work has to lead at the solution to the specific problems of the distributors of cross media content via Internet. This can be traditional of broadband form movies, documents, multimedia contents, music, coursewares, etc.

The activities that have supported the production of this deliverable are related to the analysis of the relation between CMS and specific problems of the organizations devoted to the Internet distribution of multimedia contents.

The main goal of this analysis is to identify a set of features suitable for solving problems associated with cross-media assets management and for simplifying their integration with all the components of the AXMEDIS project. Distributors devoted to handling and delivering multimedia premium content can use this solution as a starting point and move toward a complete and easy to use environment for the publication and up-selling of content over Internet.

2.1 Specification of WP4.6 Content Distribution via Internet (TISCALI)

The main goal is to analyze the architecture of distributors devoted to handling multimedia premium content and to establish stable, documented and fully featured solutions open to integration with all the components of the AXMEDIS project, so to enable a complete and easy to use environment for the publication and up-selling of content over the Internet. This WP will address the specific problems of the distributors of cross media content via Internet. Content may include movies, documentaries, multimedia content, music, course wares, etc.

A specific effort will be made on the distribution of feature movies that will constitute one of the major drivers for the development of the premium content market in the next years.

2.1.1 T4.6.1: Analysis of the CMSs related to the Internet Distribution (TISCALI) – completed–

2.1.2 T4.6.2: DRM integration and interoperability (UPC)

Based on the work done in the previous period, integration of DRM models for Internet distribution will be studied and implemented when possible. Tools and prototypes coming from WP4.5, like licensing models and PMS will be integrated here. It will be also studied how an effective integration can be done between AXMEDIS tools and existing commercial tools and solutions, especially Windows Media. A possible approach is to provide a mixed solution, based on both AXMEDIS and Windows Media and based from AXMEDIS and OMA. This work will be performed in collaboration with DSI that has already analysed the Windows Media and OMA solution for eventual integration with AXMEDIS authoring tools and players and by UPC-FUPF that already analysed the relationships from MPEG-21 REL and ODRL OMA. Other possibilities will be also studied, as new commercial products and initiatives may appear in the period. Different approaches can be taken to perform the integration of AXMEDIS with current DRM systems and tools. In this task, these approaches will be evaluated and the best solution will be proposed, presenting a prototype for performing the integration.

- M30: Study of possible integration with commercial tools. Usage of OMA tools of Siemens in AXCP (DSI). Consideration of OpenDRM tools.
- M36: Prototype of integration with commercial tools, considering the production of Windows Media licenses. This task will depend on the results of the integration study.

- M36: Support to the integration with the TISCALI distribution platform.

2.1.3 T4.6.3: P2P support for B2C application and distribution (DSI)

Study and development of an AXMEDIS P2P tool called AXMEDIA peer for the B2C distribution of AXMEDIS content to be used in the TISCALI demonstrator of WP9. Basically, actions required involve a customisation of the AXEPTool P2P support for B2B in the B2C distribution environment in order to make it simpler to work on B2C side. AXMEDIA has to be capable of accepting both AXMEDIS objects or any files: not only AXMEDIS objects can be "visible" and queried but files of any type or content (this will be customisable). AXEPTool and AXMEDIA can share the same peers so that to improve the performance of the P2P network. In any case, the BitTorrent trakers and the Query Support for P2P can be tuned to keep these worlds joined or separated. AXMEDIA tool is addressed to consumers, mainly not technical users and has to provide evidence of: AXMEDIS files that are protected and certified by providers: the GUI must specially mark protected files thus also certified files; monitor network traffic of AXMEDIS object and other files, providing a sort of real-time monitor for downloads and network traffic in order to give to the consumer the status of all running operations. A GUI should be simple and intuitive avoiding for its usage manual of training, simple and direct activation of the dowanloda from query and activation of the AXMEDIS player on the selected downloaded objects. Query interface has to be simpler: from the consumer point of view, searching for AXMEDIS objects or any file in the P2P network it is translated into making queries. A study of possible GUIs implementation will be done giving relevance to differences between the tool for B2B and to B2C sides and facing all the requirements for a consumer side point of view. First performance aspects will be faced studying different approaches in order to design the network topology, to discover query problems and GUIs usability. Identify solutions and methods for multi-device content delivery supporting access and queries from both PDA and PC. Check aspects of content distribution via Internet from many points of view: DRM (problems, solutions, control), protocols (security, quality of service), access devices, interoperability with other systems, integration with existing tools (for instance Windows Media), interoperability with DMP, etc.;

- M30: identification of the changes to be applied to AXEPTool in order to build AXMEDIA tools. Definition of the user interface of the AXMEDIA tool; early integration of the solution with TISCALI distribution chain;
- M36: almost final version of the AXMEDIA tools. Testing of the AXMEDIA tools in the real environment, distribution of the tool from the portal as free download, early performance analysis in using them in the same network with AXEPTools peers or in separate networks (defined in terms of P2P query portal and Tracker server); improved integration of the solution with TISCALI distribution chain;
- M36: Final version of the AXMEDIA tool for B2C distribution via P2P fully integrated with the TISCALI distribution model and chain.

2.1.4 T4.6.4: AXMEDIS clients for distribution on PC, PDA, Tablet PC (TISCALI)

This task is focused on the exploitation of the AXOM model and tools for the production of **AXMEDIS Players** for PC, tablet PC and PDAs. In this task the improvements of the players realized as AXCTIVEX for IE or as plug in for Mozilla will be performed, their robustness against attacks will be tested and validated exploiting the new version of the AXOM and the above mentioned new protection capabilities. The research will be focussed to assess the scalability of the AXMEDIS model and security of those players, their integration with PMS, AXCS, certification, registration, verification, validation; and the same for the added players, plug-in for processing, etc.

Integration and testing of the OSMO MPEG-4 player and of the Ambulant SMIL player to work with AXOM on both PC and PDA platforms, considering event reporting and streaming form AXOM into the SMIL and MPEG-4 player for rendering. Integration of the above tools with other integrated tools in AXMEDIS: document viewer, HTML/XML viewer, etc.

- M30: first release of the AXMEDIS player for PDA, including integration of OSMO and SMIL players, including direct access from those player to the AXOM resources, usage of the protection processor, streaming of resources from AXOM to those players;
- M36: improved release of the AXMEDIS player for PDA, including integration of OSMO and SMIL players, testing, performance analysis, integration with other player for resources of AXMEDIS. This version will provide evidence of feasibility of what can be ported on more powerful mobile devices, and an early performance analysis;
- M36: final version of the AXMEDIS player for PDA. This version will be more stable and all the necessary interfaces with external components will be implemented. A complete performance analysis will be produced.

3 Content distribution via Internet

3.1 Analysis of the CMSs related to the Internet Distribution (TISCALI)

The focus is on CMSs and on their main suitable features for distributors of cross media contents, identifying critical issues and defining strategies for addressing these issues.

A review of the main commercial and open source CMS solutions on the market will provide a state-of-the-art of these technologies and will highlight which features and systems make easier the integration with the AXMEDIS project and tools.

3.1.1 Content Management System

A Content Management System covers the complete life cycle of the contents, from providing tools to create them, through to editing, publishing, reviewing, searching and finally to archiving, always accordingly to a set of consistent predefined rules.
Within the CMSs world, there are several different types of products, and different styles of each. Elements of various CMS type and styles can even be mixed and matched. Not that long ago, the only choice was a commercial solution from a large vendor. Several commercial CMS applications are available at different conditions today, and an even greater number of free open source solutions are available.
There are hundreds of content management systems, all having different capabilities and strengths and every organization choose to adopt one instead of another depending on own unique set of specific requirements.

Not all CMSs are built to the purpose of presenting content for public consumption. For example the most part of the Document Management Systems may be considered particular CMS where the focus is just on electronics documents (such as article, presentation, paper, graph, proposals, projects, contracts, legal papers, etc.) and on their organization to efficiently create, capture, store, retrieve, classify, secure.
On the other hand for the organizations that found their business creating, aggregating, distributing the contents, the critical mission is to present, show and distribute them to the consumers. Solutions for these issues usually start from Web Content Management Systems, where the emphasis is on the management of Web contents (such as documents, reports, fact sheets, sound, video clips, etc.) trough Web pages, sites, portals. For bigger and more structured organizations manage all aspects of content publication processes, including Web, print and any alternative outputs is usually a requirement: for these specific reality Enterprise CMSs emphasize completeness over a broad scope.

Particular areas exist in the content distribution market where Digital Rights are a priority as happens to the music and video distributors. For properly managing information regarding intellectual property rights for the contents, ad-hoc DRM system solutions exist in the CMSs world. Other particular sectors where specific requirements must be satisfied, are those where the core business is the management of the so-called “digital assets" (images, video, audio, and other binary, non-textual content). Organizations that fall in this specific area are, for example, photo agencies and graphic design firms. All these systems are often complementary and new generation CMSs tend to integrate this kind of features in a more complete environment for organizations devoted to digital assets management and delivery.

This chapter will focus on Web content management, since that's what Internet content distribution organizations need. Content in the context of the Internet distribution and of a Web site includes digital items like documents (including reports, fact sheets, etc.), images, sound and video clips. All of them are basically stored within a database strictly integrated with the CMS and a Web CMS is a sophisticated tool that can be used to make the whole process of Web publishing much easier and more advantageous. Most content internet providers and distributors have their specific content management systems, larger companies have multiple systems in place as well.

Web pages and links can be dynamically generated from information held in the database so the design and presentation of content on a Web page is separated from the content creation process and this independence offers benefits for both content maker and Web developers.

With a CMS, Web developers can concentrate on the site design issues as structure, navigation, look-and-feel, etc. without having to worry about creating and maintaining content. The whole site can remain consistent since information is drawn from the content database and plugged into templates created by the Web developer for different types of Web page. The quality of being able to use templates and reusable design elements has a decisive influence making certain of coherent and uniform design through all parts of a Web site. Different templates can be designed for different areas of a site if needed; for example the layout of an on-demand cross media content list page may need to be different from the layout of a fact sheet or a news page. A CMS can also allow global changes to a site to be made much more easily if needed. The design of the template can be changed rather than having to edit each page individually. Templates can be designed to meet any necessary technical and accessibility standards.
The CMS implementations for Internet contents providers and distributors usually are focused on documents, Web contents, XML data and digital assets. The inclusion of content is controlled by the use of special tags in Web pages. Web CMS usually provides a scripting language that allows Web sites to dynamically receive, process, and display information. The most spread and common supported scripting languages are PHP, Perl, Python, but also Java especially for more complex operations. By making use of technologies such as XML (extensible Mark up Language) a CMS can also allow content that is stored in the database to be repurposed for different devices such as mobile phones, handheld computers, kiosks and Web TV as well as PCs.

On the other hand content writers focus and stay concentrate on writing content without having to worry about design issues. Content can easily be added to the database and displayed on a Web page by typing or pasting text into fields in a form or drag and drop it within the CMS administration system. Information about the content (metadata) such as title, description, keywords, author, publish date, review date, DRM data, specific media data as image size, audio/video length, format, bitrate, etc. can also be added. This facilitate searching for content that needs updating when it goes out of date and needs to be amended.

A CMS can also be used to help manage workflow and maintain quality control by allowing an overall administrator of the system to set roles and permissions for authors, editors and publishers. For example, several authors may be able to upload new material to the CMS at any time. The editor can be alerted when new content is added to the system. The new material can then be reviewed, approved, and finally published to the Web site by an authorized person. A CMS can therefore allow people without technical Web authoring skills to easily update and maintain a Web site.

Not mentioned yet other important features to always consider for a Web CMS include the integration capability with existing systems (for example think about the needs to be able to use an already existent suitable database instead of that provided by the CMS), the ability to run on existing equipments, the faculty to be comply with the current standards for Web technologies and to guarantee best compatibility across different platforms and browsers, the possibility to have different user access rights to check if users have proper authorizations to create, modify, publish, etc. the site pages, the facility to hold, format and edit text and encapsulate digital content such as images, audio and video to a Web page in an easy manner, the capability to have many simultaneous users (with features such as record locking to prevent clashing changes).

Each type of content management system provides a different set of business benefits. Most relevant business benefits (from the wide range of them) that can be obtained by using CMSs can be summarized as such the ones in the following list:

- streamlined creating/authoring process
- reduced turnaround efforts for new contents and changes
- facilitated re-use of content
- greater consistency

- increased flexibility
- support for decentralized authoring and content creation
- centralized workflow, approval processes and rules
- increased security
- reduced duplication of information
- greater capacity for growth
- reduced maintenance costs

Adopting a suitable CMS is clearly critical to the success of every Internet content provider and distributor: content is the object of the business and have to be managed as best as possible.

3.1.2 Distributors: management of media contents

In the course of the progressive development and diffusion of the multimedia contents it becomes more and more important to administrate information such as pictures, diagrams, audios and videos which may occur in the most different formats. This is a very critical issue especially for internet distributor companies that need a complete multimedia asset management to make the digital information available for optimal utilization and for delivering process over different channels.
The term "digital asset" includes all digital information (file, collections of file and metadata describing them) that create a value (i.e. asset) when accessed by a browser, a service or a player. Examples of assets include images, graphics, text, sound bites, video clips and so on. Assets can also be course materials or television programs.

The modern publishing industry knows that content doesn't follow a single and predefined path ending with the printing process: content can now be sent in multiple directions, reaching various consumer devices and also other producers or aggregators who will repackage and repurpose it to their customers.
Digital assets management technology is now very important for several factors like:

- the evolution of consumer products for capturing, storing, editing and distributing video
- the interest for P2P computing and for all the legal problems associated to protection of the rights of digital assets so to prevent the loss of copyrighted and branded digital asset
- the massive use of multimedia assets contents in many sectors
- the integration of rich media content into e-commerce web environments

Many organizations continue to maintain separate systems to manage creation and delivery of content: this separation is nonfunctional especially for cross media publishing because leads to duplicating the effort and increasing the complexity associated with quality controls. During their life cycle, documents are often revised and amended, and it is difficult to maintain consistency across media while maintaining all the different versions synchronized.

All main companies use a CMS for the management of all their contents: these systems have served many purposes with varying degrees of success. At the same time there are new organizations whose business is totally focused on delivering of media contents and old organizations starting to approach to new generation contents and to realize that digital contents represents an increasingly important element of their business.
All the considerations about cross-media management lead to the need for a content-management framework that can accept as input content that is authored with a variety of tools and prepare that content for publication according to the organization targets. Cross-media publishing means to prepare and publish content in a complex and wide-ranging environment: content can be delivered not only through different media, but also to a single medium in different formats.

For the purposes of digital asset management, CMS has to be more than a system to keep track of the contents of the website and to let non-technical users publish to a site using template structures. The best solution for content management problem is to adopt a system able to administrate contents including digital assets, facilitating an effective and efficient management of the information, organizing and repurposing those assets, streamlining costs and enhancing revenues.
If all the necessary functions for multimedia content management are available into the CMS, company business benefits from a common user interface, a good processes integration, and the ability to define a set of administrative and security rules ensuring user authentication and access control. The integration of legacy systems in the content management process allows publishers to leverage existing intellectual property on-line and, consequently, to better capitalise their assets.
Following these needs, the general trend of the market is to include digital asset management into the existing CMS (of both commercial and open source solutions) or to offer at least a limited set of related functionalities.

The life cycle of cross media content has three main phases:

- content creation
- content management
- content delivery

These three phases are interconnected through a process of data conversion and exchange. XML standard can be very important for this process because allows to identify a common format from which the content can be repurposed across media.

CMSs usually provide authoring tools but these are typically designed for text contents and are not suitable for the broad set of content types that can be produced and deployed on line. Content can be created with an appropriate tool optimized for output in a specific format. The organizations typically have a large amount of existing contents to be maintained and updated. Furthermore there are often contents provided by third party in particular formats. All these different formats lead to the need of having a unique standard (such as XML) and related tools for managing all types of contents: this allows generating and transforming contents into a common format and preparing it for inclusion in the content repository. For all these reasons it’s very important for internet distributors to adopt a CMS supporting XML standard.

Many organizations are adopting XML-based enterprise tools and architectures to provide a common infrastructure for all contents. Adopting an open standard can lead to the integration with other actual or future systems based on open standards. As a consequence, use of open standards can also mean access to open source software and decrease of organizations cost for IT.

Digital asset files may be local or remote: information on digital asset, such as its metadata description, is always available to the CMS.
Starting from this extended concept of "contents", CMS has to allow the content to be archived, searched and retrieved. It’s also important to grant permissions to be added to stored data, leading to the possibility of digital rights management (DRM).

The ideal CMS for a distributor of media contents has a set of core functionalities not strictly dependent from multimedia assets:

- *The repository*
  Builds a representation of the content utilizing a relational database or file system, or some combination. The repository generally allows versioning, categorization and upload/download of information.

- *The metadata*

Metadata contains description of the contents such as: descriptive information, administrative data, version information and often some kind of information on the relationship between objects. It is very useful to have an index on metadata information.

- *The search engine*
  To perform searches against the metadata index and the contents stored into the repository.

- *The access control system*
  It is important to have some roles defining who can see or do something on the content.

- *The workflow engine*
  This feature allows defining task and time scheduling of predefined process.

However an internet distributor of multimedia content needs some peripheral tools and processes other than the above basic tools such as:

- Integration with authoring tools of multimedia contents to allow seamless access to the repository;
- Repackaging assets for reuse;
- Image and video manipulation/transformation, to allow resizing and conversion processes eventually requested by the final users;
- Image recognition tools for visual searches;
- Applying metadata to store specific information about the digital asset, other than creation and versioning information;
- Beyond classical text indexing, also video indexing;
- Watermarking of images;
- Advanced rights management and usage tracking;
- User profiling;

Handling cross media contents into a CMS generates the idea of a searchable multimedia repository under the template-driven publishing engine. Content repository is more than a simple storage of data. It implements content services such as: author based versioning, full textual searching, fine grained access control, content categorization and content event monitoring.

Metadata becomes essential when indexing, searching and retrieving desired content especially for cross-media content: it can be used to describe the behavior, processes, rules and structure of the data, not just descriptive information. These elements are important when developing a good metadata strategy for content search and retrieval, enterprise content management, and dynamic content delivery, because they determine not only what the content is, but the consumer, the creator, the delivering process and the way to use it and any temporal limit when needed. Metadata enables content to be retrieved, tracked, and assembled automatically, and makes content accessible.

The management of the intellectual property rights is a prerequisite for the adoption of a CMS for an internet distributor: delivery of cross media content leads organizations to the need to access to technologies that enable to license use of their assets. Companies need to adopt business models that enable them to increase their revenues income and reduce the costs associated with creation and management of the assets. This is a very critical issue: many institutions recognizing the risks posed by piracy, including loss of income and control of their content, have decided to not approach to internet distribution.

The evolution of Digital Rights Management (DRM) is driven by the need to protect property rights and royalties. Other important issue is protecting privacy, confidentiality and first of all asset integrity. The support for a DRM would store basic ownership information and process rules for assets such as price, duration of license, frequency of access, type of use, type of rendering and the faculty to repurpose the assets and to transfer them to other users.

Metadata and DRM are complementary: the metadata creation process is the connection for integration between rights management systems and CMSs. As with all other types of metadata, it is desirable to have an automatic creation process avoiding relying on manual input.
The simplest way to automate the creation of rights metadata at ingestion or creation time is to program the CMS to use default rights metadata settings according to company policy.
An advanced use can be done using the CMS to insert rights metadata depending on particular information such as the type of the content, the tool used to create the content or editing it and other specific data related to the workflow routing. When no automation is possible, the only solution is that the CMS vendor integrates a template-based rights editor into the ingestion process, so that a user can fill in the appropriate rights on a case-by-case basis.

A company can achieve even more advanced ways of automating the creation of rights metadata in a CMS if it uses systems for tracking business rights, such as contracts with content creators and other sources of content.
In an Enterprise Content Management Systems (ECMS) rights metadata have to include this type of information typically found in corporate systems:

- File permissions (read, write, etc)
- Access control list for the resource
- User and/or group roles

Rights metadata can also be used for determining which contents will be made available to which users and under what conditions. An internet distributor can use other types of metadata such as keywords generated by a categorization tool for placing each content item in the appropriate location in the Web site.

In the classic B2C commerce, the distributor takes content file and metadata and creates a package that is decrypted on the client side by a hardware or software controller. Simple rights metadata could be stored in a CMS directly. Media companies could need more sophisticated integration between content and rights management: these organizations often maintain “product catalog” systems that contain product metadata. A given item of content can be included in more than one product: every product has its own metadata because different products can be intended for different type of customers and/or under different usage terms.

In summary, ever since network-based distribution of digital content became a reality, content owners and distributors have been searching for content management and distribution solutions that are truly integrated, enable them to pursue their business models and keep up with the latest technology ensuring that the contents rights are respected for both legal and economic reasons.

3.1.3 State of the art (completed)

In the previous sections, a general description of a typical content management system, its characteristics and features, and how it can be used to improve the business of an organization has been presented.
The companies that currently are in the business of managing and distributing cross media content over Internet, have the specific needs that have been previously analyzed and reported.
Currently there are literally hundreds of content management systems on the market, all having different capabilities and strengths. This is the nature of a rapidly changing business area: while there are a lot of good

products, there is very little consistency between vendors. Every organization has a unique set of requirements for a content management system, and there is no 'one-size- fits-all' solution.

This section presents an overview of the more popular and spread CMSs taking into account general features and giving special attention to specific issues related to the Internet distribution of media contents. For each CMS reviewed, this section also focuses on these points:

- platforms and development languages
- specific features and functionalities
- supported databases
- DRM and metadata support

The overview starts from open source CMS solutions and then surveys commercial products.

3.1.3.1 Open Source CMS

MySource Matrix

MySource Matrix, the successor of MySource, is an open source Web site and intranet content publishing and management system developed by the Australian Open Source developer Squiz.net.
MySource Matrix is written in PHP and currently runs on Apache Web server. The internal Database engine used is MySQL (but PostgreSQL is supported as well). The platform on which this CMS may be set up are wide start from UNIX, trough Linux, Solaris, Windows 2000/XP/2003, ending with MacOS X.
It is a browser-based system, easy to use also for technically unskilled users: it includes a WYSIWYG simple editor interface available through a Java enabled web browser.

MySource Matrix allows the use of design templates and a flexible management of permissions for users of user group: all content may have access privileges. Read, write and administration permissions can be granted or denied to the public or to a user. Permission can also be restricted according to the status of an asset in the workflow process.

Every type of multimedia file is supported. Other important features are version control, integrated cache, workflow and multilingual support. It is also possible to have additional functionalities such as a module for indexing and search over the site pages. Search mechanism is improved by addition of metadata, important information especially for digital assets.

There is only a partial support for delivering of contents on different channels and technologies such as Mobile Phohes, WAP, digital TV, etc. : new functionalities need to be added.

Product’s features

Product Name:	MySource Matrix
Organization Name:	Squiz.net
Company Web Page:	http://squiz.net
Product Web Page:	http://matrix.squiz.net
License:	MySource Public Licence
CMS Type:	General CMS
Platform:	UNIX, Linux, Solaris, Windows 2000/XP/2003, MacOS X
Web Server:	Apache
Programming language:	PHP
Database:	MySQL (PostgreSQL supported)

Main standards:	XHTML

Mambo

Mambo is a CMS distributed under the license GNU GPL that can be used for building simple websites or complex corporate applications. It is written in PHP, supports MySQL database, uses Apache and IIS web server and is available for Linux, Unix, Windows (NT/2000/XP) and Macintosh OS X servers.

Mambo is very flexible, scalable and allows to rapidly deploy professional websites. Ease of use is due to a powerful template system and a WYSIWYG content editor.
Mambo includes a page caching mechanism, provides extended user profiles, multilingual support and media (images, documents) upload and management. Mambo users are able to set automatic publishing and expiration dates. To organize the contents this CMS adopts the concept of sections, categories and items.

Mambo (Media Assets Management by Objects) is a CMS built for the management of graphics and digital image media. It is possible to archive media assets and tracking their usage and creative rights (DRM), grant access approval, facilitate transfer and handle financial transaction. Mambo provides tools for real-time transformations from one digital format to another and for the generation of thumbnails. The CMS organizes vast repositories of digital assets and metadata, enabling efficient indexing, retrieval and re-use of these valuable brand assets.

Mambo is a very featured system, but has only a limited version control. It also lacks a workflow engine.

Product's features

Product Name:	Mambo
Product Web Page:	http://www.mamboserver.com
License:	GNU GPL
CMS Type:	General CMS
Platform:	GNU/Linux, Windows, Mac OS X
Web Server:	Apache, IIS
Programming language:	PHP
Database:	MySQL
Main standards:	XHTML, CSS

Apache Lenya

Apache Lenya is an open source system, licensed under the Apache Software License and written in Java. The product is a relative young so comes with all the features an user can expect out of a modern CMS such as revision control, workflow management, scheduling, separate staging areas.

Lenya has an XML-centric architecture that allows for content delivery targeted to several devices capabilities, avoiding data lock-in. It is on top of Apache Cocoon Project which is a Web development framework providing a full set of libraries to build web applications using XML pipelining. A Cocoon application is designed to perform a sequence of actions on a given XML content (e.g. translate to HTML) before delivering it to the client. Lenya takes advantage of this framework storing all contents has to be managed as XML arranging it on the fly when required by clients. Obtained results are stored in a cache in order to reuse them, if needed.

For the authoring activities, two WYSIWYG editors with validation engines are provided, one for XML and another one for HTML. Editing pages involve revision control: whenever a page is edited a new version is created and, if needed, revert to any previous revision of a page is a feature. The authoring environments are multiuser compliant performing automatic lock to contents.

Apache Lenya workflow can be customized and it supports authoring, staging and live distinct areas. Workflow events can be scheduled: publication and archiving (deactivating) of pages takes high advantage of this feature. Multiple languages management is supported and have no impact on the site structure. Assets such as images can be managed and associated with pages. To each site managed page, a set of specific features is provided adding "ad-hoc" graphical tabs to have immediate access to metadata, assets, workflow status, revisions, access control and scheduling activities. A special site view tool is also provided, allowing easy move, copy, rename, archive, delete individual pages, or parts of the site.
An integrated search engine (Apache Lucene) allow full text and field search.

Lenya support user authentication via LDAP and the site (or parts of it as well as the single page) can be protected by SSL.
Apache Lenya doesn't support digital rights management.

Product's features

Product Name:	Apache Lenya
Organization Name:	Apache Software Foundation
Company Web Page:	http://apache.org
Product Web Page:	http://lenya.apache.org
License:	Apache SL
CMS Type:	General CMS
Platform:	GNU/Linux, Windows, Mac OS X, Unix
Web Server:	Apache
Application Server:	Tomcat
Applic. Framework:	Java
CMS Framework:	Cocoon
Programming language:	Java
Database:	MS SQL Server, MySQL, Oracle, PostgreSQL
Main standards:	LDAP, RDF, WebDAV, XHTML, XML, XSLT

Plone

Plone is a CMS created in 2001, licensed under a GNU General Public License and built on top of ZOPE (an open source application server, specialized in content management, portals, and custom applications).
Using Plone to create and manage a site is the more common situation, but the system can run as a document publishing system as well.

Plone comes with a WYSIWYG editor: to improve the produced markup pages it's possible to configure the editor to use several validation and cleaning tools. Plone has default templates: the default templates actually validate. With standard templates also set-of-page-templates (called "skin") are provided, one of them is a tableless skin. This CMS comes with an internationalized interface and is highly customizable and extensible: many add-ons (called Products) are available. In particular ATAudio, one of these products, handle audio files (currently only MP3): it support WebDAV to easy drag-n-drop uploading of audio files to the managed site.

To customize Plone, knowledge of the basics of Python, the Zope's templating language TAL and Zope Management Interface (ZMI) is highly recommended. Plone CMS has high platform compatibility: runs on Windows, Mac OS, Linux, Solaris and BSD. Plone's administrative interface also works with just about any web browser (older ones too, if needed).

Plone provides workflow controls and access control list for every item to decide who has access to that item and what the can do with it. Almost any type of data can be managed and stored. It is possible to use metadata but not to manage digital rights.

Product’s features

Product Name:	Plone
Organization Name:	Plone
Company Web Page:	http://www.plone.org
Product Web Page:	http://www.plone.org
License:	GNU GPL
CMS Type:	General CMS
Platform:	GNU/Linux, Windows, Mac OS X, Solaris, BSD
Web Server:	Apache
CMS Framework:	Zope
Programming language:	Python
Database:	Proprietary
Main standards:	XHTML, CSS

OpenCms

OpenCMS is a website content management system that can be used also without knowledge of HTML. An integrated WYSIWYG editor with a user interface similar to popular office applications helps the user creating the contents, while a sophisticated template engine enforces a site-wide corporate layout. The work environment is browser based: the OpenCms software is installed on a web server and the users can access the system with their browser.

OpenCms is a solution developed by the OpenCms group and using the LGPL Open Source license. This CMS can easily be integrated in existing environments and platforms because is based on Java and XML technology: it is written entirely in Java and the OpenCms page contents are stored in XML files. OpenCms supports various template mechanisms and JSP integration.
OpenCms provides a powerful mechanism for workflow management and an integrated system for users and permissions management: every resource has an associated owner and access right and is possible to create protected areas. All contents is maintained in "on-line" and "off-line" projects that can be approved and tested before publishing.

OpenCms supports the import of almost all common file types and the management of many digital assets.
The system is fully database driven and the managed website will be generated dynamically from the database: a page caching mechanism is used to increase performances. An integrated scheduling system can be used to trigger automatic processes. OpenCms can export the whole content repository, or portions of it, using an XML based format. Additional meta information like properties or access permissions are also exported.

Using the OpenCms module mechanism various full text search engines can be plugged into OpenCms to search the document stored into the content repository. Every resource in the content repository can be enriched with meta information to store various properties.
OpenCms offers multilingual and version control support.

Product’s features

Product Name:	OpenCms
Organization Name:	OpenCms
Product Web Page:	http://www.opencms.org
License:	GPL
CMS Type:	General CMS
Computer Platform:	Linux, Windows, Mac OS X, Solaris e BSD
Database:	MySQL, Oracle, MSSQL
Web Server:	Tomcat, Apache, IIS

Applic. Framework:	Java
Programming Lang.:	Java
Main standards:	XHTML, XML

Typo3

Typo3 is a content management system for multiple independent Websites and is a free software published under the GNU GPL. It was developed since 1998 and is written in PHP4 scripting language combined with MySQL.

Typo3 is platform independent. The system consistently separates Websites content and structure.
The user has no software requirement but one of the all modern graphical browser can be used to be able to edit the site project. It has dynamic graphical context menus interface features and only trough RTE supported by Microsoft IE on Windows platforms has a WYSIWYG tool to insert images and links while formatting text.

Typo3 has wizards to add customizable items as tables, fill forms, etc and there's no limitation of content types the user is able to add to a Webpage filling specific forms. Typo3 has features to resize and convert pictures from most different formats. It has a complete digital asset management system including the use of metadata. It lacks in DRM to achieve a secure distribution of contents.
The user administration makes it possible to manage users and groups able to access, to specify individually who may edit or delete specific content such as pages and objects. Password submission and backend storage are md5-encrypted. For administration sections but also for intranet and extranet it is possible perform filtering IP-based.

Typo3 support standard Webserver SSL to transfer encrypted data.
A version control system is provided together with a cross-format search engine.
Typo3 is extensible as users are able to integrate own or third party backend PHP functions and classes as well as to perform data processing with own ad-hoc PHP scripts.

Product’s features

Product Name:	Typo3
Organization Name:	Typo3 Association
Company Web Page:	http://www.typo3.com
Product Web Page:	http://www.typo3.com
License:	GNU GPL
CMS Type:	General Enterprise CMS
Platform:	GNU/Linux, Windows
Database:	MySQL
Web Server:	Apache, Microsoft IIS
Applic. Framework:	PHP4
Standards:	XHTML, LDAP, SSL

Drupal

Drupal is a PHP-based content management system for personal and corporate Websites. Originally a tool for managing community web portals and discussion sites, it provides lots of features to support them offering specific feature to perform blogging, news aggregation, resource directory and forum sites content management. Drupal includes also features to manage collaborative authoring environments, picture galleries and newsletters.

Drupal has a modular code architecture and the core functionality are provided by a standard set of modules providing features to edit and going on-line standard HTML pages using a web browser, 'book' feature to manage collaborative environment in which authorized people may contribute to content, workflow tools for content publication depending on review phases, discussion forum specific features providing thread and multi-thread forum structure, polls tools to collect and aggregate opinion on a defined topic, features to manipulate RSS format allowing to pull in RSS news feeds publishing them as HTML pages either to provide on the front page a feed of the contained items, tools performing search across site content, logging management on events and errors and dynamic content caching features.

On the Drupal Website there are hundreds of community ready-to-use modules more, each one adding different set of features. Some of the more useful add-on modules are PDFView (to convert content to PDF format), Flexinode (to be able to define custom content type), Typecat (to allow a category ordered content list), HTMLArea (to add a WYSIWYG content editor and tools), Webform (to add a webform nodetype to a Drupal site project), Video (to allow adding QuickTime videos to a Drupal site project with embedding format to enable video downloading).

The modular code structure makes Drupal highly customizable on specific user needs and Drupal's theme system put on distinct levels the content and its publication. Templates are built from standard HTML and PHP scripting.

Drupal has version control system features to keep tracks of content information updates and providing roll-back capability on them. It doesn’t seem to have a DRM system.

<u>Product’s features</u>

Product Name:	Drupal
Organization Name:	Drupal
Company Web Page:	http://www.drupal.org
Product Web Page:	http://www.drupal.org
License:	GNU GPL
CMS Type:	General CMS, Framework, Front End, News Portal, Blog
Computer Platform:	GNU/Linux, Windows, Macintosh OS X, BSD, Solaris
Web Server:	Apache, Microsoft IIS
Applic. Framework:	PHP
Database:	Microsoft SQL Server, MySQL, ODBC, PostgreSQL
Main standards:	LDAP, RSS, XHTML, XML

eZ publish

eZ publish is a content management system and a development framework written in PHP by eZ system. It can be used without any PHP programming knowledge and provides functionalities for web publishing, intranets and e-commerce solutions. The included framework is a powerful tool for further extensions and customizations.
eZpublish is available under GPL license for building a web site and under professional license for developing commercial software.

There is a separation between content and design: layout changes can be made by a template structure and easily applied by simple modifications. eZ publish also provides multi languages support, versioning system and workflow engine.

eZ publish uses an object oriented technology handling every type of information such as documents, video files, images or other: the contents are managed using a graphical user interface called *administration interface*. Contents are structured and stored into a database as XML: only binary files and images are stored on the filesystem for fast access.

A datatype is the minimal storage entity: there are many default datatypes that can be used to build complex structures describing the content, such as XML text, image, binary file etc.; it is also possible to create new datatypes.
eZ publish supports almost any type of content, multimedia and document type and allows, using the development framework, the extension to other special formats. This CMS supports all main audio, video and images formats and includes tools that automatically handle and scale images for different needs: the contents can be transformed so to be published in multiple formats, transferred and repurposed.

eZ publish has a powerful e-commerce engine that can be used for catalogues, B2B or content sales and includes an integration interface for payment systems.
There is an integrated search engine allowing indexing and search over all published contents.

eZ publish allows addition of metadata, role based access control and digital rights management.

<u>Product's features</u>

Product Name:	eZ Publish Content Management System
Organization Name:	eZ system
Company Web Page:	http://ez.no
Product Web Page:	http://ez.no/ez_publish
License:	GPL and Proprietary
CMS Type:	General CMS, Enterprise CMS, Framework
Platform:	GNU/Linux, Windowd, Macintosh, Unix
Web Server:	Apache
Programming Language:	PHP
Database:	MySQL, Oracle, PostgreSQL
Standards:	LDAP, RSS, SOAP, WebDAV, XML, XML-RPC

Ariadne

Ariadne is a multilingual Web Application Server and Content Management System, built entirely in PHP, available for free under the GNU GPL. It looks like Zope, a Python based CMS.

Ariadne provides a comprehensive user interface including a WYSIWYG HTML editor and it supports MySQL and PostgreSQL. The latest version also contains experimental support for Oracle 10 and scriptable workflow, new modules like multi-part pages, SOAP, XML-RPC, PDF, in-page editing, enhanced full-text searches, support for external user management using LDAP, Active Directory or Novell NDS.

Ariadne stores and retrieves contents in the database by simulating a file system. The stored objects can be accessed via filesystem-like calls (e.g. ls, find, get).
It allows to separate content, layout and logic level and to assign different people to manage each part, it will combine all these parts dynamically at the visit time.

Ariadne provides a customizable user interface and it is possible both to create a completely custom UI and select one of three pre-built user interfaces. The default UI is mainly based on the Windows XP Explorer.
Ariadne offers all the advantages of Object Oriented Programming including inheritance and polymorphy. Objects can have multiple templates containing a combination of text elements (HTML, XML, RTF and others) and PHP code, each template can be defined by users via web through the Ariadne management interface.
Templates can also include information and special operations through a scripting language called PINP, that is a safe subset of PHP which prevents access to PHP functions referring the file system or database directly.
Furthermore it is possible to create a website by editing in-situ, this means that it's possible to browse the sites and to modify them, just switching to edit-mode and updating the pages in a WYSIWYG editor.

In addition there is an FTP server built into Ariadne that allows users to upload or download a large number of files into the CMS, thus programmers and designers can use their preferred editing environment to create their templates and then upload them into the CMS directly.

Product's features

Product Name:	Ariadne
Organization Name:	Muze
Company Web Page:	http://www.muze.nl/
Product Web Page:	http://www.ariadne-cms.org/
License:	GNU GPL
CMS Type:	General CMS
Platform:	GNU/Linux, Unix, Windows
Web Server:	Apache, IIS
Application Server:	mod_php
Applic. Framework:	PHP
Programming lang.:	PHP
Database:	MySQL, CoPostgreSQL, Oracle
Main standards:	LDAP, XML, SSL

3.1.3.2 Commercial CMS

Vignette V7

Vignette V7 is a commercial suite of content management products developed by the Vignette Corporation (http://www.vignette.com). It is an enterprise CMS providing an integrated platform of applications and Web services aimed to enabling organizations to manage content, build portals, integrate content and enterprise applications, improve process management and perform analysis and reporting.
Designed around an open, standards-based Java architecture, Vignette V7 provides an easy integration with existing enterprise standards and platforms. It is available for all platforms (Windows NT/2000, IBM AIX, Sun Solaris, Linux, HP-UX) and can integrate data from a large number of applications as well as databases (Oracle, Microsoft SQL Server, Sybase, IBM DB2, ODBC, MySQL, JDBC).

Vignette V7 comprises six application services: organizations may purchase products individually from these six service categories or pre-bundled in suites to meet their specific needs. They are:

- Vignette V7 Content Services
  This solution provides the ability to access and manage content in almost any format, and publish the most appropriate content to the right user at the right time. It includes library services, content type modeling, workflow, taxonomy, and search.

- Vignette V7 Portal Services
  Vignette V7 Portal Services allows to quickly configure and deploy robust portals and applications. Includes modules for email, ERP, CRM and calendaring.

- Vignette V7 Integration Services
  Vignette Integration Services allows integration of structured and unstructured content from different sources such as existing enterprise applications or data stores.

- Vignette V7 Collaboration Services
  Vignette V7 Collaboration software allows colleagues, customers and partners to securely communicate and share knowledge, documents and information.

- Vignette V7 Process Services
  The Vignette V7 system was designed to enable technical and business users to take part to the process of content creation, management and delivery. Vignette V7 Process Services provides a standards-based process workflow engine and graphical process modeler for building and deploying business processes across the enterprise application infrastructure.

- Vignette V7 Analysis Services
  Vignette V7 provides automatic, real-time report and analysis tools to measure the success and effectiveness of web initiatives, web sites and portals.

Vignette V7 also includes page and content caching as well as load balancing on the application server and a good digital asset management.

Vignette has uses ICE (Information & Content Exchange), a protocol that supports content syndication, i.e., automated piecemeal licensing of content over the Internet. ICE began as an effort within Vignette Corp. to create technology that enabled Vignette's customers to exchange content. Instead of building a proprietary product within the company, Vignette decided to create an an XML-related standard for DRM in collaboration with some customers. In ICE, two types of software communicate with each other: syndicators and subscribers. Syndicators make collections of content available to subscribers by sending them packages periodically with instructions to add new content items to their collections or remove content items from them.

Product’s features

Product Name:	Vignette V7 Content Management
Organization Name:	Vignette
Company Web Page:	http://www.vignette.com
Product Web Page:	http://www.vignette.com/contentmanagement/0,2097,1-1-1928,00.html
License:	Proprietary
CMS Type:	Enterprise CMS, Framework
Platform:	Windows NT/2000, IBM AIX, Sun Solaris, Linux, HP-UX
Application Server:	BEA, Sun
Applic. Framework:	COM, Java
Programming language:	Java
Database:	Oracle, Microsoft SQL Server, Sybase, IBM DB2, ODBC, MySQL, JDBC
Main standards:	ICE, XHTML, XML

Stellent

Stellent Universal Content Management is an integrated, flexible, scalable content management solution providing lots of functionality and it is built upon a unified architecture that allows to manage Web content, document, collaboration, records, digital assets.

Stellent Universal Content Management can maintain growing volumes of content from a wide variety of sources and make that content accessible across an entire enterprise. Indeed the system manages and delivers dynamic content such as documents, graphics and Web pages to fixed content including scanned images, email and records (images, multimedia or other digital formats). Every content, regardless of its type, is stored in a Web-based repository for management, reuse an access features. A set of services onto the repository content are provided, including workflow, index/search, security, conversion and administration services. On these core services Stellent Universal Content Management provides a single product architecture offering Web content management, document management, collaboration, records management, and digital asset management functionalities with specific application modules.

The Web content management application module takes to the user a standardized platform to manage and publish hundreds of intranets, extranets and public Web sites, allows enterprise portal content applications integration, provides tools such as template-based web pages, multi-site management, re-usable content and XML chunks.

The document application module provides additional features for acquiring, sharing and protecting, version controlling both digital either paper documents and reports, e-mail notification workflow, content expiration and a customizable metadata model.

The collaboration module provides tools centered of collaborative web-based workspaces, allowing team space creation, customizable meta-data model, discuss bind to content forums and support for all file types.

The records module is able to classify any piece of content as a record and provides specific tools to manage records based just on active content, such as creating, removing and destroying, recovering and roll-backing features, automating the workflow process.

The digital asset module offers functionality to automate task and manage the media content, providing automatic thumbnail generation (for example used in the search and retrieve tasks) and specific additional functionality for image transformations, formatting and compression as well as for enhanced audio and video streaming.
Stellent also support digital rights management.

For all the above listed modules, common features provided are content release and expiration, full-text and meta-data search, version control and e-mail messages and attachment management.

Product's features

Product Name:	Stellent Universal Content Management
Organization Name:	Stellent
Company Web Page:	http://www.stellent.com
Product Web Page:	http://www.stellent.com/stellent3/idcplg?IdcService=SS_GET_PAGE&nodeId=64&ssSourceNodeId=7
License:	Proprietary
CMS Type:	General Enterprise CMS
Platform:	MS Windows, Sun Solaris, Red Hat/SuSe Linux, Hewlett-Packard HP-UX, IBM AIX
Database:	FAST (by Verity Inc.)
Applic. Framework:	Java, JSP
Database supported:	Microsoft SQL Server, Oracle Application Server, Sybase Adaptive Enterprise, Server, IBM Informix and DB2, Tamino XML Server
Web server:	Microsoft IIS, Sun ONE, Sun Java, Apache
Main standards:	XHTML, XML, LDAP

FatWire Content Server

FatWire Content Server, classified as enterprise CMS, is a commercial suite providing support for multisite management, content-centric applications and eBusiness initiatives. It is used for corporate Web sites, product marketing sites, partner extranets and employee intranets.

FatWire Content Server can be used to simply manage entire life cycle of the content including unstructured content, documents, and digital media assets: users can compose documents using familiar tools such as Microsoft Word and put them into the repository using Explorer. It is also possible to enrich content with metadata. The systems tracks revisions and versions. Simple and complex workflows can be applied at any step in the process to ensure collaboration between different organization areas and quality control.

The management of all content types, including digital assets such as video presentations, graphics, scanned documents, audio files etc. makes easy to associate and aggregate different types of content. It is possible to search over the digital assets of the repository: the system displays images thumbnails and asset metadata allowing to browse and find the right resource. FatWire software includes tools for conversion of many formats in HTML or XML format. A filtering technology allows automatic processing of digital resources such as extraction of thumbnails, resizing, extracting information from a video or an audio file. These powerful tools for digital assets management are suitable for delivery of multimedia resources to multiple Web sites and channels. Digital assets can be associated with personalization rules to help deliver the right content to the right person at the right time: there is a rights management allowing to decide who can use a content and under what condition. FatWire's e-commerce capabilities allow businesses to sell digital assets and include these assets in promotions, cross-selling, and up-selling protecting against unauthorized accesses: digital assets are delivered in secure mode.

FatWire Content Server is 100% Java and is integrated with the components of the WebLogic platform. FatWire Content Server also provides integration with Lotus Domino and Notes.

Product's features

Product Name:	FatWire Content Server
Organization Name:	FatWire Software
Company Web Page:	http://www.fatwire.com
Product Web Page:	http://www.fatwire.com/products/contentserver.html
License:	Proprietary
CMS Type:	General CMS, Enterprise CMS
Computer Platform:	Solaris, AIX, Windows, Linux
Web Server:	Apache, IIS, Sun Java Web Server
Application Server:	BEA WebLogic, IBM WebSphere, Sun JES, Oracle, Tomcat, JBoss
Applic. Framework:	Java, JSP
Programming Language:	Java, XML, XSLT, Javascript, JSP
Database:	Oracle, Microsoft SQL Server or MySQL, DB2, Sybase iAnywhere
Standards:	LDAP, SOAP, WebDAV, XHTML, XML, XSLT, SSL

Documentum

Documentum is a content management platform of enterprise level. It is a commercial set of packaged solutions suitable for creating, managing, storing and delivering almost any type of content. This CMS allows the creation of particular business rules used to guide the processes of revision and approval of any content. All the tools are supported by a browser based user interface.
Users can access the repository using a simple browser based user interface: available features include security controls, versioning and search tools.

Documentum allows to manage and store any kind of content such as documents, HTML and XML files, multimedia assets. All these contents can be stored in a distributed repository and protected by a strong access control.

Documentum also helps in the simultaneous publication of contents on different channels, supporting all common protocols for delivering content from source to destination. The content distribution can be done according to a scheduling mechanism or a specific client request: the content is transformed in the appropriate format to make it suitable for the final application used by the customer and finally is delivered.
The Content Transformation Services allow the transformation of the common desktop documents and rich media formats. Digital assets can be transformed and managed using the Digital Asset Manager: it is possible to obtain images preview and format conversion so to decrease image resolution or pass from commercial video to streaming formats.

Assets management allows to prepare the contents for delivering on different channels such as web, mobile phones or video broadcast.

Documentum is based entirely on open standards such as J2EE, JDBC, and JMS, supports XML standard and scales easily so can be used also for great amount of contents.

Documentum provides a complete mechanism for categorization and automatic creation of metadata gaining good results in indexing and searching processes. It can also capture metadata when users import contents into the repository.

Documentum provides a number of capabilities to make content management secure across the entire business value chain: it allows automatic delivery of contents from the repository to every farm or server around the world, using secure mechanisms such as SSL encryption. Sensitive contents can be stored in encrypted form into the repository or may be encrypted before delivering for external use. In this case, the external user must obtain a license key in order to use the content.
Documentum allows Digital Rights Management also in web publishing: contents can be accessed only from users or partners authenticated by the DRM server. In fact, the content management system is able to authenticate users and to manage user groups and permissions: it is possible to determine to whom a license should be granted and which features to enable for each user, according to the access control list associated to the document.

Product's features

Product Name:	Documentum
Organization Name:	Documentum
Company Web Page:	http://www.documentum.com
Product Web Page:	http://www.documentum.com/products/contentmanagement_products.html
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Any
Web Server:	Apache
Database:	Sybase, Informix, Oracle, SQL Server
Main Standards:	ICE, XML

IBM DB2 Content Manager

DB2 Content Manager is the solution developed by IBM for the management of almost any type of content including HTML, XML, electronic documents and audio/video assets. Its distributed multi-tier architecture is scalable and suitable also for geographically dispersed enterprises. DB2 Content Manager uses a XML data model and supports several hardware platforms, operating systems (including Linux), databases and applications: it can be integrated in existing software environment such as PeopleSoft, Siebel and SAP Applications. DB2 Content Manager also provides a set of APIs to make all its functionalities available to any kind of application.

IBM offers two versions of the DB2 Content Manager: the Express Edition for small organizations and the Enterprise Edition for big enterprises.
DB2 Content Manager provides the basic infrastructure for several solutions, allowing the management of multimedia contents, the control of the content life cycle, the records and the web content management. The solution for digital asset management and the required database offer support for the handling of audio, video and images in many file formats: it is possible to store huge amount of digital media with the associated metadata and to use the provided search engine to retrieve the desired content. In particular, the DB2 Content Manager VideoCharger is an advanced solution for the management of video assets: it allows the delivery of real time multimedia streaming over corporate intranets or over the Internet, using a peculiar technology

called ARMS (Adaptive Rich Media Streaming) and supporting the most popular standard formats such as MPEG-4 and Apple QuickTime.

DB2 Content Manager includes advanced features for the management of electronic office documents, allowing version control, cache mechanism for fast access and administration tools for the restriction of the accesses to reserved information. Document life cycle can be controlled using the workflow tool that enforces business rules at each step. It is also possible to create, store and manage records also using the workflow capabilities.

The XML support allows to capture, manage and store into a common repository documents adhering to specific XML schemas.

DB2 Content Manager also provides the infrastructure for the IBM Workplace Web Content Management solution that can be used to create, store and publish content sites and portals. The solution include the use of templates.

Digital rights management is performed by the EMMS (Electronic Media Management System), another product of the Content Manager family: it ensures secure distribution and protection of digital assets such as books, video games, music and software for the entire life cycle.

Product's Features

Product Name:	IBM DB2 Content Manager Enterprise Edition
Organization Name:	IBM
Company Web Page:	http://www.ibm.com/
Product Web Page:	http://www-306.ibm.com/software/data/cm/cmgr/mp/
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Sun Solaris, AIX, Windows, Linux
Web Server:	IBM HTTP Server, Apache, others
Database:	IBM DB2
Programming Language:	Java, C++
Main Standards:	XML, HTML,LDAP

Livelink

Livelink is an Enterprise Content Management suite developed from Open Text Corporation.

Livelink provides a robust, secure and scalable content repository allowing to share the information between different sectors of the enterprise, to control access permissions and content version. It is possible to capture and retrieve information from intranet, extranet and other document repositories and index and search this content by a search engine.
Livelink enables organizations to archive business SAP documents and data such as notes, invoices, orders, delivery notes, etc. It also offloads aged data and documents from the Siebel system while retaining access to the information.

The User Services supports person-to-person interactions, allowing different people to work together across applications and providing the necessary security controls.

Livelink simplifies and enhances all aspects of web content creation: authoring, review, publication, etc. There is a good integration with all the main desktop tools: Livelink functionalities can be accessed also within Windows applications.
The web content management server allows the creation and management of content for the publication in intranet o web sites. It also includes a template mechanism reducing the effort to maintain the web pages

adherent to the enterprise style and standard. Livelink provides a WYSIWYG content editor and a full integration with tools like Microsoft Word or Microsoft FrontPage.

Business process can be automate using a workflow mechanism ensuring that content approval and publication occurs on time.

The use of the Artesia technology provides digital asset management and supports a sophisticate metadata mechanism: it allows to access to all rich digital media of the enterprise and the underlying metadata information. Artesia is a easy to use DAM enabling to find, access, share, reuse, distribute, and archive all types of digital content.
The digital asset management also includes a Java API for integrating Artesia with other systems and customizing key behaviour.
Livelink doesn't support any digital rights management mechanism.

Product’s features

Product Name:	Livelink
Organization Name:	Open Text Corporation
Company Web Page:	http://www.opentext.com
Product Web Page:	http://www.livelink.com
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Sun Solaris, UNIX, Windows 2000, Windows NT, other
Web Server:	Apache, ISS, Sun iPlanet, Tomcat, other
Database:	MS SQL Server, Oracle, SyBase, other
Programming Language:	Java, VB, other
Main Standards:	XHTML, XML, other

Mediasurface

Mediasurface is an Enterprise Content Management software that helps customers to create, manage and publish content across their organization. It also provides functionalities for management and delivery with an architecture that supports delivery to any digital channel.

Mediasurface is easy to use and doesn't require technical skill for accessing and managing the content. Morello, the client office application for content management, offers a WYSIWYG tool allowing to easily create and edit content, build and organize new sites, managing templates, etc.
The CMS can be integrated with many back office application and content source using a wide variety of technical interfaces available.

Mediasurface provides multiplatform choices, user management and scalability. It manages documents and digital assets allowing metadata handling, transformation services, streamed media integration and multi-channel support. In particular, it offers a range of system based image manipulation capabilities, including the ability to resize, crop, reformat images.

All the content is stored in a Oracle-based repository. Content can be automatically classified, summarized and searched. Mediasurface uses a large number of interfaces to extract and receive content in many formats such as XML, JMS, .Net, etc.

Mediasurface manages all of the processes necessary for the full life cycle of the content: a workflow tool control the creation, approval, publication and archival rules for each content through to the complexities associated with management of the solution - including user access rights, security, structure and design. It doesn't integrate any digital rights management solution.

Product's features

Product Name:	Mediasurface ECM
Organization Name:	Mediasurface
Company Web Page:	http://www.mediasurface.com
Product Web Page:	http://www.mediasurface.com/product/
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Sun Solaris, Windows 2000
Web Server:	Apache, ATG Dynamo, ISS, Orion, Sun iPlanet, Tomcat
Programming Language:	Java, VB
Database:	Oracle
Main Standards:	XHTML, XML, other

QP7.Enterprise

QP7.Enterprise is an enterprise content management system developed by Quantum Art and is part of the QP7.Framework family of products that provides the capabilities to manage large, complex web sites, portals and intranet. This CMS combines the easy of use of an intuitive browser-based interface with the flexibility of a template development.
QP7 is based on Microsoft's .Net technology and managing design and templates for site content requires familiarity with the .Net platform.

QP7 include a set of applications to address specific needs of intranet management, government on line services, educational on line initiatives and corporate marketing strategies.

It is possible to create a set of role and permissions to control the access to the contents. The workflow mechanism allows the definition of processes for content creation, editing and approval.
QP7 also provides a versioning mechanism and a full text search on the individual content and the content of the entire site. Through the addition of virtual contents, use can link multiple content types and query third-party and outside tools and database for content.

QP7 allows use of metadata and digital assets management and digital rights management.

Product's features

Product Name:	QP7.Enterprise
Organization Name:	Quantum Art
Company Web Page:	http://www.quantumart.com
Product Web Page:	http://www.quantumart.com/qp7enterprise
CMS Type:	Enterprise CMS
Platforms:	Any
Database:	MSSQL, MSDE
Programming Language:	C++, C#, ASP, ASP.NET, PHP, HTML, JavaScript
Web Server:	Any
Main Standard:	XML, XHTML

Tridion R5

Tridion R5 is an advanced XML Enterprise Content Management solution for managing large amounts of content for Internet and intranet environments. It enables enterprises to create, manage, distribute and deliver content to multiple websites, on line applications or channels (such as Web TV, PDA, mobile devices, ect.).

Content can be created using the most popular desktop applications: an advanced browser based interface allows to create, manage and publish the content. Contents already present in the enterprise repository can be easily integrated and re-used: the ClientConnector tool makes the Tridion repository transparent to the development applications.

Tridion content management includes the support for complex XML data types and a multimedia search engine allowing the search of words also in binary files such as Word documents, pdf files etc. BluePrint Management functionality provides centralized control of brand and content across multiple Web sites and alternative channels, such as print, e-mail, PDA, mobile, and online applications.
Tridion also includes an advanced workflow tool for designing business processes, metadata facilities and group access rights.

Once the content has been created and managed, Tridion Content Distributor allows non-technical users to publish and distribute it to any location and application in the right way. To guarantee the delivery of the content, transport on the network is transactional, allowing rollback in case of errors.

Delivering contents via the Internet leads to the need of a Digital Rights Management. Tridion uses the DRM technology supplied by the partner SealedMedia: content owners can control sealed content even after it has been downloaded by the end user. This DRM tool allows to impose constraints on the delivered contents such as limited functionality previews, time limited subscriptions, concurrent usage restrictions, etc.

Product's features

Product Name:	Tridion R5
Organization Name:	Tridion
Company Web Page:	http://www.tridion.com
Product Web Page:	http://www.tridion.com/Products/R5/Overview.asp
CMS Type:	General CMS, Enterprise CMS
Platforms:	IBM AIX, Linux, Sun Solaris, UNIX, Windows 2000, Windows 9x, Windows NT, other
Application Framework:	Java, .NET
Database:	IBM DB/2, MS SQL Server, Oracle, Tamino, X-Hive
Programming Language:	Java, VB, other
Web Server:	Apache, ATG Dynamo, ISS, Jetty, Orion, Resin, Sun iPlanet, Tomcat, other
Main Standard:	XML, LDAP, XHTML

Interwoven

Interwoven is a commercial full suite of content management software solutions developed by Interwoven Inc., a leader player in the CMS market arena. Interwoven is an enterprise CMS addressing the whole content management needs, providing an integrated platform of applications to enable collaborative creation, management, publishing, delivery, archiving of big volumes of digital content, both within and beyond the enterprise borders.

By "WorkSite" product the Interwoven CM platform offers comprehensive collaborative document management functions as user level security, check-in/check-out, version control, audit trails, archiving, categorization, full text, and meta data searches. All the documents, including scanned images, and e-mail messages and attachments, are stored in a secure centralized repository. The access to the repository is granted by browser-based user interfaces or in offline mode trough document management applications fully integrated with Microsoft Office applications (integration with Lotus Notes and Novell GroupWise is provided by available extensions).
Interwoven provides a good content integration for the management of data stored in different systems or repositories using various formats. However, there is also the Interwoven Developer Suite that can be used to

develop, customize and integrate other applications with the Interwoven platform: it includes a web services-based API and a developer server which provides a safe environment for testing new applications.

The "LiveSite" and "TeamSite" are instead Web content management products by which the platform enables users to easily create, publish and manage content-rich intranet and internet sites. The above products comprises page and site templates, a reference site, pre-built components to consolidate Web properties into a single managed environment and ensure Web consistency. The "TeamXML" is a product that adds XML capabilities devoted to make possible content reuse and publish in multiple formats for multiple channels.

Interwoven suite includes also a Digital Asset Management set of products of the "MediaBin" family: assets, such as images, audio files, videos, etc, can be stored and cataloged along with their related metadata. Media can be searched also on metadata. Images can be translated into various formats and it is possible to create thumbnails and watermarks. Trough the MediaBin interface, the users can easily import or ingest assets and add to them unlimited metadata fields. Interwoven also provides solutions for digital rights management a very strong workflow management tool.

Interwoven has a solution for marketing content management enabling the company to customize the offer of specific products to meet the needs of specific customers.

Product’s features

Product Name:	Interwoven Content Management System
Organization Name:	Interwoven Inc.
Company Web Page:	http://www.interwoven.com
Web Page:	http://www.interwoven.com
CMS type:	Enterprise CMS
License:	Proprietary
Computer Platform:	Windows, Unix, Solaris
Application server:	built-in (none required)
Databases:	built-in (none required)
Programming lang.:	Java, JSP, Perl, CGI, C++, JavaScript, HTML
Web Server:	IIS, Apache
Standard:	XHTML, LDAP, SOAP, WebDAV, XML

Microsoft Content Management Server 2002

Content Manager Server 2002 (MCMS 2002) is the solution developed by Microsoft to improve content management and publishing on Web sites. The solution is scalable and reliable because includes mechanisms for load balancing and failover. It is based on the .NET technologies and is suitable for organizations that want to reduce costs and make easier on line communication, allowing business users to create, manage and publish content.

The focus of the MCMS 2002 is to get non-technical users to create and publish content using easy-to-use tools including template mechanism: content can be handled using the Internet browser or the Microsoft Word application. The content management includes tools for content approval, workflow, scheduling of content refreshes or archiving, revisions tracking, addition of metadata and content indexing. It is possible to establish a set of rights groups and roles to decide who has permission to perform a given task on the content.
MCMS 2000 can be used in combination with the Microsoft Mobile Internet Toolkit to build Web site that can be browsed by mobile devices such as PDAs and Web phones.

The content is stored in a Microsoft SQL Server repository as XML, HTML and binary objects: contents and templates, or other design elements, are stored separately. The use of the XML format and the separation between the content and the presentation of the data, enhances the availability and reuse of the same content for a wide range of application, including Web services, and devices. Furthermore, organizations can easily share content with partners, suppliers or customers regardless of the specific platform.

Integrating Web content management and digital asset management, organizations can handle almost any type of enterprise content from the creation to the publishing process. Using Microsoft digital rights management system, content providers can protect music, video or other digital asset from unauthorized copying or uses. Microsoft DRM technology is based on an encrypting mechanism able to attach usage rules to the content so to determine who can use the content and under what conditions. There rules are transparent for the users unless he attempt to violate them: they generally prevent illegal copy or limit the number of times the content can be played.

MCMS 2002 provide a full integration with the Microsoft Visual Studio .NET allowing developers to use a familiar environment to extend or create new applications for web services.

Product's Features

Product Name:	Content Management Server 2002
Organization Name:	Microsoft
Company Web Page:	http://www.microsoft.com/
Product Web Page:	http://www.microsoft.com/cmserver/
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Windows
Web Server:	IIS
Database:	Microsoft SQL Server
Programming Language:	.NET
Main Standards:	XML, LDAP

ECMsuite

ECMsuite is an enterprise system developed by Componence for the management of contents for portals, extranet, intranet and e-learning. It is a scalable software suitable for small and large enterprises or organizations. The user-friendly and web-based environment allows easy interaction with the system also for non technical skilled users.

ECMsuite provides is based on open standards such as J2EE, allows a secure interaction with other back office systems and is platforms and databases independent. Content management uses a WYSIWYG interface and an automatic publication tool from MS Word, Excel and PowerPoint. Every piece of content is automatically tagged with a version number and it is also possible to rollback to previous versions when needed.
It is possible to automatic or manually generate metadata for search engines.

The standard configuration of the system includes a template management mechanism, a workflow process and a web content management tool that can be used to easily add extra functionality to a web site, such as a forum, a poll, a web-log, etc.

ECMsuite provides many security facilities such as granular privileges mechanism for the content access control and the support for encryptions.
The CMS allows to decide publication and archival date for the content and, eventually, it is possible to determine if publications are visible only for a selected number of visitors.

ECMsuite allows digital asset management, supporting all current standards for audio and video files such as Real Audio, Real Video, QuickTime, ASF, WMV, AVI, MP3 etc.

Product's Features

Product Name:	ECMsuite
Organization Name:	Componence BV
Company Web Page:	http://www.componence.com
Product Web Page:	http://www.componence.com
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Linux, MacOS, Sun Solaris, UNIX, Windows 2000, Windows NT
Web Server:	Apache, ISS, Orion, Tomcat, other
Database:	IBM DB/2, Hsdldb, Lotus Domino, MS Access, MS SQL Server, MySQL, Oracle, PostGresQL, Progress, SyBase, Lotus Notes, Tamino, X-Hive, Xindice
Programming Language:	Java
Main Standards:	LDAP, SSL, XHTML, XML

CoreMedia CMS 2005

CoreMedia CMS 2005 is an enterprise content management system developed by CoreMedia AG which can be integrated with many existing IT infrastructures and third party systems. The software is written in Java and is based on open standards such as XML, UML, SOAP, HTTP, WebDAV. It is compatible with all the leading databases (Oracle and IBM DB2), application servers (BEA, Tomcat and IBM WebSphere) and operating systems (MS Windows, Sun Solaris, IBM AIX, HP UX, Red Hat Linux and Suse Linux).

CoreMedia provides two different environments for the creation and the delivery of the content to end users, each with its own applications and databases: in the publication phase, a copy of the content is transferred into the delivery environment.

The editorial tools are simple, web-based and use a WYSIWYG interface. The CoreMedia provides a template mechanism and an integrated interface for the creation and transfer of content from common applications such as MS Word or Adobe Photoshop. It is also possible to automatically import content in structured formats like XML, CSV etc.

CoreMedia is able to manage every kind of content, including binary files or digital assets like videos, images and music. The entire life cycle of the content can be controlled using workflow mechanism: there are some predefined workflows and new ones can be defined to meet specific requirements. There is a version control and a repository management tool for renaming, move, restore, delete or publish the contents.

Content is stored in a neutral format and is converted in the target format (HTML, PDF, XML, etc.) only before the delivery: this is a very useful feature for the delivery of contents across multiple channels and digital devices such as website, mobile phones, PDAs, iTV, etc.

CoreMedia CMS 2005 uses CoreMedia DRM, the CoreMedia sofware for DRM, enabling business to protect content from unauthorized access. This DRM mechanism is based on Open Mobile Alliance (OMA) standard. Various content objects can be aggregated into business objects to be delivered to end users: during aggregation it is possible to include additional information such as product availability, or complex rules for access rights, filtering and personalization.

CoreMedia DRM offers three levels of security for digital rights: 'lightweight' protection for inexpensive content, 'secure' protection form higher-value content and 'trusted protection' for premium services. It is possible to use DRM mechanism to limit the use of the delivered content, for example allowing the customer to use the purchased content on a pre-selected number of PCs.

CoreMedia supports the superdistribution, a marketing tool using peer-to-peer data transfer based on the separation of encrypted content and usage rights (including an appropriate key): consumers can try the contents (music tracks, ringtones, videos, etc), using free previews (music tracks, ringtones, videos, etc), but the key and rights to use the content in full must be purchased separately.

Product's Features

Product Name:	CoreMedia CMS 2005
Organization Name:	CoreMedia
Company Web Page:	http://www.coremedia.com
Product Web Page:	
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Any
Web Server:	Apache, IIS, Tomcat, other
Database:	Oracle, IBM DB2
Programming Language:	Java
Main Standards:	LDAP, XML

Ektron CMS400.NET

CMS400.NET is a content management system developed by Ektron using the Microsoft .NET framework. It provides many out-of-the-box functionalities for an easy creation and publication of content on web sites, intranets and extranets. Using the .NET platform it is also possible to customize and extend the product features using Visual Studio .NET and Ektron ASP.NET server controls.

CMS400.NET includes Ektron's DMS400, a document management solution allowing also non technical users to create, edit, share, and publish many types of content such as Microsoft Office or PDF files, audio, video, etc.
eWebEditPro+XML is a WYSIWYG editor integrated in the system: it allows the creation of unstructured and structured contents: content can be easily created once and delivered to multiple channels.
CMS400.NET is an XML based solution: the power of the XML format can be used, for example, to enhance distribution on multiple devices or to store dynamic content presentation options. Content editing can also use template mechanism.

CMS400.NET allows the addition of metadata to the content: the indexing and search tool can use this information to optimize the engine and obtain good results.

A workflow suite provides complete processes management control, including scheduling of content for publication and expiration, content history tracking, rollback functions and approval chain. Content protection is achieved using an access control: it is possible to create user and/or users groups enabled to view a content.

CMS400.NET allows digital asset management and provides a full integration with Ektron WebImageFX, an images editing tool. WebImageFX has a browser-based interface and allows creation and management of digital images, offering features like resize, rotation, thumbnails creation etc.
The integration of WebImageFX also enhances digital rights management because content authors can watermark or copyright digital assets, protecting the integrity of the content to be delivered to end users.

Product's Features

Product Name:	CMS400.NET
Organization Name:	Ektron
Company Web Page:	http://www.ektron.com
Product Web Page:	http://www.ektron.com/cms400.aspx
License:	Proprietary
CMS Type:	Enterprise CMS

Platform:	Microsoft Windows 2000/2003/XP Professional
Web Server:	ISS
Database:	SQL 2000, Microsoft MSDE 2000
Main Standards:	SSL, RSS, XML, XHTML

Communiqué

The Day Communiqué content management application provides a comprehensive framework for intranet, Internet and extranet initiatives of an enterprise. Based on the Java technology, Communiqué is an integrated platform supporting the main standards such as J2EE and XML and providing multilingual support, scalability and security.

Communiqué allows the management of all digital business data, systems, applications and processes using a content-centric architecture bringing together contents from different systems, platforms and locations. The ContentBus creates a Virtual Repository containing the representation of data existing in many repositories or applications: the data is indexed and mapped and can be accessed in real-time regardless its original source or format. The ContentBus accesses different raw data and homogenizes it using the Content Services which is able to create a virtual layer adding versioning, access control and encryption. Other components used to easily deploy enterprise application on the web are Application Services and Connection Services. Application Services is an out-of-the-box functionality for content and portal management, digital asset management, collaboration and management of B2B applications and catalog. Connection Services allows to easily access the content wherever it is.

The Integrated Development Environment includes tools for content creation and management fully integrated with main business systems such as CRM and ERP. A simple GUI provides editors for many types of files and template creation. The systems has team development tools allowing workflow controls, version tracking, projects overviews, etc.

Communiqué provides some advanced capabilities such as load balancing, caching and secure syndication protecting privacy and confidentiality while still delivering content over the public internet.

Focusing on digital asset management, Communiqué allows users to store, search, manage, edit and repurpose all types of digital asset: images rendition can be done using integrated tools without the use of outside applications. Digital assets management also includes watermarking, DRM and availability to all authorized users.

Product's Features

Product Name:	Communiqué
Organization Name:	Day Software AG
Company Web Page:	http://www.day.com
Product Web Page:	http://www.day.com/site/en/index/products/content-centric_applications/overview.html
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	IBM AIX, Linux, Sun Solaris, UNIX, Windows 2000, other
Web Server:	Apache, ISS, Sun iPlanet
Database:	IBM DB/2, Informix, Lotus Domino, MS Access, MS SQL Server, MySQL, Oracle, SyBase, Lotus Notes, other
Programming Lang.:	Java, other
Main Standards:	SSL, XML, LDAP, RSS, XHTML

Ingeniux CMS

Ingeniux CMS is an enterprise web content management system developed by Ingeniux Corporation and built on a open-standard XML technology.

The XML architecture allow organization to create the content only once and reuse it for multi-channel deployment to different partner or devices.

Administrative tools provides access control for users and groups and allow to define unlimited permissions levels.
Non technical people can create and manage contents using WYSIWYG interfaces and template mechanism.

Ingeniux CMS provides a configurable workflow system for content tracking and automating tasks through publishing process. Content management also includes scheduled publish and archival dates, content approval process, addition of metadata and multiple site search.

Ingeniux CMS includes support for RSS format and syndication capabilities: it can query a database or capture syndicated information from any web site for which it has permissions and repurpose it for syndication to partners or others.

Ingeniux CMS can manage all type of contents including digital assets but doesn't include advanced functionalities for digital rights management.

Product's Features

Product Name:	Ingeniux CMS
Organization Name:	Ingeniux Corporation
Company Web Page:	http://www.ingeniux.com
Product Web Page:	http://www.ingeniux.com/x628.xml
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Linux, MacOS, Windows 2000, Windows NT
Web Server:	Apache, Sun iPlanet, Tomcat, other
Database:	MS SQL Server, other
Programming Language:	Java
Main Standards:	LDAP, SSL, XHTML, XML

FileNet P8

FileNet P8 is an enterprise content management platform providing the ability to create a single enterprise catalog across all of an organization's content repository, increasing the ease of use and the management capabilities. FileNet P8 is the base of the FileNet's family of ECM suites: Business Process Manager, Content Manager, Email Manager, Forms Manager, Image Manager, Records Manager, Team Collaboration Manager, and Web Content Manager. Each suite is a pre-packaged solution emphasizing a particular aspect of an enterprise content management and interoperating with all the others suites: customers can choose their solution and extend the installation in the future integrating other packages according to their needs.

Business Process Manager provides a workflow control on the business processes allowing to automate and optimize business operations: the business analysts can manage great amount of data, modify processes and apply new business rules immediately. Business Process Manager provides a scalable architecture able to support thousands of users and millions of transactions. Real-time and historical tracking of processes allow the analysis of the results and the consequent optimization of the processes.

FileNet Content Manager is an enterprise solution allowing the management of every type of digital assets including Microsoft Office documents, XML-tagged files, images, sounds, templates etc. It provides a configurable workflow tool, a versioning mechanism and support for content classification. Business users can create, find, manage and share content using the business and desktop applications directly integrated into the Content Manager environment. The use of metadata models ensures comprehensive search and retrieval tools.

Email Manager is an email management solution enabling to capture corporate email messages matching specific business rules: the email messages can be indexed and archived in a repository.

Forms Manager provides the ability to design, deploy and process electronic forms (eForms) connected to the business applications. The suite supports digital signatures and tracking for audit trails to help in meeting regulatory compliance requirements.

Image Manager is a suite for highly scalable and rapid access for end-users to fixed objects or content that is not intended to be modified such as documents or rich media. It securely stores great amount of data and critical information and provides high-performance information retrieval tools.

Records Manager allows to securely store, organize and administrate records: the information is retained and destroyed according to the selected policy.
Team Collaboration Manager promotes knowledge exchange and group decisions providing collaboration tools for member groups.

Web Content Manager is a suite designed to allow web content creation and management also for non skilled technical users, separating content from layout and using templates. The software enables the creation, approval and publication of web contents on multiple sites, in multiple formats and languages. Web Content Manager also includes version control, rollback mechanism, workflow and management of unstructured content (including paper documents, HTML, XML, rich media, PDF and email).

In conclusion, FileNetP8 delivers an XML Web Services-enabled framework for developing applications and managing information throughout the enterprise. FileNetP8 platform provides solution for digital asset storing but lacks specific tools for the managing and repurposing of digital assets. It includes digital signatures support in the Forms Manager suite but doesn't provided more sophisticated mechanisms for DRM enabling the protection of digital rights for the contents delivered to end-users.

Product's Features

Product Name:	FileNet P8
Organization Name:	FileNet
Company Web Page:	http://www.filenet.com
Product Web Page:	http://www.filenet.com/English/FileNet_P8/index.asp
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Microsoft Windows, Sun Solaris, IBM AIX, HP HP-UX, Red Hat Linux
Web Server:	BEA WebLogich, IBM Websphere
Database:	IBM DB2, Microsoft SQL Server, Oracle
Programming Language:	Java
Main Standards:	XML, SSL, XML

RedDot XCMS

RedDot XCMS is an enterprise solution including four different modules: Web Content Manager, Document Manager, Collaboration Manager and Business Process Manager. Customers can select the needed modules and add new ones when their needs change.

Web Content Manager is the solution for creation and management of web content: the module includes a simple interface and can be used also from non technical skilled users. The module allows the use of multiple versions of templates, in multiple languages and formats. User permissions can be specified through an administrative interface: Web Content Manager supports an authentication structure and can enforce content approval process ensuring that every content change follows the predefined workflow before publication.
Web Content Manager includes an Asset Manager tool used to store, catalogue and manage (cropping or resizing) all types of images; it is also possible to create images thumbnails.

Document Manager is the RedDot's module for storing and managing documents. It offers an integrity protection allowing the use of access rights for individual users or groups. Document can be created using Microsoft Office applications which are fully integrated. The Document Manager ensure documents integrity tracking every access and changes on the contents and ensure the version control. Adding metadata documents can be enriched with important information used from indexing and searching tools.

Collaboration Manager is the module used to create an on line secure environment to communicate and share information between internal and external people of an enterprise: it is possible to create shared workspaces where members of teams can share documents, communicate using emails and forums in a protected environment.

Business Process Manager supplies an advanced workflow control, allowing business processes automation. This modules interacts with Collaboration Manager: all assigned tasks and process are visible in each team members personal workspace.

Product's Features

Product Name:	RedDot XCMS
Organization Name:	RedDot
Company Web Page:	http://www.reddot.com/
Product Web Page:	http://www.reddot.com/products_enterprise_content_management.htm
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Microsoft Windows 2000 - 2003
Web Server:	Any
Database:	MS SQL Server, ODBC, Oracle
Programming Language:	.NET, PHP, Java, JavaScript, JSP, other
Main Standards:	LDAP, SSL, XML, XHTML

Percussion Rhythmyx

Percussion Rhythmyx is an Enterprise Content Management system is designed to meet the needs of mid-tier enterprise for the managing of Web and portal content, documents, digital assets, and scanned images.

The Web Content Management (WCM) module allows content creation and contribution using the most popular desktop applications: it is possible to use template, add metadata to the content and decide publish and expiration date. Contents, including media files, are then indexed and can be easily located through and integrated full text search engine.
Rhythmyx WCM can manage every type of content, including digital assets such as images and other media files. Digital assets can be created with the most popular tools (Adobe Photoshop, etc.) or with an inline image editor, enriched with metadata information, indexed and searched. The Inline Image Editor also allows content contributors to watermark or copyright digital assets achieving a DRM functionality even if limited.
The system automatically extracts metadata from each asset, renders images to any specified dimension.

Rhythmyx provides some out-of-the-box workflows and allows the creation of custom ones or approval cycles. Each workflow can automatically perform actions and send notification alerts or email messages. Content updates management includes a versioning control.

Rhythmyx is based on industry standard such as Java, XML, XSL and can be easily integrated with many IT environments; furthermore, the separation between content production and delivery allows the repurpose of the content over multiple channels including web sites, hand-held devices etc.

Product's Features

Product Name:	Rhythmyx
Organization Name:	Percussion
Company Web Page:	http://www.percussion.com
Product Web Page:	http://www.percussion.com/products/content-management/rhythmyx/
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Windows, Unix, Sun Solaris
Web Server:	Any
Database:	Oracle, Microsoft SQL Server, Sybase, IBM DB2 UDB
Programming Language:	Java
Main Standards:	ICE, SOAP, XML, XSLT

Terminalfour Site Manager

Site Manager is an out-of-the-box enterprise content management system developed by Terminalfour and based on open standards such as XML, Java, JDBC and HTTP. This solution can be easily integrated with old and new IT systems and manages the entire life cycle of the content allowing creation, editing, approval, security, publication, search, expiration and archival.

Site Manager is browser based and can be used without particular technical skills. This solution is suitable not only for web content management but also for intranets, extranets and content delivering through various channels.
Organizations can find, extract, present and share in secure mode information from different enterprise systems (such as SAP, ERP, CRM etc.) and databases.

There are three solution for content editing: a "WYSIWYG" interface similar to the MS Office applications, a dynamic editor and the eForm. The eForms functionality allows the creation of custom forms: users can fill these on-line forms the content of which is transmitted to the Site Manager and used to fill or update the repository.
Site Manager includes a “template wizard” for the creation of custom templates that can be used for the layout of the web pages and the addition of metadata.

Site Manager includes an advanced workflow to improve process automation: content publication can be scheduled for a later date and it is also possible to pre-schedule content for archival at a pre-selected date in the future.

Content editing and management is controlled by a secure rights and role model. Content is delivered in secure mode, using security features such as SSL encryption, PKI and password protection. Every document within Site Manager has properties associated with it, such as when and where it should be published to.

There is a strong version control designed to ensure that each version of the content is locked and cannot be lost. Site Manager provides a multilingual support: document can have a number of language version associated to it, each with its version number.

All actions are logged in a central database repository visible to the administrators: the full lifecycle of any piece of content can be traced and rolled-back through all of its versions.

Terminalfour Site Manager allows the management of multimedia content in almost any format.
Contents can be published to multiple sites or channels such as WAP, PDA, Digital TV, E-mail, SMS, etc.
Site Manager uses a relational database and a XML structure providing an ideal structure for content reuse, syndication, importing, exporting and publishing. Site Manager can also use the Apache Cocoon application for the repurposing of the XML content to various devices.
Terminalfour uses Verity technology to provide a keyword / meta-tag and full text search facility.

Product's Features

Product Name:	Terminalfour Site Manager
Organization Name:	Terminalfour
Company Web Page:	http://www.terminalfour.com
Product Web Page:	http://www.terminalfour.com/products/sitemanager/
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Any
Web Server:	Apache, ISS, Sun iPlanet, Tomcat
Database:	IBM DB/2, MS SQL Server, MySQL, Oracle, SyBase
Programming Language:	Java, other
Main Standards:	LDAP, SSL, XHTML, XML

Oracle Portal

Oracle Portal is a component of the Oracle Application Server developed to provide an integrated framework for content management on enterprise portals.
Oracle Portal provides a browser-based interface for the creation of the portal: content can be edited and published without programming or HTML knowledge. The use of templates allows to exploit predefined layout styles and to include default items or portlets. Web interfaces and portlets can also be developed in Java and integrated in the Oracle Portal environment. The Oracle Integration Server can be used as interface for third party applications such as SAP and Siebel. Oracle Portal supports the main standards such as HTTP, XML and SOAP that can be used for data exchange among remote applications of Portal instances. Furthermore, the Oracle Portal Developer Kit (Oracle PDK) allows the integration of standards-based applications and services such as Web services.

Oracle Portal offers an integrated multichannel access enabling users to access content from several different devices: it is possible to automatically transform the portal pages structure and content so to obtain a form suitable for smaller screen devices such as mobiles, PDAs and PocketPCs.

Oracle Portal offers a full integration with Oracle interMedia, a set of platform services for the management of multimedia assets. The interMedia tools allow to add media columns or objects to existing database tables and to easily search and retrieve multimedia data. A set of plugins also enables the development of video streaming services or other multimedia applications.

If the content needs to be reviewed and approved before publishing, it is possible to set an approval process with the related notifications to be sent to the approvers. The content can also be scheduled for automated publication and expiration. Using Oracle Workflow, information can be routed according to predefined business rules for modeling and automating business practices.

Oracle Portal uses the on-demand grid computing capabilities integrated into the Oracle Application Server. Grid computing is a software architecture developed to enable customers to efficiently use a large amount of different hardware resources (CPU, database, memory) available within the enterprise and create a virtual computing resource. Customers can use transparently this pool of resources in a inexpensive and consistent way, regardless the location of the resources and, in case, fulfilling the need for computing capacity on-demand.

Oracle Portal includes advanced search capabilities allowing content to be easily grouped and searched from Oracle databases or other remote sources.
Security controls are based on the definition of users, groups and related roles and access rights and the support for communications using SSL connection. Oracle Portals provides a set of optimization tools for load balancing and failover and cache mechanisms to accelerate the delivery of static or dynamic web content.

Product's Features

Product Name:	Oracle Portal
Organization Name:	Oracle
Company Web Page:	http://www.oracle.com
Product Web Page:	http://www.oracle.com/appserver/portal_home.html
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Linux, Windows, Solaris, HP-UX, AIX, Compaq Tru64
Web Server:	Apache
Database:	Oracle, DB2, Sybase, SQL Server, others
Programming Language:	Java, others
Main Standards:	HTTP, XML, SOAP, LDAP

Xerox DocuShare

Xerox DocuShare is a enterprise content management system built on a Java platform as a modular solution allowing organizations to buy what they need and scale to include more sophisticated features as they requirements grow. It is also integrated with Xerox multifunction systems.

DocuShare provides an out-of-the-box solution that enables all levels people to create, manage, share and publish all content types. The life cycle of the content is controlled by workflow mechanism including time based events and routes for reviews or approvals. SSL encryption and digital signatures are supported and it is possible to define multiple levels of security and permissions to control the access to the content. The main features also include a version control and full-text and metadata indexing and search with the Verity K2 search engine.

DocuShare is built on an open standard platform and offers support for a wide variety of contents: it is tightly integrated with desktop applications, Xerox multifunction devices and allows an easy integration with other solutions.
DocuShare supplies the capability of end-to-end content capture, processing and distribution and renditioning of image and text formats for Web presentation.

The CMS allows the creation of a collaborative environment in which individuals and groups can interact and share information associated with a project or task. The included record management function allows users to classify and create records at any time in the document process.

DocuShare helps Web content publishing including a web-based interface for content management and publishing tools and a template-driven editing process.

DocuShare doesn't explicitly include any DRM control: however through ContentGuard, a Xerox product, the company offers a comprehensive solution for rights management of digital contents, based on the XrML language and including tools for publishing, electronic commerce, user authorization and usage tracking.

Product's Features

Product Name:	DocuShare
Organization Name:	Xerox
Company Web Page:	http://www.xerox.com
Product Web Page:	http://docushare.xerox.com
License:	Proprietary
CMS Type:	Enterprise CMS
Platform:	Windows, Solaris, Linux
Web Server:	IIS, Apache, Sun ONE, Sun Java, other
Database:	MSDE 2000 SP3, SQL , Oracle, PostgreSQL
Programming Language:	Java, other

3.1.4 Comparison table

The following table compares the main relevant features of the CMSs reviewed in the previous sections.

CMS Product	Workflow engine	Digital Asset management	Image resizing	Template	Metadata	Versioning	Granular privileges	DRM	XML	SSL	RSS	LDAP
Apache Lenya	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
Ariadne	yes	no	yes	yes	yes	no	yes	no	yes	yes	yes	yes
Communiqué	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
CoreMedia CMS 2005	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Documentum	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Drupal	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
ECMsuite	yes	yes	no	yes	yes	yes	yes	no	yes	yes	yes	yes
Ektron CMS400.NET	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
eZ Publish	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
FatWireCS	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
FileNet P8	yes	no	no	yes	yes	yes	yes	no	yes	yes	no	yes
IBM DB2	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Ingeniux CMS	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
Interwoven	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Livelink	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
Mambo	no	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes
Mediasurface	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
Microsoft CMS 2002	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
MySource Matrix	yes	yes	yes	yes	yes	yes	yes	no	no	yes	yes	yes
OpenCMS	yes	yes	no	yes	yes	yes	yes	no	yes	yes	yes	no
Oracle Portal	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
Percussion Rhythmyx	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Plone	yes	yes	yes	yes	yes	yes	yes	no	no	yes	yes	yes
QP7.Enterprise	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
RedDot XCMS	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	no	yes
Stellent	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes
Terminalfour	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
Tridion R5	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Typo3	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
Vignette V7	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Xerox DocuShare	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	no	yes

Table 1 – Comparison table for CMSs

3.1.5 CMS for Internet distribution in the AXMEDIS project

As it should be fairly clear from the analysis of the state of the art, a CMS for digital asset management should be more than a simple tool for tracking website contents and for helping non-technical people in publishing information to a site through templates. In the AXMEDIS demonstrator for Internet content delivery, an effective and efficient solution for contents management will be implemented starting from the analysis of the tools currently available on the market: the AXMEDIS system will be capable of handling, organizing and repurposing heterogeneous content, reducing distribution costs and increasing revenues for owners and distributors.

Standards support is also an essential feature of any CMS solution for media distributors: XML, for example, can be very important for creating, managing and exchanging digital assets in different formats and should be considered as a critical requirement in the process of identifying a common format from which the content can be repurposed across media.

Here is a list of the main features that will be implemented in the CMS:

- configurable engine for uploading and handling contents from multiple sources;
- support for multiple targets (multi channel delivery);
- support for content in many different formats (text, images, video, AXMEDIS objects, etc.);
- user profiling (editor, publisher, project manager, web designer, etc.);
- workflow engine;
- web accessible interface;
- multi platform (Unix/Linux, Windows, Mac, etc.);
- compliant to the AXMEDIS framework;

Content will be described as “core content” and will be mainly managed through metadata (in XML format) associated to it. The CMS will support creation of new content types and, for each of them, the creation of schemas and layouts for multiple target delivery. A configurable content import engine will support metadata uploading in XML as well as XML/XSL transformation, while a plug-in structure will manage content definitions that are not in XML format. The delivery target engine will allow delivering/publishing of content to more than one target at the same time, with XSL layout customization and support for different methods (ftp, http, SOAP, etc.). The workflow manager will allow creation of complex publishing structures where content could be published depending on conditions and/or results of other actions performed in the CMS. The CMS used in the demonstrator will be written in Java and will work according to the following schema:

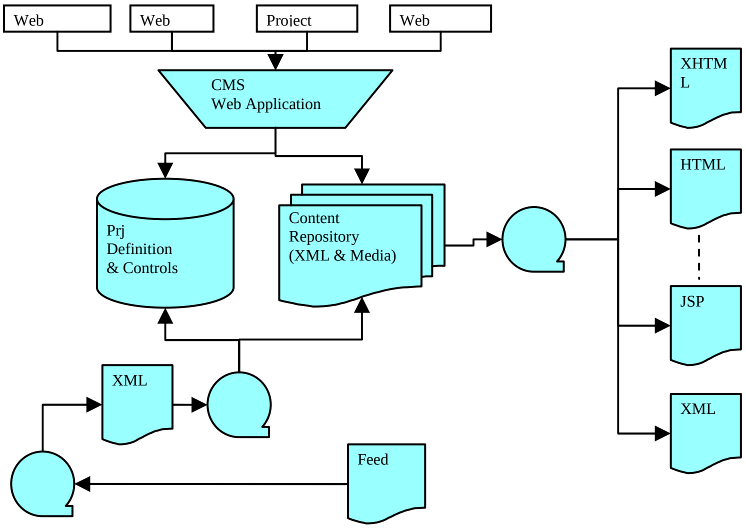


Figure 1 – Schema of the CMS used in the demonstrator

3.1.6 CMS Prototype description

These sections will describe status of the CMS prototype. This will be used to implement the demonstrator in WP9.

3.1.6.1 CMS Class Diagram

Diagram below shows the CMS Model Class Diagram implemented in the demonstrator:

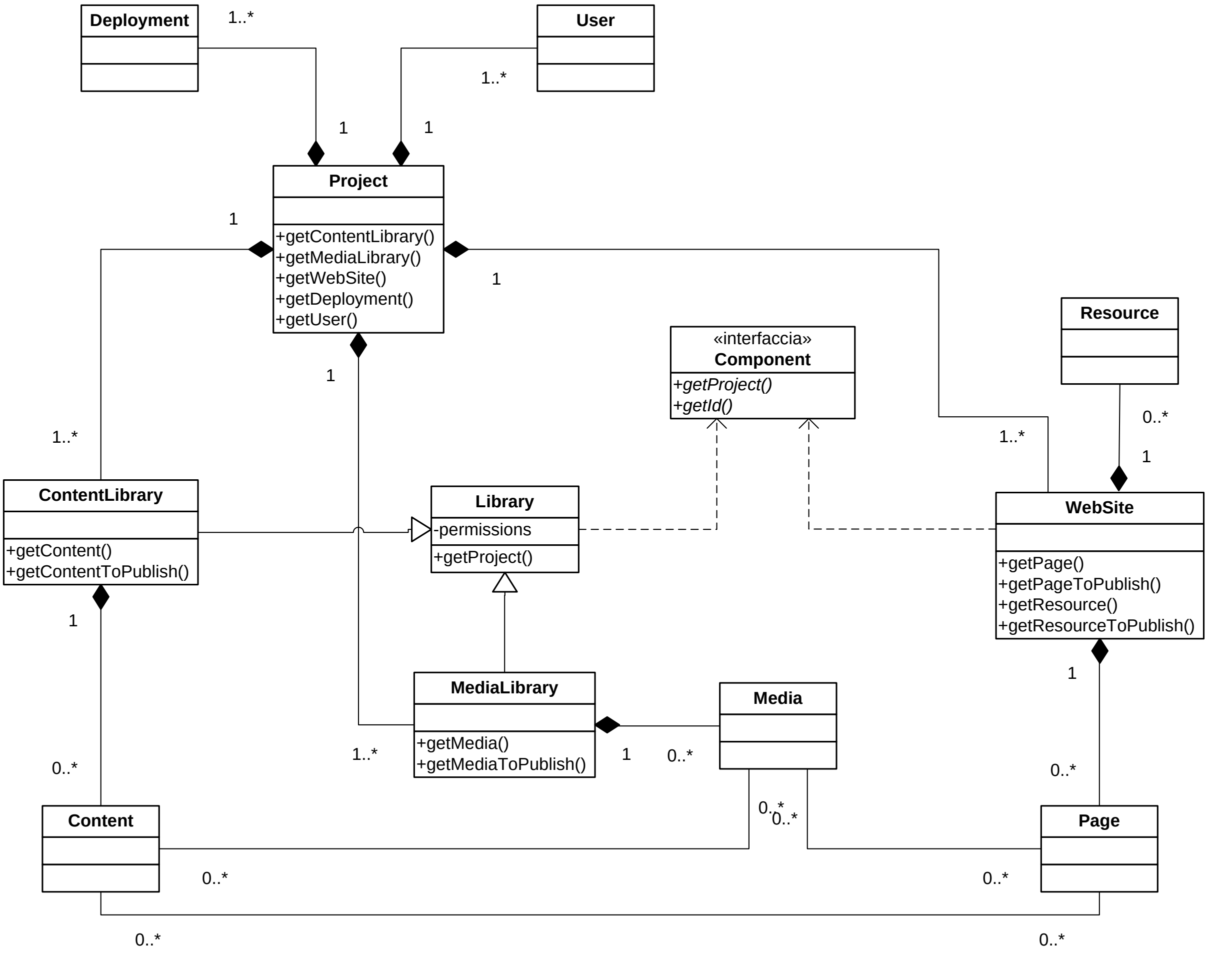


Figure 2 – CMS Model Class Diagram

3.1.6.2 Current functionalities

In the Xaura2 Class Model, the "Project" class is a container of Libraries, WebSites, Deployments and Users. It stores every fundamental element for the content publishing management and contains their structures.

The main components are: "ContentLibrary", "MediaLibrary" and "WebSite" class. All, directly and not, implement the "Component" interface and manage the publication of elements associated to them.

The "ContentLibrary" class manages all features related to contents as the creation of “ContentType" (the form structure of a content) and the creation of "TargetType" (the layout structure of a content).
The "MediaLibrary" class manages the creation and the categorization of media contents.
The "WebSite" class manages the structures of the pages (the form and layout structure) and the resources associated.

The model classes principally display the methods affecting their interactions, while the "actions" packages manage the actions of creation, editing, publishing and deleting of the "Content", "Media", "Page", "Resource" objects.
Among other packages, the "xml" packages manage the presentation of the system objects and the "threads" package manages the classes affecting the publishing engine.

3.1.6.3 Multidevice content delivery

The key point of Xaura2 CMS is the way how is organized and managed contents.
Xaura2 content structure is fully XML based and is maintained in the file system instead of DB, while control information such us publishing infos, publication layout related to the content etc are stored in the DB.
Adopting this structure of content model became very easy to manage contents with a very different xml structure in the same web site and/or to provide the same content in multi device presentation.

The following schema highlights the Xaura2 architectural components that guarantees the multidevice publishing.

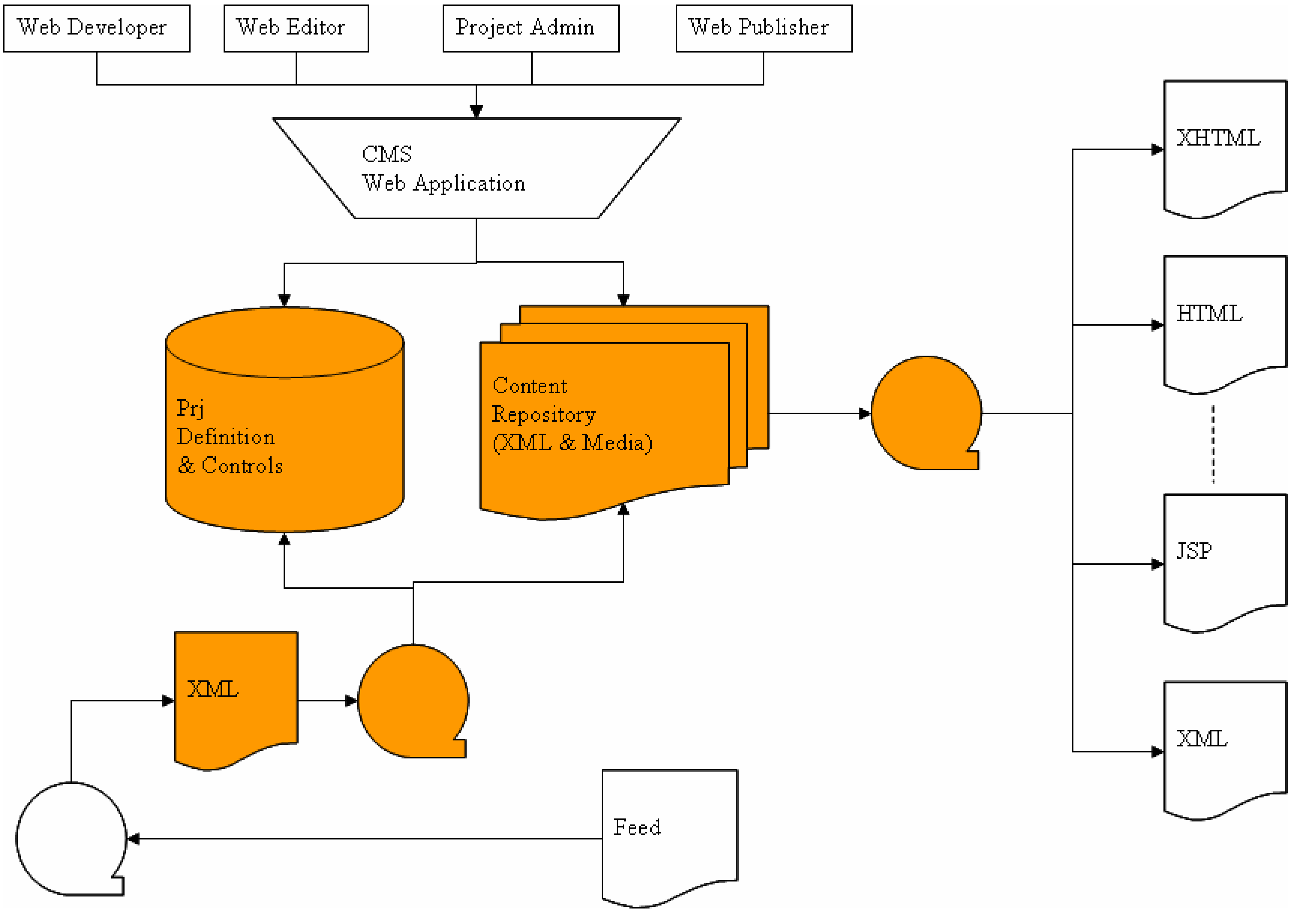


Figure 3 – Xaura2: architectural components

Contents, if needed, are first transformed in a xml structure, than are stored in the file system. In the same time a content reference is inserted in the db.
Now in the db are stored the content references, the project where the content will be present and, for each project the “web sites” (targets) that are contained in the project and, for each of them, the schema and the layout used to manage the content.
Each “target” may correspond to a specific device (content published in the right format for the target device) or a “target” or a feed file format that will be included in another system. The right output for each of them is obtained applying the right stylesheet to the xml content structure.

The following workflow diagram summarizes the steps taken during the content publishing process.

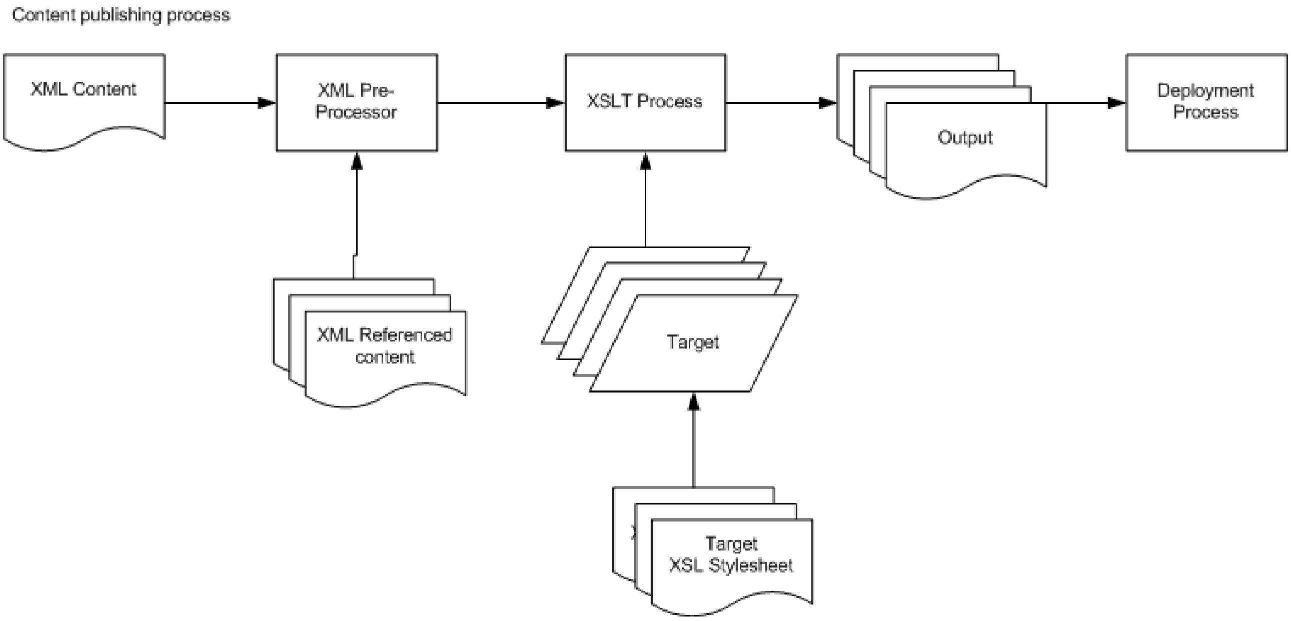


Figure 4 – Content publishing process

The following workflow diagram summarizes the steps taken during the page publishing process in a specified target.

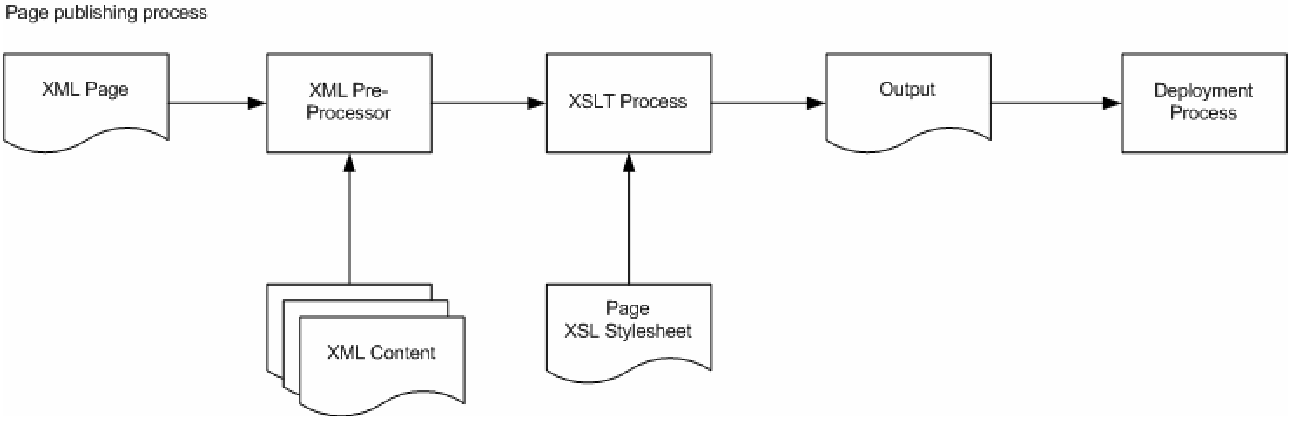


Figure 5 – Page publishing process

In this case more the one content will contribute to a “page” creation.

3.1.6.4 AXFW location of the demonstrator

Applications/mediaclub/
Applications/mediaclub/cms.jar
Applications/mediaclub/javadoc.zip

3.1.6.5 List of libraries used

None at this stage.

3.1.6.6 Prototype project responsible

Giovanni Nateri (TISCALI).

3.1.6.7 References to other components needed

AXEPTOOL, AXPMS, AXCS and CAMART will be used.

3.1.6.8 Configuration and execution context

None at this stage.

4 DRM integration and interoperability (FUPF, UPC from M25)

4.1 State of the art

This section describes the state of the art in DRM integration and interoperability.

4.1.1 Windows Media DRM

Windows Media digital rights management (DRM) [24] is a proven platform to protect and securely deliver content for playback on a computer, portable device, or network device. It's flexible to support a range of business models from single downloads or physical format delivery. The latest version of Windows Media DRM enables new scenarios and provides consumers even greater access to protected audio and video content.

Scenarios below present some business models and acquisition scenarios that Windows Media DRM can enable.

- Direct License Acquisition
- Indirect License Acquisition
- Subscription Services
- Purchase and Download Single Tracks
- Rental Services
- Video-on-Demand and Pay-Per-View

The Windows Media DRM platform enables the secure delivery of protected content for playback on a computer, portable device, or network device.

- Windows Media DRM 10 for Portable Devices
- Windows Media DRM 10 for Network Devices
- Windows Media Rights Manager 10 SDK
- Windows Media Format 9.5 SDK
- Windows Media Data Session Toolkit

Windows Media Rights Manager lets content providers deliver songs, videos, and other digital media content over the Internet in a protected, encrypted file format. Windows Media Rights Manager helps protect digital media (such as songs and videos) by packaging digital media files. A packaged media file contains a version of a media file that has been encrypted and locked with a "key." This packaged file is also bundled with additional information from the content provider. The result is a packaged media file that can only be played by a person who has obtained a license.

<u>Architecture</u>

The basic Windows Media Rights Manager process is as follows:

- Packaging: Windows Media Rights Manager packages the digital media file. The packaged media file has been encrypted and locked with a "key." This key is stored in an encrypted license, which is distributed separately. Other information is added to the media file, such as the URL where the license can be acquired. This packaged digital media file is saved in Windows Media Audio format (with a .wma file name extension) or Windows Media Video format (with a .wmv file name extension).
- Distribution: The packaged file can be placed on a Web site for download, placed on a media server for streaming, distributed on a CD, or e-mailed to consumers. Windows Media Rights Manager permits consumers to send copy-protected digital media files to their friends, as well.
- Establishing a License Server: The content provider chooses a license clearing house that stores the specific rights or rules of the license and implements the Windows Media Rights Manager license services. The role of the clearing house is to authenticate the consumer's request for a license. Digital

media files and licenses are distributed and stored separately, making it easier to manage the entire system.

- License Acquisition: To play a packaged digital media file, the consumer must first acquire a license key to unlock the file. The process of acquiring a license begins automatically when the consumer attempts to acquire the protected content, acquires a predelivered license, or plays the file for the first time. Windows Media Rights Manager either sends the consumer to a registration page where information is requested or payment is required, or "silently" retrieves a license from a clearing house.
- Playing the Media File: To play the digital media file, the consumer needs a media player that supports Windows Media Rights Manager. The consumer can then play the digital media file according to the rules or rights that are included in the license. Licenses can have different rights, such as start times and dates, duration, and counted operations. For instance, default rights may allow the consumer to play the digital media file on a specific computer and copy the file to a portable device. Licenses, however, are not transferable. If a consumer sends a packaged digital media file to a friend, this friend must acquire his or her own license to play the file. This PC-by-PC licensing scheme ensures that the packaged digital media file can only be played by the computer that has been granted the license key for that file.

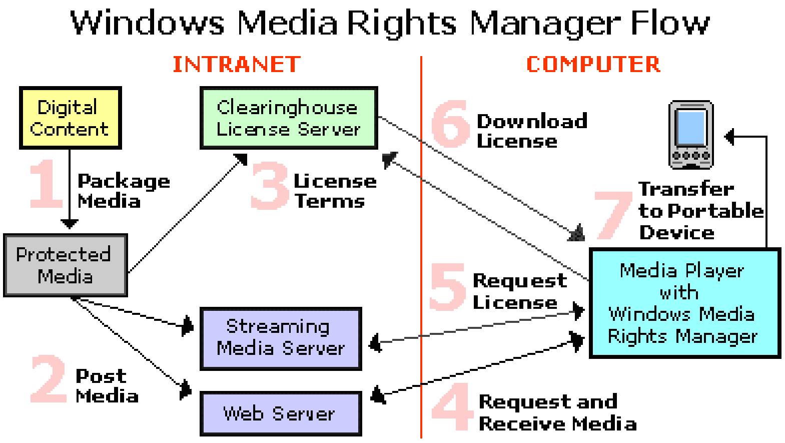

Windows Media Rights Manager Flow
INTRANET
COMPUTER
Digital Content
Clearinghouse License Server
1 Package Media
3 License Terms
Protected Media
2 Post Media
Streaming Media Server
Web Server
6 Download License
7 Transfer to Portable Device
5 Request License
Media Player with Windows Media Rights Manager
4 Request and Receive Media

Figure 6 – Windows Media Rights Manager Flow

Licenses and keys

The content owner locks their content with a "key" to create a packaged file. Before the consumer can play the file, the license clearing house creates a license containing the key that can unlock the packaged file and download the license to the consumers PC. The following diagram shows how keys are created and used in Windows Media Rights Manager.

To generate a key, a license key seed and a key ID are needed:

- The license key seed is a value that is known only to the content owner and license clearing house.
- The key ID is created by the content owner for each Windows Media file. This value is included in the packaged file.

When the license clearing house needs to issue a license for a packaged file, a key can be recreated by retrieving the key ID from the packaged file. The Windows Media License Service uses the license key seed (which the clearing house provides) and the key ID from the packaged file to create a key. The key is included in the license sent to the consumer's computer. Using the key included in the license, the player on the consumer's computer can open and play the protected file.

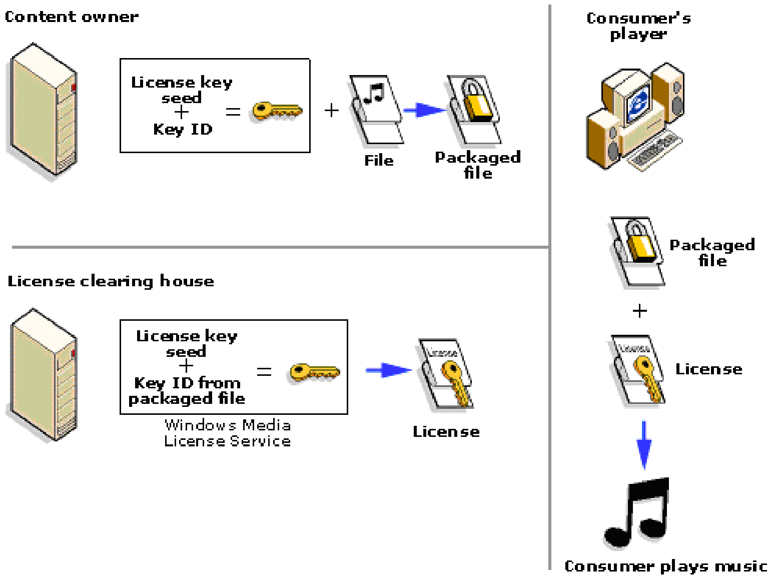


Figure 7 – Licenses and keys

Each license contains the key to unlock the Windows Media file. The license also contains the rights, or rules, that govern the use of the digital media file. The content owner sets these rights to determine which actions are allowed from minimal control over playback to more restrictive licenses. The licenses in Windows Media Rights Manager can support a wide range of different business rules, including:

- How many times can a file be played.
- Which devices a file can be played or transferred on. For example, rights can specify if consumers can transfer the file to portable devices that are compliant with the Secure Digital Music Initiative (SDMI).
- When the user can start playing the file and what is the expiration date.
- If the file can be transferred to a CD recorder (burner).
- If the user can back up and restore the license.
- What security level is required on the client to play the Windows Media file.
- And many others.

Licenses can be delivered in different ways and at different times, depending on the business model. The content owner might want licenses pre-delivered, or they might want the license delivered after a consumer has downloaded and attempted to play a packaged file for the first time. Licenses can be delivered with or without the consumer being aware of the process using silent or non-silent license delivery.

Windows Media DRM 10

Microsoft is introducing two new technologies that allow devices to use media files protected with Microsoft® Windows Media® digital rights management (DRM) technology.

- Microsoft Windows Media DRM 10 for Portable Devices: a lightweight DRM technology that enables portable devices, for example music players, digital video recorders and mobile phones, to use protected content. A portable device built on this technology can host a player that allows approved users to acquire and play protected content, subject to the rights in the license.
- Microsoft Windows Media DRM 10 for Network Devices: Network devices use Windows Media DRM 10 for Network Devices to render protected content from a computer running Microsoft Windows® XP or Windows XP Media Center Edition over a home network. These devices include set-top boxes for video playback (sometimes called Digital Media Receivers), or audio playback (sometimes called Digital Audio Receivers). A network device only requests protected content and renders it immediately. It cannot store or perform any other actions on content.

Both technologies require the device designer to sign a licensing agreement with Microsoft.
It is a porting kit that enables portable devices to play content that was protected using the Windows Media Rights Management Software Development Kit (SDK).

Windows Media DRM 10 for Network Devices

Windows Media DRM 10 for Network Devices is a porting kit that includes documentation, sample applications, and reference code to help programmers develop applications with Windows Media DRM 10 for Network Devices. These tools ensure that developers can write applications that a device can use to request and receive protected content from a personal computer on a home network. The Windows Media DRM 10 for Network Devices Porting Kit describes the requests and headers that must be exchanged in order for the device to request and receive content. The content can be encoded using any codec included with the Windows Media Format Software Development Kit (SDK).
By using this protocol, the network device acts as a simple rendering device in a home network. This network can be wired or wireless. Whether or not the device is within the home is determined by periodic proximity detection. The network computer storing the content must be running any version of Microsoft Windows® XP and host the Windows Media Connect software component (also called Media Connect). This component handles all media requests from the network device and can also handle license acquisition for content that needs a new or updated license. If licenses must be acquired through the Internet, the computer must have an Internet connection and a browser. A network device can only request files currently on the network computer; it cannot receive streaming Internet content.
Examples of network devices include Digital Media Receivers that stream video to a television and Digital Audio Receivers that play music on a home stereo.

Here is how a network device works:

- The first time a device is used, it must be registered and authorized by the server through Universal Plug and Play. Registration involves a device ID number supplied during manufacturing and a signed XML device certificate. This certificate identifies the device and contains information used to ensure secure communication.
- During initial registration, the server pings the device to verify that it is close enough to be considered inside the home. This is determined by measuring whether the device responds to server proximity detection within 7 milliseconds. This proximity detection also validates the device.
- Periodically, the server repeats proximity detection to revalidate the device.
- The device requests content for playback from the server.
- If the server determines that the device is validated and has the right to play the content, it sends a response containing a new, encrypted session key, a rights policy statement specifying the security restrictions that the device must enforce, and finally the content. The content is encrypted by the session key. Each time content is requested, a new session is established.
- The network device must parse the rights policy and determine if it can adhere to the required rights. If it can, it may render the content.

The following diagram illustrates devices around the home streaming content over a home network.

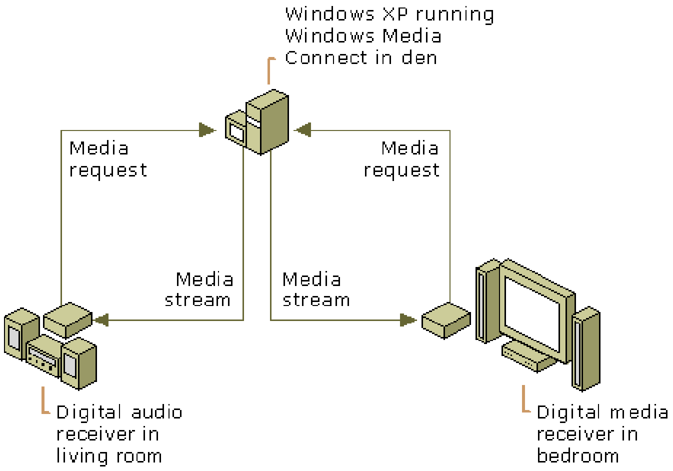


Figure 8 – Diagram showing network devices streaming protected content

4.1.2 MPEG-21

MPEG's approach is to define a Multimedia Framework to ensure that the systems that deliver multimedia content are interoperable and that the transactions between them are simplified and automated. This approach should apply to the infrastructure requirements for content delivery, content security, rights management, secure payment, and the technologies enabling them. The result is an open framework for multimedia delivery and consumption for use by all the players in the delivery chain. This open framework will provide content creators, producers, distributors and service providers with equal opportunities in the MPEG-21 [1] enabled open market. This will also be to the benefit of the content consumer providing them access to a large variety of content in an interoperable manner.
The MPEG-21 multimedia framework has two essential concepts the Digital Item, a fundamental unit of distribution and transaction and the Users that interact with Digital Items.

MPEG-21 aims to identify and define the different mechanisms and elements needed to support the multimedia delivery chain, the relationships and the operations supported by them. In the different parts of the MPEG-21 standard, these elements are elaborated by defining the syntax and semantics of their characteristics, such as interfaces to these elements. The MPEG-21 standard is currently formed by seventeen parts.

Part 1: Vision, Technologies and Strategy [2]. The purpose of this part of the standard is to define a vision for a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices to meet the needs of all users. This part has as objective to achieve the integration of standards to facilitate harmonisation of technologies for the creation, management, distribution and consumption of digital items. Moreover it shall define a strategy for achieving a multimedia framework based on well-defined functional requirements.

Part 2: Digital Item Declaration (DID) [3]. The second part of the MPEG-21 standard describes a set of abstract terms and concepts to form a useful model for defining Digital Items. A Digital Item is the representation of a work, and as such, it is the thing that is acted upon (managed, described, exchanged, etc.) within the model.

Part 3: Digital Item Identification (DII) [4]. This part of the standard provides a schema that can be used to include identifiers into a Digital Item Declaration. Then, Digital Items and parts thereof (such as resources) could be uniquely identified.

Part 4: Intellectual Property Management and Protection (IPMP) [5].This part of MPEG-21 will define an interoperable framework for Intellectual Property Management and Protection. It includes standardized ways of retrieving IPMP tools from remote locations, exchanging messages between IPMP tools and between these tools and the terminal. It also addresses authentication of IPMP tools, and integration of rights expressions according to the Rights Data Dictionary and the Rights Expression Language.

Part 5: Rights Expression Language (REL) [6]. A Rights Expression Language is seen as a machine-readable language that can declare rights and permissions using the terms as defined in the Rights Data Dictionary. The REL is intended to provide flexible, interoperable mechanisms to support transparent and augmented use of digital resources in a way that protects digital content and honours the rights, conditions, and fees specified for digital contents. It is also intended to support specification of access and use controls for digital content in cases where financial exchange is not part of the terms of use, and to support exchange of sensitive or private digital content.

Part 6: Rights Data Dictionary (RDD) [7]. The Rights Data Dictionary comprises a set of clear, consistent, structured, integrated and uniquely identified Terms to support the MPEG-21 Rights Expression Language. This part of the standard also specifies the methodology and structure of the RDD Dictionary and specifies how further terms may be defined under the governance of a registration Authority. The RDD System is made up of the RDD Dictionary and RDD Database taken together. It will facilitate the exchange of information between different parties involved in the administration of rights.

Part 7: Digital Item Adaptation (DIA) [8]. One of the goals of MPEG-21 is to achieve interoperable transparent access to distributed advanced multimedia content by shielding users from network and terminal installation, management and implementation issues. This will primarily enable the provision of network and terminal resources on demand so that multimedia content can be created and ubiquitously shared, always with the agreed/contracted quality, reliability and flexibility. Towards this goal, the adaptation of Digital Items is required. Digital Items are subject to a resource adaptation engine, as well as a descriptor adaptation engine, which together produce the adapted Digital Items.
The target for this part of the standard is to specify tools that provide input to the adaptation engine, so that any constraints on the delivery and consumption of resources can be satisfied, and the quality of the user experience can be guaranteed.

Part 8: Reference Software [9]. In the eighth part of the standard is presented the normative and informative reference software developed in other parts of the MPEG-21 standard, such as REL, RDD, DID, DIA, DIP…, and software modules that integrates the functionalities of these parts. Reference software will form the first of what is envisaged to be a number of systems-related specifications in MPEG-21. The development of the Reference Software will be based on the requirements that have been defined in the different parts of MPEG-21.

Part 9: File Format [10]. In this part of the MPEG-21 standard a file format shall be defined. An MPEG-21 Digital Item can be a complex collection of information. Both still and dynamic media can be included, as well as Digital Item information, metadata, layout information, and so on. It can include both textual data and binary data. For this reason, the MPEG-21 file format inherits several concepts from MP4, in order to make 'multi-purpose' files possible. A dual-purpose MP4 and MP21 file, for example, would play just the MPEG-4 data on an MP4 player, and would play the MPEG-21 data on an MP21 player. A 'resource map' allows the inclusion of multiple referenced resources in the same or other files, and for systems-level management of those resources.

Part 10: Digital Item Processing (DIP) [11]. The objective of this part of the standard is to provide a normative set of tools for specifying processing of a Digital Item in a predefined manner. In this way, it will be possible to extend Digital Item Declaration Language in order to add user specific functionality inside the Digital Item. Therefore, the standardisation of Digital Item Processing will allow interoperability at the processing level. The main idea behind the Digital Item Processing Architecture is that, on receipt of a DID, a list of DI Methods that can be applied to the Digital Item is presented to the User. After that the User chooses one Method that is then executed by the DIP Engine.
A Digital Item Method, DIM, is the tool whereby a User specifies some desired functionality. It is expressed using the Digital Item Method Language, DIML, which includes a binding for Digital Item Base Operations. The Digital Item Base Operations, DIBOs are the functional building blocks utilised by a Digital Item Method. They can be considered somewhat analogous to the standard library of functions of a programming

language. Digital Item Methods are defined by the Digital Item Base Operations they use to accomplish the handling of the Digital Item according to the intentions of the Digital Item Method author.

Part 11: Evaluation Methods for Persistent Association Technologies [12]. This part consists of the comparison of technical report documents that evaluate persistent association technologies, for example, technologies that link information to identify and describe content using the content itself.
This part of the MPEG-21 standard does not contain any normative behaviour, its purpose is to allow evaluations of such technologies to be conducted using a common methodology rather than to standardise the technologies themselves.

Part 13: It was removed from MPEG-21 standard as its content was not suitable for MPEG-21 objectives.

Part 12: Test Bed for MPEG-21 Resource Delivery [13]. This part of the MPEG-21 standard provides a software-based test bed for the delivery of scalable media delivery, and testing/evaluating this scalable media delivery in streaming environments, for example by taking into account varying network environments.

Part 14: Conformance Testing [14]. The purpose of this part is to define conformance testing for other parts of MPEG-21.

Part 15: Event Reporting (ER) [15]. The purpose of this part of the MPEG-21 standard is to provide a standardised way for sharing information about events, referred to digital items and peers that interact with them, within the MPEG-21 multimedia framework. Use cases that help to understand the necessity of event reporting are the monitoring of usage of copyrighted material and the necessity for network nodes to know the connectivity condition between peers within a network when trying to deliver multimedia content. This part of the standard specifies how to express Event Report Requests (ER R) that contain information about which Events to report, what information is to be reported and to whom; and Event Reports (ER) which are created by an MPEG-21 Peer in response to an Event Report Request when the conditions specified by an ER-R are met.

Part 16: Binary Format [16]. This part of the standard describes the methods to binarise MPEG-21 documents.

Part 17: Fragment Identification of MPEG Resources [17] specifies a normative syntax for URI Fragment Identifiers.

Part 18: Digital Item Streaming, specifies the way of streaming digital items.

Digital Item Declaration (DID)

The two major goals of the Digital Item Declaration part [3] within MPEG-21 are first to establish a flexible and interoperable schema for declaring Digital Items and second to be as general and flexible as possible, providing hooks to enable higher level functionality and interoperability.
A Digital Item is defined in [3] as a structured digital object, including a standard representation, identification and metadata. It is the fundamental unit of distribution and transaction inside MPEG-21.
The Digital Item Declaration technology is defined in three normative parts: DID Model, Representation and Schema. Digital Item Declaration Model consists on the description of a set of abstract terms and concepts to form a useful model for defining Digital Items. Within this model, a Digital Item is the digital representation of a work, and as such, it is the thing that is acted upon within the model. The aim of this model is to be as general and flexible as possible, providing hooks to enable higher level functionality and interoperability. The model provides a common set of abstract concepts and terms that can be used to define a scheme, or to perform mappings between existing schemes capable of Digital Item Declaration, for comparison purposes.
The terms that form the model are described below:

- Container: The structure that allows items and/or containers to be grouped. Descriptors are useful in order to label the containers with the appropriate information.
- Item: A grouping of sub-items and/or components that are bound to relevant descriptors. Descriptor can contain information about the item.

- Component: The binding of a resource to a set of descriptors, that contain information about all or part of the specific resource instance. These descriptors contain control or structural information about the resource, such as bit rate, encryption information….
- Anchor: binds descriptors to a fragment. A fragment corresponds to a specific location or a part of a resource.
- Descriptor: associates information with the enclosing element. This information may be a component or a textual statement.
- Condition: describes the enclosing element as being optional. Multiple conditions associated with an element are combined as a disjunction when determining if include or not the element.
- Choice: describes a set of related selections that can affect the configuration of an item.
- Selection: describes a specific decision that will affect one or more conditions within an item.
- Annotation: describes the information about another element of the model.
- Assertion: defines a configured state of a choice by asserting true, false or undecided values for the predicates associated with the selections for that choice.
- Resource: an individually identifiable asset such as an image, audio, video clip or a textual asset. It may also potentially be an physical object. The resources must be locatable via an unambiguous address.
- Fragment: designates a specific point or range within a resource.
- Statement: a literal textual value that contains information. It can include descriptive, control or identifying information.
- Predicate: an unambiguous identifiable declaration that can be true, false or undecided.

Next figure shows the relationship among some of the terms of the model defined. The digital item represented has a container, which inside groups some items together with their descriptors and components.

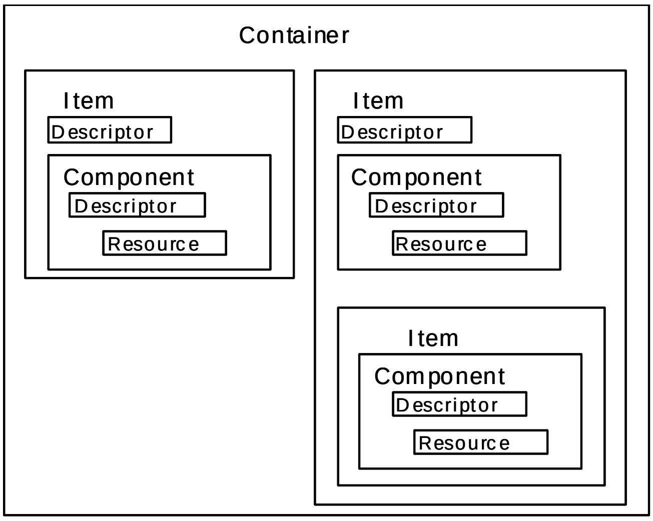


Figure 9 – Example Digital Item Declaration

Apart from the declaration model, in MPEG-21 standard part 2 it is also specified the Representation, which describes the syntax and semantics of the Digital Item Declaration elements. The abstract elements defined in the Model that have been represented in DIDL are: Container, Item, Component, Anchor, Descriptor, Choice, Selection, Condition, Annotation, Assertion, Resource and Statement. Two special element types that do not correspond to any of the model elements, Reference and Declarations, have been also defined. The Reference element is used to link the contents of an element inside another element. The Declarations element is used to define a set of Digital Item Description Language elements in a document without actually instantiating them.

A DIDL document consists of a DIDL root element with an Item child element or a Container child element. An Item is a grouping of possible sub-Items and/or Components, bound to a set of relevant Descriptors containing descriptive information about the item. A Component groups a Resource element with a set of Descriptors containing descriptive information about the resource, plus a set of Anchors specifying points or regions of interest in the resource. The Component, being a logical union of a resource with relevant descriptive data and anchors, is intended to be the basic building block of digital content within a DIDL document. A Descriptor associates information with its parent element; this information may be contained in a Component element or in a Statement element. A Statement defines a piece of information pertaining to the parent element. It can include descriptive, control, revision tracking or identifying information. It can contain any data format, including plain text and various machine-readable formats such as well-formed XML. A Statement can also be defined by reference, by specifying the URI of the Statement. A Resource is an individually identifiable asset such as a video or audio clip, an image, an electronic ticket or a textual work.

The ability of inserting data in any kind of data format, specially well-formed XML, inside a Statement provides a wide field for inserting information for the protection and processing of multimedia data. For example, if we want to associate rights expressions to a particular resource within a Digital Item, the REL License can be placed in the Statement of the Descriptor element related to the resource.

Rights Expression Language (REL)

The different parties involved in the online distribution and consumption of multimedia resources need to exchange information about the rights, terms, and conditions associated with each resource at each step in the multimedia resource lifecycle. For example in distribution and super distribution business models, the information related to the rights and the terms and conditions under which the rights may be exercised needs to be communicated to each participant in the distribution chain.

In an end-to-end system, other considerations such as authenticity and integrity of Rights Expressions become important. For example, any content provider or distributor who issues rights to use or distribute resources must be identified and authorized. In addition, a Rights Expression may be accessed by different participants which requires mechanisms and semantics for validating the authenticity and integrity of the Rights Expression. A common Rights Expression Language that can be shared among all participants in this digital workflow is required.

Part 5 of the MPEG-21 standard specifies the syntax and semantics of a Rights Expression Language. MPEG chose XrML as the basis for the development of the MPEG-21 Rights expression language.

MPEG-21 Rights Expression Language (REL) [6] specifies the syntax and semantics of the language for issuing rights for Users to act on Digital Items, their Components, Fragments, and Containers.

The most important concept in REL is the license that conceptually is a container of grants, each one of which conveys to a principal the sanction to exercise a right against a resource. A license if formed by the following elements:

- Title: this element provides a descriptive phrase about the License that is intended for human consumption in user interfaces. Automated processors must not interpret semantically the contents of such title elements.
- Inventory: this element is used for defining variables within a License. In the Inventory element of a license can be defined LicensePart elements that in turn can have licensePartId attributes that can be referenced from elsewhere in the license.
  Therefore, REL provides a syntactic mechanism for reducing redundancy and verbosity in Licenses that can be used throughout a License.
- Grant or GrantGroup: The Grants and GrantGroups contained in a license are the means by which authorization policies are conveyed in the REL architecture. Each Grant or GrantGroup that is an immediate child of a license exists independently within that license, no collective semantic (having

to do with their particular ordering or otherwise) is intrinsically associated with the presence of two or more of them within a certain license.

- Other information: Using the wildcard construct from XML Schema, a License provides an extensibility hook within which license issuers may place additional content as they find appropriate and convenient. This can be useful for conveying information that is peripherally related to, for example, authentication and authorization, but is not part of the REL core infrastructure.
  It should, however, be carefully understood that not all processors of REL licenses will understand the semantics intended by any particular use of this extensibility hook. Processors of the license may choose wholly at their own discretion to completely ignore any such content that might be present therein.

Next figure shows the structure of a REL License.

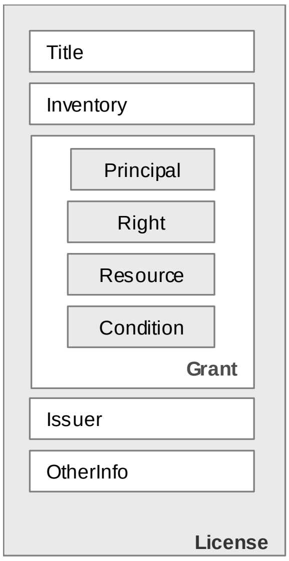


Figure 10 – REL License Structure

The most important concept within a license is the grant that conveys to a particular principal the sanction to exercise some identified right against some identified resource, possibly subject to the need for some condition to be first fulfilled. A Grant is an XML structure that is at the heart of the rights management and authorization policy semantics that REL is designed to express.

A grant is formed by four elements, a Principal that represents the unique identification of an entity involved in the granting or exercising of Rights. A Right that specifies an action or activity that a Principal may perform on, or using, some associated target Resource. A Resource that represents the object against which the Principal of a Grant has the Right to perform. The use of a digital resource in a Grant provides a means by which a sequence of digital bits can be identified within the Grant. The Condition element represents grammatical terms, conditions and obligations that a Principal must satisfy before it may take advantage of an authorization conveyed to it in a Grant. The issuer element that may contain two pieces of information, a set of issuer-specific details about the circumstances under which he issues the license, and an identification of the issuer, possibly coupled with a digital signature for the license. The optional issuer-specific details are found in the details element of the issuer. These details optionally include any of the following information the specific date and time at which this issuer claims to have effected his issuance of the license and an indication of the mechanism or mechanisms by which the Issuer of the license will, if he later Revokes it, post notice of such revocation. When checking for revocation, REL processing systems may choose to use any one of the identified mechanisms, that is, they are all considered equally authoritative as to the revocation status of the issuance of the License.

The structure of a REL license is the one described if it is in clear text, but a REL license can contain only an encryptedLicense element if the license is encrypted. The encryptedLicense element provides a mechanism by which the contents of a License may be encrypted and so hidden from view from inappropriate parties. This mechanism makes straightforward use of XML Encryption Syntax and Processing (XML Encryption). Specifically, the XML content model of a License is a choice between a sequence containing the elements previously described in this section and an encryptedLicense element that represents the encryption of the contents of the License element.

The principals, rights, resources and conditions of the REL are organized in three main groups. The first one, the Core specifies structural elements and types and how are they related. The standard extension and the multimedia extension specifies standard or multimedia principals, rights, resources and conditions. Each one of the parts is related to a namespace. Next table gives the prefix and the corresponding namespace.

Part	**Namespace prefix**	**Namespace**
Core	r	urn:mpeg:mpeg21:2003:01-REL-R-NS
Standard	sx	urn:mpeg:mpeg21:2003:01-REL-SX-NS
Multimedia	Mx	urn:mpeg:mpeg21:2003:01-REL-MX-NS

Table 2 – Namespace prefixes

At the heart of REL is the REL Core Schema whose elements and types define the core structural and validation semantics that comprises the essence of the specification. The REL Core Schema includes different elements and types organised in four main groups:

- Principals: Within REL, instances of the type Principal represent the unique identification of an entity involved in the granting or exercising of rights. They represent the subject that is permitted to carry out the action involved in exercising the Right. The principal element and its type are conceptually abstracts. Then, it does not indicate how a particular principal is actually identified and authenticated. Rather, this is carried out in types that are derivations of Principal. Such derived types may be defined in extensions to REL in order to provide, for example, a means by which Principals who are authenticated using some proprietary logon mechanism may be granted certain Rights using the REL License mechanism.
  There are derivations that are important and central enough to be defined within the REL core itself:
  - allPrincipals: Structurally, an AllPrincipals Principal is a simple container of Principals. Semantically, an AllPrincipals a represents the logical conjunct of the Principals represented by all of its children.
  - keyHolder: Instances of a KeyHolder Principal represent entities which are identified by their possession of a certain cryptographic key. For example, using a KeyHolder, a Principal that uses public-key cryptography may be conceptually identified as that Principal which possesses the private key that corresponds to this-here public key.
- Rights: Within REL, instances of the type Right represent a verb that a Principal may be authorized to carry out under the authority conveyed by some authorized Grant. Typically, a Right specifies an action or activity that a Principal may perform on or using some associated target Resource. The semantic specification of each different particular kind of Right should indicate which kinds of Resource if any may be legally used in authorized Grants containing that Right.
  The element right and its type are conceptually abstract. Therefore, the type Right itself does not indicate any actual action or activity that may be carried out. Rather, such actions or activities are to be defined in types that are derivations of Right. Such derived types will commonly be defined in extensions to REL. However, the following rights are related to the domain of the REL core itself:

  - issue: When an Issue element is used as the right in an authorized grant, it is required that resource against which the right is applied in fact be a grant or grantGroup. The grant then conveys the authorization for the principal to issue the resource.
    At the instant a License is issued, the issue right must be held by the issuer of the License with respect to all the grants and grantGroups directly authorized therein.
  - obtain: When an obtain element is used as the right in an authorized grant, the resource must be present and be a grant or a grantGroup. The use of the obtain right can be conceptualized as an offer or advertisement for the sale of the contained grant
  - possessProperty: The possessProperty right represents the right for the associated principal to claim ownership of a particular characteristic, which is listed as the resource associated with this Right.
  - revoke: The authorized act of exercising the revoke right by a principal effects a retraction of a dsig:Signature that was previously issued and thus accomplishes a withdrawal of any authorization conveyed by that dsig:Signature.
- Resources: An instance of type resource represents the direct object against which the subject principal of a grant has the right to perform some verb. The actual element resource and its type are conceptually abstracts. That is, the type resource itself does not indicate any actual object against which a Right may be carried out. Rather, such target objects are to be defined in types that are derivations of Resource. Such derived types will commonly be defined in extensions to REL. The relevant resources defined within the REL core:
  - digitalResource: Use of a digitalResource resource in a grant provides a means by which an arbitrary sequence of digital bits can be identified as being the target object of relevance within the grant. Specifically, such bits are not required to be character strings that conform to the XML specification, but may be arbitrary binary data. The means by which this is accomplished breaks down into several cases. For example, the bits are to be physically present within the digitalResource or the bits are to be physically located at some external location (e.g. in a Web site).
  - propertyAbstract: An instance of type propertyAbstract represents some sort of property that can be possessed by principals via possessProperty right.
- Conditions: Within REL, instances of the type Condition represent grammatical terms and conditions that a Principal must satisfy before it may take advantage of an authorization conveyed to it in a grant containing the condition instance. The semantic specification of each different particular kind of condition must indicate the details of the terms, conditions, and obligations that use of the Condition actually imposes. When these requirements are fulfilled, the Condition is said to be satisfied.
  The actual element condition and its type are conceptually abstracts. That is, the type Condition itself does not indicate the imposition of any actual term or condition. Rather, such terms and conditions are to be defined in types that are derivations of Condition. Such derived types will commonly be defined in extensions to REL. The conditions defined within the REL core that we consider relevant to detail:
  - AllConditions: Structurally, an allConditions is a simple container of conditions. Semantically, the allConditions represents a logical conjunct of the conditions represented by all of its children.
  - validityInterval: A ValidityInterval condition indicates a contiguous, unbroken interval of time. The semantics of the condition expressed is that the interval of the exercise of a right to which a validityInterval is applied must lie wholly within this interval. The delineation of the interval is expressed by the presence, as children of the condition, of up to two specific fixed time instants:
    - notBefore element, of type xsd:dateTime, indicates the inclusive instant in time at which the interval begins. If absent, the interval is considered to begin at an instant infinitely distant in the past

- notAfter element, also of type xsd:dateTime, indicates the inclusive instant in time at which the interval ends. If absent, the interval is considered to end at an instant infinitely distant in the future.

The Standard Extension schema defines terms to extend the usability of the Core Schema, some of them are:

- Right Extensions: Right Uri.
- Resource Extensions: Property Extensions and Revocable.
- Condition Extensions: Stateful Condition, State Reference Value Pattern, Exercise Limit Condition, Transfer Control Condition, Seek Approval Condition, Track Report Condition, Track Query Condition, Validity Interval Floating Condition, Validity Time Metered Condition, Validity Time Periodic Condition, Fee Condition and Territory Condition.
- Payment Abstract and its Extensions: Payment Abstract, Rate, Payment Flat, Payment Metered, Payment per Interval, Payment per Use, Best Price Under, Call for Price and Markup.
- Service Description: WSDL and UDDI
- Country, Region and Currency Qualified Names: Namespace URI Structure, Country Qualified Names, Region Qualified Names and Currency Qualified Names.
- Matches XPath Function: Regular Expression Syntax and Flags.

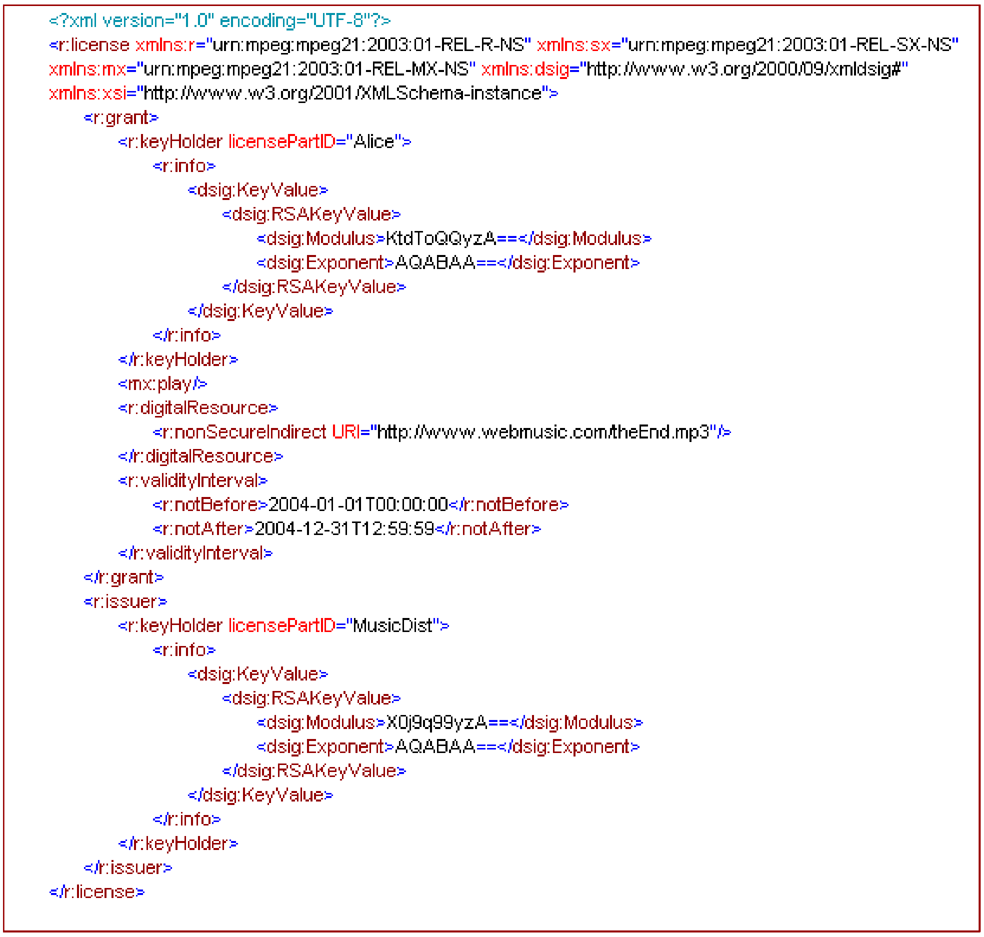
The REL Multimedia Extension expands the Core Schema by specifying terms that relate to digital works. Specifically describes rights, conditions and metadata for digital works, that includes:

- Rights: Modify, Enlarge, Reduce, Move, Adapt, Extract, Embed, Play, Print, Execute, Install, Uninstall and Delete.
- Resources: Digital Item Resources.
- Conditions: Resource Attribute Conditions, Digital Item Conditions, Marking Conditions, Security Conditions and Transactional Conditions.
- Resource Attribute Set Definitions: Complement, Intersection, Set and Union.

A typical example of a REL license issued to an end-user. In this case a distributor, MusicDist, issues to a user, Alice, a license that permits her the right of play a song, TheEnd.mp3, during this year. The license is sketched in the following figure.

The elements of the license are the grant and the issuer. The grant element is formed by four elements. The keyHolder that represents the user, Alice, which is identified by her possession of a certain cryptographic key. Alice is identified as the Principal that possess the private key that corresponds to this-here public key. The play element that represents the right. The definition of Play in the Rights Data Dictionary is to derive a transient and directly perceivable representation of a resource. The digitalResource element that provides a means by which an arbitrary sequence of digital bits can be identified as being the target object of relevance within the Grant. Conceptually, an instance of DigitalResource defines an algorithm by which a sequence of bits is to be located. If the bits are to be physically located at some external location, for example in this example they are located on a Web site, we use that nonSecureIndirect element child where we indicate the algorithm used to allocate the bits. In this example we indicate that the song is in the URI http://www.webmusic.com/TheEnd.mp3. And the fourth one, the ValidityInterval element that represents the condition. It indicates a contiguous, unbroken interval of time. The semantics of this Condition is that the interval of the exercise of a Right to which a ValidityInterval is applied must lie wholly within this interval. The delineation of the interval is expressed by the presence, as children of the Condition, of up to two specific fixed time instants, notBefore of type xsd:dateTime, indicates the inclusive instant in time at which the interval begins, 1 January 2004. And the notAfter element of type xsd:dateTime, indicates the inclusive instant in time at which the interval ends, 31 December 2004. Therefore, with this license the user can play the song during this year.

The issuer element indicates the entity that issues the license. In this example, it represents the music distributor that has the right to issue this kind of licenses to end-users.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <r:grant>
        <r:keyHolder licensePartID="Alice">
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
        <mx:play/>
        <r:digitalResource>
            <r:nonSecureIndirect URI="http://www.webmusic.com/theEnd.mp3"/>
        </r:digitalResource>
        <r:validityInterval>
            <r:notBefore>2004-01-01T00:00:00</r:notBefore>
            <r:notAfter>2004-12-31T12:59:59</r:notAfter>
        </r:validityInterval>
    </r:grant>
    <r:issuer>
        <r:keyHolder licensePartID="MusicDist">
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
    </r:issuer>
</r:license>
```

Figure 11 – REL License Example

Other important concept of the REL is the authorization model. It is used by any implementation of software which makes an authorization decision using REL licenses. The central question that lies in this decision making process *"is a principal authorized to exercise a right against a resource?"*

The REL Authorization Model makes use of an authorization request, an authorization context, an authorization story, and an authorizer.

An authorization request can be conceptualized as representing the question if is it permitted for a given Principal to perform a given Right upon a given Resource during a given time interval based on a given authorization context, a given set of Licenses, and a given trust root.

The authorization request contains the following members:

- the principal element, which is the identity of the entity for which permission is requested
- the right element, which embodies the semantics of the action which is requested to be permitted
- the resource element identifying the Resource upon which permission is requested
- the interval of time during which the requested performance of the right by the principal upon the resource is considered to take place. This may be either an instantaneous point in time or an unbroken interval of time
- the authorization context containing properties representing statements that are to be considered true for the purposes of establishing the requested permission

- the set of license elements that may be consulted to establish the requested permission. The algorithm will attempt to find authorized grants or grantGroups within this licenses that it can use to establish a basis for an affirmative authorization decision
- the set of grant elements that do not require an authorizer for the purposes of establishing the requested permission

Next figure shows the structure of an Authorization Request.

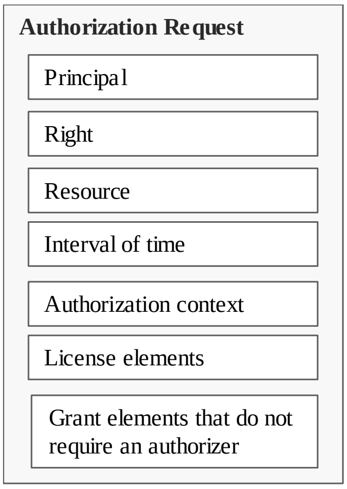


Figure 12 – REL Authorization Request

The authorization story, the following figure, contains the following elements:

- a primitive grant, it is used to demonstrate to which authorization requests the authorization story applies
- either a grant or a grantGroup, it represents the actual grant or grant group that is authorized by the authorizer of the authorization story
- an authorizer, it contains the following members:
    - the license in which the principal is authorized
    - the principal that authorized the license above
    - the time instant in which the license was issued
    - the authorization context that contains the properties representing statements that were considered true for the purposes of establishing the permission
    - an authorization story

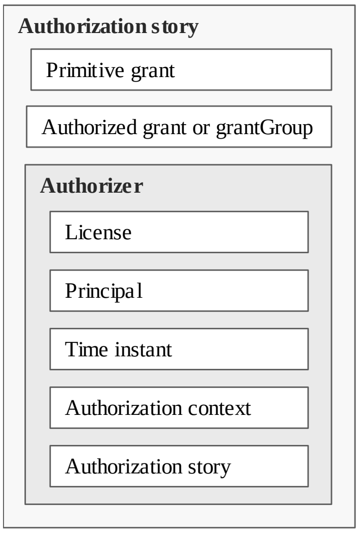


Figure 13 – REL Authorization Story

Rights Data Dictionary (RDD)

The Rights Data Dictionary [7] comprises a set of clear consistent, structured, integrated and uniquely identified Terms to support the MPEG-21 Rights Expression language (REL).

The RDD Dictionary has the characteristics of a structured ontology, in which meaning, once it has been defined, can be passed on from one term to another by logical rules of association such as inheritance and opposition. An ontology, in this context, is a structured catalogue of entities in which meaning, once defined, can be passed on from one term to another by logical rules of association such as inheritance and opposition. The structure of this ontology is designed to provide a set of well-defined terms for use in rights expressions governing the use of Digital Items. In recognition of the great diversity and complexity associated with multimedia content, it is also designed to represent as many different specializations of meaning as its users require, and to show their relationships in a structured way in order to support the mapping and transformation of terms between different schemas and systems.

The methodology described has been used to create the Standardized Terms for the RDD Dictionary, and may be used in future so that new terms can be introduced under the governance of a Registration Authority. The Standardized Terms in the RDD Dictionary are therefore not a closed list, but the foundations of a widely extensible Rights Data Dictionary.

The RDD System is comprised of the following three elements, the Specifications contained in the RDD Standard. A Dictionary, the Terms and their TermAttributes defined according to this specification. And a Database, the tool containing the RDD Dictionary and supporting its maintenance.

The use of the RDD System will facilitate the accurate exchange and processing of information between interested parties involved in the administration of rights in, and use of, Digital Items, and the Rights Expression Language.

The RDD System is designed to support the mapping of Terms from different namespaces. Such mapping will enable the transformation of metadata from the terminology of one namespace (or Authority) into that of another namespace (or Authority). Mapping, to ensure minimum ambiguity or loss of semantic integrity, will be the responsibility of the Registration Authority.

The RDD Dictionary is a prescriptive Dictionary, in the sense that it defines a single meaning for a Term represented by a particular RddAuthorized TermName, but it is also inclusive in that it can recognize the prescription of other Headwords and definitions by other Authorities and incorporates them through mappings. The RDD Dictionary also supports the circumstance that the same name may have different meanings under different Authorities. Therefore Terms that are directly authorized by the RDD Registration Authority neither define nor prescribe intellectual property rights or other legal entities.

The RDD defines the meaning for the terms defined in the REL. Next table summarizes the ActTypes in this part of the MPEG-21 standard that have been defined in response to requirements identified in the process of developing the REL and RDD Standards, particularly focussed on common processes in the use and adaptation of Digital Resources.

ActType	Parent	Definition
Adapt	Derive, ChangeTransiently	To ChangeTransiently an existing Resource to Derive a new Resource.
Delete	Destroy	To Destroy a DigitalResource.
Diminish	Adapt	To Derive a new Resource which is smaller than its Source.
Embed	Relate	To put a Resource into another Resource.
Enhance	Adapt	To Derive a new Resource which is larger than its Source.
Enlarge	Modify	To Modify a Resource by adding to it.
Execute	Activate	To execute a DigitalResource.
Install	UseTool	To follow the instructions provided by an InstallingResource.
Modify	Change	To Change a Resource, preserving the alterations made.
Move	Modify	To relocate a Resource from one Place to another.
Play	Render, Perform	To Derive a Transient and directly Perceivable representation of a Resource.
Print	Render, Fix	To Derive a Fixed and directly Perceivable representation of a Resource.
Reduce	Modify	To Modify a Resource by taking away from it.
Move	Modify	To relocate a Resource from one Place to another.
Play	Render, Perform	To Derive a Transient and directly Perceivable representation of a Resource.
Uninstall	UseTool	To follow the instructions provided by an UninstallingResource.

Table 3 – RDD ActType and its parent and definition

When using RDD actTypes in REL is not only important the meaning of the act, it is also important the RDD Hierarchy of the term. In next chapters we will see how it influences the authorization process. An authorization can be erroneous if the hierarchy of the right is not taken into account.

Intellectual Property Management and Protection (IPMP) Components

Intellectual Property Management and Protection [5], part 4 of the MPEG-21 standard is in the Committee Draft (CD) phase of MPEG standardisation process. MPEG has defined the requirements for MPEG-21 IPMP based on input from a wide variety of interested parties.

In this part of the standard an interoperable framework for Intellectual Property Management and Protection will be defined. MPEG-21 must provide a framework that encourages the creation of new services that can be used to support new business models. These services should meet the needs of the different members of the networks associated with the distribution of digital items.

IPMP is central to the creation of these business models. It must provide more functionally than the existent technologies that have focused its efforts on content protection.

In this part of the MPEG-21 standard is included the expression and enforcement of rights that are associated with digital item distribution, management and usage by all members of the value chain.

The Intellectual Property Management and Protection part deals with the standardisation of a general solution for the management and protection of Intellectual Property. Digital Items can be protected in order to ensure that the access to the contents is done according to the license terms. The solution lies in the use of protection techniques over the digital content, which makes it possible to deploy a business model that ensures the accomplishment of the license terms in a controlled way. These kinds of objects are called IPMP DIDL documents that consist of the protected object (or part of the DIDL document) and the IPMP information expressions. IPMP expressions contain protection information, such as the IPMP tools that protect the content, initialisation settings, keys, etc.; and governance information, such as licenses that govern the content or references to these licenses or license services. In our system, the protection server is the responsible for protecting the content and managing the protection keys and tools. It can also generate the protection information required to be included in Digital Items.

4.1.3 MPEG-21 updates

4.1.3.1 REL Profiles

A Rights Expression Language (REL) is a machine-readable language that declares rights and permissions. The MPEG REL, as defined by ISO/IEC 21000-5, provides flexible, interoperable mechanisms to support transparent and augmented use of digital resources throughout the value chain in a way that protects the digital resource and honors the rights, conditions, and fees specified for it. The standard REL can support guaranteed end-to-end interoperability, consistency, and reliability among different systems and services. To do so, it offers richness and extensibility in declaring rights, conditions, and obligations; ease and persistence in identifying and associating these with digital content; and flexibility in supporting multiple usage/business models.

MPEG recognizes that different applications require different levels of complexity and flexibility in the REL and that specific industries and user communities may need to modify the language to better meet their specific needs. To facilitate easy mapping of the REL to these industry-specific applications, MPEG has developed a process of extending and profiling the language to create MPEG REL profiles that can be used by these individual applications.

In simple terms, an MPEG REL profile consists of a subset of the MPEG REL with some application oriented extensions in terms of its types, elements, attributes and authorization model, and is usually defined as the result of first extending (when needed) and then profiling the language with the extensions. Extending the REL enables users of the REL to define new types, elements and attributes specific to their needs. This includes extending the REL schematic elements with new ones for new the basic entities of the REL, namely, principal, right, resource and condition, to improve efficiency in a specific domain. Profiling the REL enables users to select only those language elements, as a subset of the language, required to meet a specific application need. This optimizes payload of digital items and computation requirements of MPEG terminals. An MPEG REL profile, resulting from extending and profiling the REL, can thus be used to optimize the applicability of the REL to one specific application.

<u>Overview of the REL Profiles</u>

Currently there are two MPEG REL profiles under development. One is the REL MEM profile, and the other is the REL DAC profile. The MEM (Mobile And optical Media) profile is based on requirements from mobile and optical media applications, whereas the DAC (Dissemination And Capture) profile is based on requirements from broadcast related applications.

<u>The REL MAM Profile</u>

The MPEG-21 REL MAM profile is for distributing and consuming content delivered via channels including mobile communications and pre-recorded optical media to devices like mobile phones and high-definition DVD players.

To support applications in the mobile and optical media domains, the MAM profile extends the MPEG REL with the following elements:

- Principals identified with identifiers from a given identification system.
- Rights to copy and move digital resources according to pre-determined rules defined by some governance organization.
- Resources protected by some form of (symmetric key and/or public key) encryption.
- Conditions to derive digital resources based on their relations with other resources when creating a collection of resources like a play-list and a customized movie, conditions to copy, move and output digital resources to other DRM systems, and conditions to seek dynamic permissions at the time of using digital resources.

The MAM profile consists of elements for

- Principals that can be identified by either an identifier in some identification system or a cryptographic key.
- Rights to play, print and execute resources, to copy and move according to rules defined by a governance body, and to possess attribute properties (related to, say, subscribers or authorization domains).
- Resources whose confidentiality and integrity are protected or not protected.
- Conditions related to time intervals, usage counts and territories, conditions to copy, move and output digital resources to other DRM systems, and conditions to seek dynamic permissions at the time of using digital resources.

The MAM profile supports usage models like

- Unrestricted usage
- Usage tied to individuals
- Preview
- Super-distribution
- Subscription based usage
- Usage within authorized domains

The REL DAC Profile

The MPEG-21 REL DAC profile is for representing the usage right of the digital broadcast contents, which are delivered from broadcasting company to DTV receiver or set-top-box, and then used on DTV or DVR.
Unlike other types of DRM protected content, broadcast content has some different features like the following:

- Its transmission media is usually packet-based live signal rather than file-based downloading.
- It can be transferred as a CLEAR signal from broadcasting company to DTV receiver or set-top-box. e.g. Terrestrial Broadcast
- It has possibility to be transferred to another legacy device in the form of non-protected signal through the output port on rendering device while it is rendering.
- It is recommended that the broadcast program be able to be used freely on personal digital network environment (PDNE)

To support these unique features of broadcast contents, MPEG-21 REL DAC Profile covers the Rights Expression to specify following rights or conditions as well as existing MAM (Mobile And optical Media) profile functions.

- time-shift-operation control
- simultaneous access control
- output signal control
- store control
- specifying destination entity and condition
- specifying security level
- dynamic license updating

4.1.3.2 Intellectual Property Management and Protection (IPMP) Components

Intellectual Property Management and Protection [5], part 4 of the MPEG-21 standard deals with the standardisation of a general solution for the management and protection of Intellectual Property. Digital Items can be protected in order to ensure that the access to the contents is done according to the license terms. The solution lies in the use of digital signatures and encryption techniques over the digital content, which makes it possible to deploy a business model that ensures the accomplishment of the license terms in a controlled way.

The MPEG-21 IPMP Components (ISO/IEC 21000-4) standard addresses the need for effective management and protection of intellectual property in the ISO/IEC 21000 MPEG-21 multimedia framework over heterogeneous access and delivery infrastructures. This specification is currently at FDIS level .MPEG-21 IPMP Components represents a flexible and extensible framework for applying protection mechanisms to a Digital Item. It purposely does not specify protection measures, keys, key management, trust management, encryption algorithms, certification infrastructures or other components. In its current form, the specification is designed to be applicable to a wide range of application areas.

IPMP Base Profile

Similarly to previously developed MPEG standards, the utility of this standard will be enhanced by the careful identification and development of specific profiles supporting functionality relevant to various user communities. In that spirit, we propose a “base” profile aimed at supporting use cases in widespread use in the area of commercial content distribution.

This profile is specifically aimed at supporting use cases in widespread use in the area of commercial content distribution. This proposed Base Profile purposely provides a limited scope in order to facilitate the implementation in devices with limited computational/storage capabilities.

In this contribution, we provide an updated text to revise our previous submission which is not in the status of profile under consideration.

Application Areas

This proposed Base Profile provides sufficient functionality to support current and emerging practices for distribution of commercial content, with a special focus on entertainment content such as movies and music, while reducing the requirements on end devices (e.g. footprint, memory usage, computational power, storage).

List of Tools

The base profile is proposed by restricting the occurrence of some elements from the full set of MPEG-21 IPMP elements. The guidelines for the restrictions are as follow:

1. The base profile is designed to maintain the capability to express the protected Digital Item.
2. As base profile is designed to reduce the requirement on end devices (e.g. device footprint, memory usage, computational power, storage), the following properties are restricted:

   a. The base profile is designed to have no recursion. Recursion may consume more memory and computing power. Moreover, it may complicate the implementation.
   b. The base profile is designed to carry at most one protection tool in one document. Since there is no recursion, having more than one tool may be redundant as protecting a single resource with same tool many times gives no additional protection effect.
   c. The base profile is designed to have no duplication of elements in many locations. For elements that can be inserted in different locations, a suitable place among the current existing ones should be chosen to ease the implementation while not sacrificing the feature(s) offered by that element.
3. The base profile is designed to be simple for transfer. This requires the XML instance of the base profile should be as small and compact as possible. For example, rather than being able to carry the tool binary representation in the XML document (with base64 encoding format), it is better to just carry reference to it.
4. For simplicity, tool is assumed to be ready for use in the terminal. The procedure to acquire and/or install the tool in the terminal is out of scope of the base profile.

IPMP DIDL

As the base profile maintains the full capability to express the protected Digital Item; it shall include all the elements in the IPMP DIDL schema. These elements taken together are a Representation of the DID model that allows for inclusion of governance information.

IPMP General Information Descriptor

The Base Profile shall include the root element IPMPGeneralInfoDescriptor. It shall:

- Include ToolList element
- Optionally include LicenseCollection element. It may contain any number of RightsDescriptor elements (in case there are multiple assets in the digital item), although in most instances a single RightsDescriptor element is likely to be used. The RightsDescriptor in the Base Profile excludes the possibility of having an IPMPInfoDescriptor child
- Optionally include Signature element. If present, it is sufficient to ensure the integrity of the children elements.

The ToolList element shall:

- Include at most one instance of ToolDescription. Note that Tools to be used in IPMPInfoDescriptor can only reference elements in ToolList; this ensures that only a single Tool can be defined for any given DI.
- Provide no support for Signature

The ToolDescription element shall:

- Include IPMPToolID element
- Optionally include Remote element but without Signature element
- Provide no support of MemberOf element and its children
- Provide no support for Inline tool definition
- Provide no support for ConfigurationSettings
- Provide no support for RightsDescriptor. RightsDescriptor elements are carried all together under the LicenseCollection element
- Provide no support for Signature

IPMP Information Descriptor

The Base Profile shall include the root element IPMPInfoDescriptor as defined with the following constraints:

- Support for (at most) a single Tool
- No support for RightsDescriptor (no need to specify governance over the Tool itself). The RightsDescriptor elements are collected and expressed under the LicenseCollection element

- The element dsig:Signature remains optional.

The Tool element shall have the following constraint:

- No attributes are needed. Since there is at most one Tool, order is no longer relevant.
- No support for ToolBaseDescription. All tool definition is carried in the ToolList while the Tool element will only refer to it
- Include a ToolRef element
- Optionally support the InitializationSettings element. However, there will be no support for ToolBaseDescription. The only supported child element shall be InitializationData (specifying governance over the InitializationData is considered out of scope for this profile).
- No support for RightsDescriptor
- No support for Signature

4.1.3.3 Event Reporting

Event Reporting is required within the MPEG-21 Multimedia Framework to provide a standardised means for sharing information about Events amongst Peers and Users. Such Events are related to Digital Items and/or Peers that interact with them. In the MPEG-21 context, the reporting messages that include information about different aspects of media usage are called Event Reports.

Event Reporting could be useful when monitoring of the usage of copyrighted material. The provider offering Digital Items for download would specify in an Event Report Request that, whenever a Resource within a Digital Item is rendered (e.g. played), he would receive an Event Report enabling him to manage his royalties. Upon rendering, the Peer will generate an ISO/IEC 21000 Event Report which will be delivered to the rights holder specified, in an Event Report Request, containing information about the Digital Item, the Resource, and the conditions under which it has been rendered.

Fundamentally, Event Reporting will facilitate interoperability between consumers and creators, thereby enabling multimedia usage information to be both requested and represented in a normalized way. Examples where Event Reports may be requested include usage reports, copyright reports, financial reports and technical reports.

On the other hand, an Event occurs when a set of specified conditions are met. The Event Reporting standard specifies the elements that can be used to define Conditions under which an Event is deemed to have occurred. These conditions could be:

Time-based that specifies a time period in which the Event must occur

DI-related operations that have been applied to the specified resource, defined by the Rights Data Dictionary

Peer-related operations: events that are related to the Peer itself rather than an instance of a Digital Item

Combinations thereof.

The basic model of Event Reporting indicates that Events that need to be reported may be specified by interested parties through the use of an Event Report Request (ER-R). An Event Report Request (ER-R) is used to define the conditions under which an Event is deemed to have occurred. Events defined by ER-Rs trigger the creation of an associated Event Report (ER), which contains information describing the Event, as specified in the associated ER-R.

The ER purpose is to indicate which Peer created it, define the data items that are to be included in such an Event Report(s), provide a reference to the originating ER-R, provide status information regarding its completion and creation, along with a free-form description.

When an Event Report is packaged within a Digital Item, the identity and access control specification of an ER is given through the use of two elements that need to be inserted at the top level of the item which contains an Event Report, before the Event Report Descriptor. These two elements are the Event Report identifier that will identify Event Report and their related Event Reports Requests using standard Digital Item Identification mechanisms and the Event Report access control information that will indicate specifies access rights which will apply to all ER-Rs, and all Event Reports that are generated as a result of processing an ER-R.

The ERR element serves as the root element for describing an entire Event Report Request. It consists of three elements the ERRDescriptor, the ERSpecification and the EventConditionDescriptor. Figure 14 depicts the structure of the ERR element.

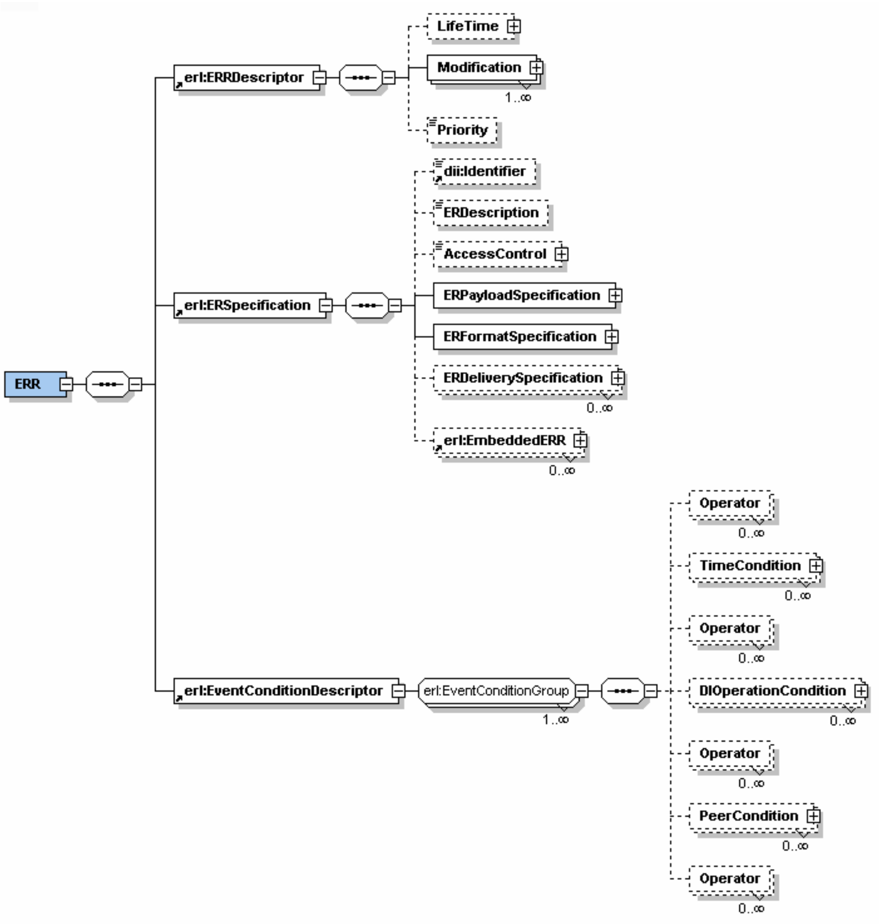


Figure 14 – Err element

The ERRDescriptor provides a descriptor of the Event Report Request including aspects as the lifetime of the ERR, the modification that contains the history of the ERR and the priority of the ERR.

The ERSpecification element provides information about the Event Reports that are created as the result of processing this ERR. This information includes a unique identifier to be used in the ER to be created as a result of this ERR, a free form field to provide comments on the ER, rights expressions specifying the peers and users that are allowed to access to specific parts of the ER, the data that must be reported, the identifier that should be used for all ERs that will be generated when this ERR is processed, the identifier and the geographical location of the Peer in which the event has occurred and the identifier of the users of this Peer, the operation that triggered the event, domain specific data, the format of the Event Report contained within the ReportData statement and the optional embeddedERR that contains an ERR that shall be processed upon receipt of the Event Report.

The ERConditionDescriptor element specifies the event conditions, then the occurrence of the event will trigger the creation and delivery of the ER. The ERConditionDescriptor could contain Operator(s) and at least one condition represented by the TimeCondition, DIOperationCondition or PeerCondition elements. The Operator element is used when de event will occur by the combinations of TimeCondition and/or DIOperantion conditions, or it can also be used to specificy multiple events within a certain EventCondition. The TimeCondition element specifies a time based condition. The DIOperationCondition element specifies conditions relating to operations on a Digital Item. Finally, the PeerCondition specifies other event conditions except time-based conditions and DI-related conditions. It enables users to define new event conditions as necessary.

On the other hand, in order to represent Event Reports has been defined the ER element that consists of three main elements the ERDescriptor, the ERData and the EmbeddedERR as shown in Figure 15.

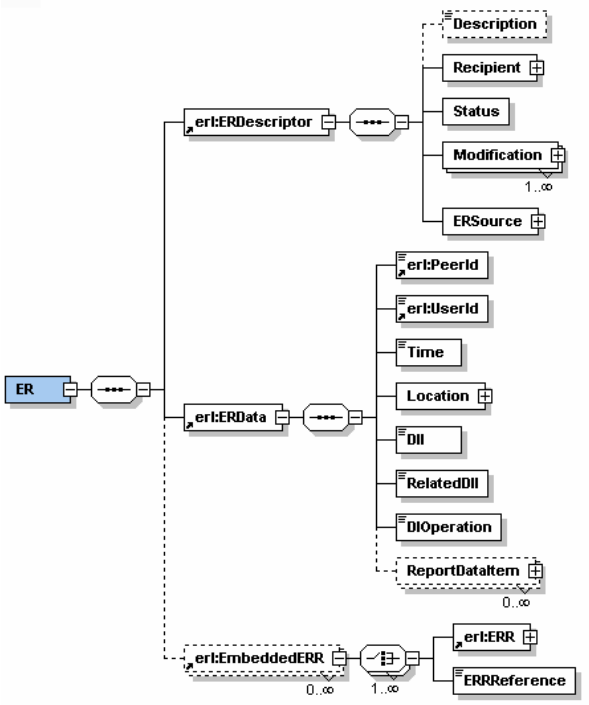


Figure 15 – ER element

The ERDescriptor element describes the Event Report. It contains the Description element that is a free form string field to provide comments on the Event Report. The Recipient element contains the identity of the User or Peer which is the intended recipient of the Event Report. The Status element that provides information on whether the Peer was able to compliantly generate the event report. The Modification element contains history of modifications or the ER. This field shall be used at least once to hold the information pertaining to the creation of the ER. In the ER, the first occurrence of this element will always describe the creation of the ER. And the ERSource element indicates the original source, for example an ER-R or a source application, that has created the ER. It is either specified as a reference to the ER-R or actually embeds the ER-R.

4.1.4 Digital Media Project (DMP)

The Digital Media Project (DMP) [29] is a non-profit Association registered in Geneva, Switzerland. Its mission is to promote the successful development, deployment and use of digital media that respect the rights of creators and rights holders to exploit their works, the wish of end users to fully enjoy the benefits of digital media and the interests of value-chain players to provide products and services.

4.1.4.1 DMP DRM

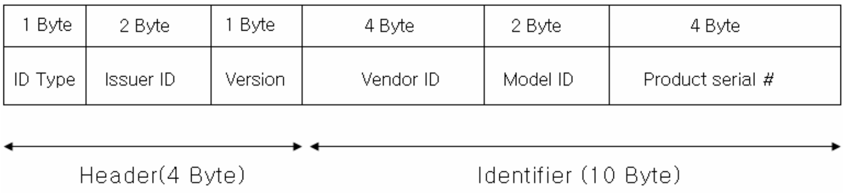
DMP has noted that DRM has the potential to substantially alter the balance that has been in existence in the analogue world between different Users of Content, in particular when one of them is the End-User. If not appropriately remedied, this imbalance may lead to a significant reduction of the scope of Traditional Rights and Usages (TRU) of Users. A possible outcome is the outright rejection of the new technology on the part of some Users, in particular End-Users perceiving the DRM media experience as inferior.

DMP is not claiming that an established TRU necessarily implies a *right* of a User to a particular Use of digital media but simply that, if Users have found a particular Use advantageous in the analogue domain, they are probably interested in continuing to exercise that Use in the digital domain as well. Leveraging upon this interest may provide opportunities for new “Digital Media Business Models” that are attractive to Users but respectful of the Rights of those who have created Works and invested in making Content.

Therefore DMP will be adding technologies to its specifications to make the exercise of TRUs technically possible. However, even a summary analysis shows that many TRUs have a legislative/regulatory impact that needs to be addressed by proper authorities. This can only be done within individual jurisdictions by determining which TRUs shall mandatorily be supported by the Interoperable DRM Platforms operating under their jurisdiction and which TRUs can be left to private negotiations between Users. This is a challenging task because it requires blending knowledge encompassing the legal, social and economic fields with in-depth knowledge of the highly sophisticated and unusual DRM technologies.

DMP has specified basic standard technologies [30], called Tools, that are required to build Value-Chains.For ease of treatment these Tools has been grouped in categories as listed below:

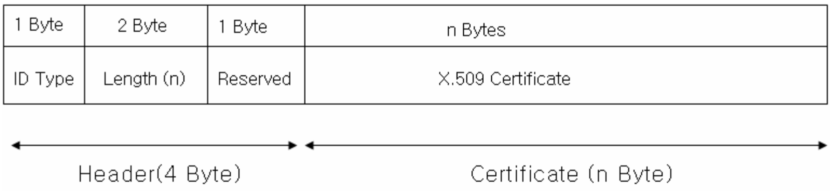
1. Represent: specifies the set of Tools used to Represent combinations of Resource Types and Metadata that constitute Content Items.
    - Content: DMP refers to this Content Representation Tool as DMP Content Information (DCI). DCI is an XML structure, based on a DMP-defined subset of the MPEG-21 Digital Item Declaration (DID) [3], MPEG-21 Digital Item Identification (DII) [4] and MPEG-21 IPMP Components [5], and extended by the DMP namespace to express DMP-specific information.
    - Keys: DMP has specified the element KeyInfo is in XMLDSIG [23] in order to enable the recipient(s) to obtain the key needed to validate digital signatures. KeyInfo may contain keys, names, certificates and other public key management information, such as in-band key distribution or key agreement data. The XMLDSIG specification defines a few simple types but applications may extend those types or altogether replace them with their own key identification and exchange semantics using the XML namespace facility. However, questions of trust of such key information (e.g., its authenticity or strength) are out of scope of the XMLDSIG specification and left to the application.
    - Rights Expressions: Specifies the Tool to Express Rights associated with Content that in turn map onto specific End-User Device behaviour consistent with the semantics of the Rights Expressions. It does not cover the expression of commercial offers or of details of financial transactions between the Service Provider and the End User.
2. Identify:
    - Content: The Content Identifier satisfies the characteristics defined in RFC 1737 [31], which is known as URN (Uniform Resource Names) scheme. Therefore identifiers that conform to URN schemes can be used to identify Content. Currently, there are several registered URN schemes such as ISBN and ISSN, each of them serving a specific purpose and having a unique namespace under IANA (Internet Assigned Numbers Authority).
    - License: License identification is performed as content identification, as a license can be represented as a DCI.
    - Device: Device identifier is mainly used for device authentication. There can be two kinds of device identification:
        - Device info-based identification: the identifier is generated only based on the device information by the device identification server run by the Registration Agency. Figure below shows the identifier format of device info based identification.



1 Byte	2 Byte	1 Byte	4 Byte	2 Byte	4 Byte
ID Type	Issuer ID	Version	Vendor ID	Model ID	Product serial #

Header(4 Byte)
Identifier (10 Byte)

        - Certificate-based identification: in which a X.509 certificate is utilized for device identifier. Figure below shows the identifier format for the certificate-based identification.



1 Byte	2 Byte	1 Byte	n Bytes
ID Type	Length (n)	Reserved	X.509 Certificate

- Domain: The allocation of Domain Identifiers is carried out by Domain Registration Agencies, which are appointed by the Domain Registration Authority. The format of the Domain ID is:

  <element name = "Domain_ID " type = "r:Keyholder"/>

  As a Domain is typically utilised by a group of Users it is often beneficial to be able to Identify sub-Domains within a given Domain, where each sub-Domain corresponds to a member of the group. The creation of a sub-Domain is achieved in the same way as the original Domain was established, this time generating a sub-Domain ID. The sub-Domain ID can be included in a Content License as representing the target of the Content in the same way as the Domain or Device ID. Multiple sub-Domains may appear within a Content License. Format of Sub-Domain ID is:

  <element name="SubDomain_ID" type="anyURI"/>

3. Package
   - Content: DMP provides Tools to Package Content in files whose format using a DMP-defined subset of the MPEG-21 File Format [10], which contains the DCI with some or all of its ancillary Resources, potentially in a single package. The MPEG-21 File Format is based on the ISO Base Media File Format, which defines how to contain timed media information for a presentation. The file structure is object-oriented; a file can be decomposed into constituent objects very simply, and the structure of the objects inferred directly from their type. Files are formed as a series of objects, called boxes. All data is contained in boxes; there is no other data within the file. Each Box is characterised by two attributes: boxtype and size.
4. Authenticate
   - Device: DMP defines three different types of Device Authentication, which are closely related with the Identification part of this Approved Document. This section will provide means to Authenticate Devices for the three classes of Device Identification.
     - Devices having unique certificates
     - Devices that are uniquely identified by data
     - Devices without a unique data with certificate proxy
5. Manage
   - Domains: DMP specifies the Domain Management Protocols. The functionality of these protocols includes:
     - Setting up a Device Domain Context
     - Controlling the Use of Content within the Domain.
     - Managing Device Domain membership – joining and leaving
6. Access
   - Content: DMP specifies the Protocol to Access a Content Item with a License Bundled within the Content and bound to a specific Device and/or Domain. The Remote Content Access Protocol (RCAP) is used by an eXternal Device (XD) connected to a PAV Device to obtain a Content Item with a License Bundled within, granting the Use of this Content Item to that PAV Device or to a Domain. This protocol is based on the exchange of messages between two basic components: the XD and the Content Provider.
   - License: DMP specifies two Protocols:
     - Remote License Access Protocol (RLAP) employed when an external device (XD) connected to a PAV Device Accesses a License from a License Provider;
     - Local License Access Protocol (LLAP) employed when a PAV Device Accesses a License Bundled within a Content Item.
   - Update/Upgrade License:
7. Process

  - Binarise XML
  - Encrypt/Decrypt

4.1.5 DMP DRM update

Media content has always played an important role in all societies and manifold technologies have been invented and deployed to provide means to store, distribute and consume it. The complexity of these technologies and the stimulus to provide ever-enhanced end-user experiences have created very complex media content value-chains populated by an increasing number of interacting intermediaries, each providing increasingly sophisticated services to the two extremes of the value-chains – creators and end users – as well as to the various intermediaries in between. Note that in DMP all players in the value chain – Creators, intermediaries and End-Users – are generically called Value-Chain Users or, simply, Users. Note that terms beginning with a capital letter are defined in the DMP Terminology [30].

Media value-chain technologies have been designed with two main purposes in mind: the first to provide or augment the end-user experience, and the second to provide or augment the capability to distribute media content. The latest round of technologies – the digital technologies – have augmented the end-user experience, e.g. by providing very high quality audio and video that does not deteriorate with time and use. Further digital networks have also dramatically increased the distribution potential of media content.

As a result the traditional means to manage the value of media content along the value-chain are rapidly losing their established meaning. This is the source of various difficulties and is the major cause of the poor exploitation of the potential of digital media technologies. Digital Rights Management (DRM) has been advocated by many as the set of technologies that can overcome these difficulties because Users are given the possibility to manage Content while it moves along the Value-Chain.

The Digital Media Project agrees that DRM has the potential to combine the benefit of digital technologies with the need for a virtuous circle that motivates Creators to continue creating because remuneration is facilitated by DRM technologies. However, DMP sees serious problems in the introduction of DRM technologies that are lacking Interoperability.

A DRM system can be described as a particular form of communication designed to provide controlled communication between two or more Users. Therefore the implementation of a DRM system may require a broad range of communication technologies. Unless these are designed in such a way as to enable communication of Content between two different implementations, DRM becomes an obstacle that prevent Users from having the seamless and rewarding communication that digital media technologies have enabled. This has particularly serious consequences in the case of the End-User because the lack of Interoperability detracts from the End-User experience and thus may seriously impede the take off of services designed to provide appropriate remuneration to relevant value-chain users.

Standards can bring benefits to the very special type of communication systems called DRM. However, the application of DRM standards obeys different rules because DRM is tightly connected to business practices. As the introduction of digital technologies is currently forcing changes in the way value-chain users conduct their business, it is hard to define today what kinds of standards are required, much less to forecast what kinds of standard will be needed in the future.

DMP approaches the problem of DRM Interoperability by specifying technologies – that DMP calls Tools – required to implement what DMP calls “Primitive Functions”. These are “smaller” functions obtained when the functions value-chain users perform when they do business between themselves are broken down into more atomic elements. It is expected that, while functions may undergo substantial changes as a consequence of the evolution of the media business in the value-chain, Primitive Functions will generally remain more stable.

Therefore DMP is not developing a universal “DRM standard” capable of providing interoperability between every variety of different Users in arbitrary Value-Chains or across different Value-Chains. DMP provides specifications of Tools enabling Primitive Functions along with examples of how Value-Chains serving specific goals can be set up using the standard Tools. DMP specifications are developed in phases, so as to achieve gradual development of standards technologies.

The DMP approach to DRM standardisation is based on the following process

1. For each phase Use Cases deemed to be significant are identified and documented;
2. Primitive Functions required to implement the selected Use Cases are singled out;
3. Requirements for Primitive Functions are developed through inputs from relevant Users;
4. Tools serving the needs represented by the Use Cases are standardised;
5. Calls for Proposals for Tools with the identified requirements are issued;
6. The Tools are selected and documented through an open process. DMP favours Tools that have already been developed, standardised or adopted by other bodies, possibly adapting them to DMP needs;
7. Specifications of how Tools can be assembled to implement the selected Use Cases are developed;
8. In subsequent phases, Calls for Proposals for additional Tools needed to support new Primitive Functions or additional functionalities of existing Tools are issued.

DMP calls the ensemble of all standardised DRM Tools “Interoperable DRM Platform (IDP)”. The IDP provides several major advantages:

1. The specifications are industry agnostic, i.e. Users are free to build a great variety of Value-Chains that suit their business models by combining the Tools appropriate for them;
2. The capabilities of a Value-Chain or new Value-Chains can be extended by adding more Tools, possibly through additional standardisation;
3. The cost to access standardised Tools may be reduce because in general Tools have multiple usages and may be provided by multiple suppliers;
4. Full interoperability can be achieved within a Value-Chain;
5. An enhanced degree of interoperability can be achieved between different Value-Chains;
6. Innovation can be continuously fed in the system.

4.1.5.1 DRM technologies

In spite of the value DMP attaches to Interoperable DRM as the main digital media-enabling technology, DMP has noted that DRM has the potential to substantially alter the balance that has been in existence in the analogue world between different Users of Content, in particular when one of them is the End-User. If not appropriately remedied, this imbalance may lead to a significant reduction of the scope of Traditional Rights and Usages (TRU) of Users. A possible outcome is the outright rejection of the new technology on the part of some Users, in particular End-Users who will perceive the media experience in a DRM environment as inferior.

DMP is not claiming that an established TRU necessarily implies a right of a User to a particular Use of digital media but simply that, if Users have found a particular Use advantageous in the analogue domain, they are probably interested in continuing to exercise that Use in the digital domain as well. Leveraging upon this interest may provide multiple opportunities for new “Digital Media Business Models” that are attractive to Users but respectful of Rights Holders.

Therefore DMP intends to add technologies to its specifications to make the exercise of a broad range of TRUs technically possible. However, even a summary analysis shows that many TRUs have a legislative/regulatory impact that needs to be addressed by proper authorities. This can only be done within individual jurisdictions by determining which TRUs shall be mandated in Interoperable DRM Platforms operating under their jurisdiction and which TRUs can be left to private deals between Users. This is a challenging task because it requires blending knowledge encompassing the legal, social and economic fields with in-depth knowledge of the highly sophisticated and unusual DRM technologies.

4.1.5.2 DMP Specifications

DMP has produced the following specifications in their last meeting:

1. **Value Chain Functions and Requirements** [31]: a collection of Primitive Functions derived from today's media value-chains with corresponding Requirements.
2. **Architecture** [32]: a general architecture that describes some of the digital extensions of today's media value-chains and collects the basic assumptions and technologies underlying the establishment of IDP-enabled Value-Chains.
3. **Interoperable DRM Platform** [33]: a collection of technical specifications of basic Tools that are needed to implement Primitive Functions.
4. **Use Cases and Value Chains** [34]: a collection of all Use Cases along with normative specifications of examples of (portions of) Value-Chains implementing the Use Cases using the Tools drawn from the IDP Toolkit.
5. **Certification and Registration Authorities** [35]: a set of operational rules for Certification Authorities established to Certify Devices and DRM Tools, and Registration Authorities established to Assign Identifiers to Content, DRM Tools, Devices, Users and Domains.
6. **Terminology** [30]: a set of terms and corresponding definitions that are used throughout DMP ADs providedto overcome the problem of DRM being a new field that impacts many existing fields with their own established and sometimes conflicting terminologies.

In addition DMP is currently developing the following ADs:

7. **Reference Software** [36]: a software implementation of IDP Tools. DMP strives to provide the reference software as Open Source, with a license aligned to established practices. When this is not possible DMP provides the reference software with a "modify, use and distribute" license.
8. **End-to-End Conformance** [37]: a set of Recommended Practices that Value-Chain Users can reference to ascertain that the Tools employed by other parties conform to DMP Technical Specifications and Technical References.
9. **Mapping of Traditional Rights and Usages to the Digital Space** [38]: a set of example support of TRUs using DMP Tools possibly complemented by recommendations to appropriate authorities to enable the benefit of TRUs in a DMP-enabled world of digital media.

4.1.6 ISMA (Internet Streaming Media Alliance) DRM

The Internet Streaming Media Alliance [28] is composed of companies from the information technology, consumer electronics and media industries. Apple Computer, Cisco, IBM, Kasenna, Philips and Sun Microsystems Inc. are the founder members of the ISMA. In addition there are more than 30 other companies in the ISMA. They are jointly specifying protocols for media streaming over IP networks.

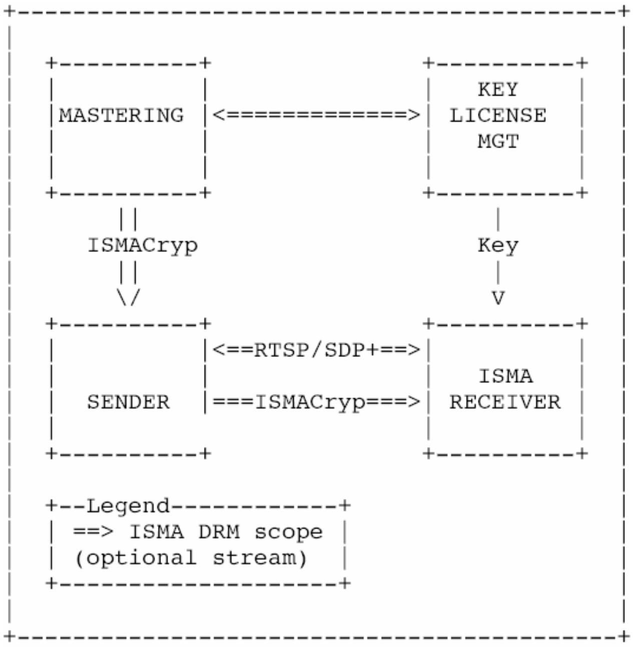
The ISMA/DRM must preserve the ISMA interoperability goals using standard encryption, authentication and integrity validation for ISMA conforming media and protocols. There are three general goals for the first release of the ISMA DRM specification according to the ISMA DRM Recommendations.

1. Ensure the "support and consistency" of ISMA 1.0 specifications when DRM is added.
2. Remove technical barriers to the dissemination of rights-managed content on platforms that run the ISMA protocols. If ISMA media and protocols are to serve as open, standard interfaces to "content protection" devices, ISMA needs to accommodate the technical protection measures (TPM) in those devices and it needs to implement the change control needed for platform licensing.

3. Identify what needs to be standardized, which bodies are developing needed standards and what needs to be invented. For example, MPEG is developing the decoder interfaces and services for DRM; the IETF, SMPTE, and OMA are developing cryptographic and key management protocols. The IETF defines cryptographic protocols for IP-network applications, particularly for the high-security needs of governments, enterprises, and individuals.

ISMA DRM Architecture

This section briefly reviews the ISMA DRM architecture. Figure below summarizes the flow within the architecture.



```
+------------------------------------------+
|                                          |
|  +----------+             +----------+   |
|  |          |             |   KEY    |   |
|  |MASTERING |<===========>| LICENSE  |   |
|  |          |             |   MGT    |   |
|  |          |             |          |   |
|  +----------+             +----------+   |
|      ||                        |         |
|   ISMACryp                    Key        |
|      ||                        |         |
|      \/                        V         |
|  +----------+             +----------+   |
|  |          |<==RTSP/SDP+==>|          |   |
|  |          |             |   ISMA   |   |
|  |  SENDER  |===ISMACryp===>| RECEIVER |   |
|  |          |             |          |   |
|  +----------+             +----------+   |
|                                          |
|  +--Legend------------+                  |
|  | ==> ISMA DRM scope |                  |
|  | (optional stream)  |                  |
|  +--------------------+                  |
|                                          |
+------------------------------------------+
```

Figure 16 – ISMA DRM architecture

In previous figure, mastering is where a content work is prepared for dissemination. It may be encrypted and associated with a rights specification that is formatted according to a "rights expression language". Mastering is important to interoperability when cryptographic transforms are applied: The standard receiver needs standard cryptographic transforms for the decryption, authentication, and integrity of content works. The Mastering step may apply encryption to the mastered media work.

The Key/License MGT entity associates a rights specification and cryptographic keys with an ISMA content work. Key/License MGT translates the rights specification into a license. The license authorizes particular types of access to the work, possibly according to a set of "business rules." The access may be at a highly-granular level of access such as to view/hear the content, write to a DVD, or send to a friend. Key/License Management is referenced but not specified; ISMA needs to accommodate a variety of Key/License management systems, both standard and proprietary.

Although the previous figure shows Key/License MGT providing only keys to the Receiver, the Sender may obtain the key using the same elements of procedure as the Receiver. The process may be completely different on the Sender if the content is pre-encrypted at the Mastering step leaving the Sender with no need to hold the key to the content work.

The Receiver decrypts and authenticates content works contained in the media flow and may decrypt and authenticate control flows. Depending on the nature of the key management protocol in use, the Receiver may perform mutual authentication with the Key/License MGT entity to prove that the receiver is an authorized platform. This process is controlled by the license, which specifies the terms and conditions under which a key is provided to an ISMACryp device. The license determines what authenticating information is exchanged, such as information about the Receiver’s hardware, software or human user. This information needs to be governed by a specification as to what can be collected and how it can be used. Exchanges with

Key/License MGT need to be secured in practically all circumstances to protect the user identity and the user's content-work transactions as well as the content-work keys. The media decryption keys must also be secured, and the receiver may be a licensed content-protection platform (CPRM). The first release of ISMA DRM supports but does not specify the interfaces, messaging, or processing of content-protection platforms.

Thus, the information assets to be protected go beyond content works; they include information related to the user's privacy and authenticating information. These assets also include resources such as the CPU, storage, service and bandwidth of the provider and the user.

4.1.7 Creative Commons

Introduction

Creative Commons [39], an initiative launched in the beginning of 2003. A non-profit organization founded on the notion that some people would prefer to share their creative works (and power to copy, modify, distribute works) instead of exercising all of the restrictions of copyright law. It offers a flexible solution that situates itself between traditional copyright and the strict open source and open content licenses. It allows rights owners to specify that their content is "Some Rights Reserved", not "All Rights Reserved". Users can choose and combine a custom-made license, based on four fundamental choices. Should the work always be attributed to its original maker? Can the work be reused for commercial purposes? Are derivatives allowed or should the work be reproduced in its original form? Should the work be reproduced under the same conditions?

Creative Commons legal solidness, flexibility and simplicity, combined with the wide exposure it received, makes it a promising new initiative - many agree that more flexibility is needed in order to bring the current, disproportioned copyright system back to a realistic framework.

Creative Commons main goals are Goals to provide an easy way to announce that works are available and an easy way for people to find such works that are in the public domain or licensed on generous terms

Creative Commons licenses

Creative Commons first project, in December 2002, was the release of a set of copyright licenses free for public use. Taking inspiration in part from the Free Software Foundation's GNU General Public License (GNU GPL), Creative Commons has developed a Web application that helps people dedicate their creative works to the public domain — or retain their copyright while licensing them as free for certain uses, on certain conditions. Unlike the GNU GPL, Creative Commons licenses are not designed for software, but rather for other kinds of creative works: websites, scholarship, music, film, photography, literature, courseware, etc.

They hope to build upon and complement the work of others who have created public licenses for a variety of creative works. Their aim is not only to increase the sum of raw source material online, but also to make access to that material cheaper and easier. To this end, they have also developed metadata that can be used to associate creative works with their public domain or license status in a machine-readable way. They hope this will enable people to use their search application and other online applications to find, for example, photographs that are free to use provided that the original photographer is credited, or songs that may be copied, distributed, or sampled with no restrictions whatsoever. They hope that the ease of use fostered by machine- readable licenses will further reduce barriers to creativity.

Creative Commons offer to work creators the possibility offer their work using CC licenses without giving up their copyright. It means offering some of creators rights to any taker, and only on certain conditions.

The different licenses offered by Creative Commons are:

- Attribution. You let others copy, distribute, display, and perform your copyrighted work — and derivative works based upon it — but only if they give you credit.
- Noncommercial. You let others copy, distribute, display, and perform your work — and derivative works based upon it — but for noncommercial purposes only.
- No Derivative Works. You let others copy, distribute, display, and perform only verbatim copies of your work, not derivative works based upon it.
- Share Alike. You allow others to distribute derivative works only under a license identical to the license that governs your work.

Creative Commons Metadata files have two major parts: a work description, and a license description. The work description uses Dublin Core properties to provide information about the work. Here are the properties that are used:

- dc:title:A title or name for the resource.
- dc:description: A text description of the resource.
- dc:subject: Key words and phrases describing the topic of the resource.
- dc:publisher: A cc:Agent responsible for making the resource available.
- dc:creator: A cc:Agent who created the resource.
- dc:contributor: An cc:Agent who contributed to the creation of the resource.
- dc:rights: The cc:Agent who holds the copyright on the resource.
- dc:date: A copyright date for the resource.
- dc:format: The Media Type of the resource.
- dc:type: The DCMI Type (schema) of the resource.
- dc:source: A Work that the resource was derived from.
- cc:derivativeWork: A Work that was derived from the resource.
- cc:license: A copyright license for the resource, a structured cc:License. If there are two cc:licenses, then the licensee gets to pick which to use.

Licenses are described by their characteristics, which come in three types:

- Permissions (rights granted by the license)
    - Reproduction: the work may be reproduced
    - Distribution: the work (and, if authorized, derivative works) may be distributed, publicly displayed, and publicly performed
    - DerivativeWorks: derivative works may be created and reproduced
- Prohibitions (things prohibited by the license)
    - CommercialUse: rights may be exercised for commercial purposes
- Requirements (restrictions imposed by the license)
    - Notice: copyright and license notices must be kept intact
    - Attribution: credit must be given to copyright holder and/or author
    - ShareAlike: derivative works must be licensed under the same terms as the original work
    - SourceCode: source code (the preferred form for making modifications) must be provided for all derivative works

Example of what a full file will look like:

```
<rdf:RDF xmlns="http://web.resource.org/cc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<Work rdf:about="http://example.org/gnomophone.mp3">
  <dc:title>Compilers in the Key of C</dc:title>
  <dc:description>A lovely classical work on compiling code.</dc:description>
  <dc:creator><Agent>
    <dc:title>Yo-Yo Dyne</dc:title>
  </Agent></dc:creator>
  <dc:rights><Agent>
    <dc:title>Gnomophone</dc:title>
  </Agent></dc:rights>
  <dc:date>1842</dc:date>
  <dc:format>audio/mpeg</dc:format>
  <dc:type rdf:resource="http://purl.org/dc/dcmitype/Sound" />
  <dc:source rdf:resource="http://example.net/gnomovision.mov" />
  <license rdf:resource="http://creativecommons.org/licenses/by-nc-nd/2.0/" />
  <license rdf:resource="http://www.eff.org/IP/Open_licenses/eff_oal.html" />
</Work>
```

```
<License rdf:about="http://creativecommons.org/licenses/by-nc-nd/2.0/">
  <permits rdf:resource="http://web.resource.org/cc/Reproduction" />
  <permits rdf:resource="http://web.resource.org/cc/Distribution" />
  <requires rdf:resource="http://web.resource.org/cc/Notice" />
  <requires rdf:resource="http://web.resource.org/cc/Attribution" />
  <prohibits rdf:resource="http://web.resource.org/cc/CommercialUse" />
</License>
</rdf:RDF>
```

4.1.8 Real Networks DRM

Introduction

In 1995, RealNetworks, Inc. [40] pioneered the entire Internet media industry, and continues to fuel its exponential growth.Because the Internet was built to handle text-based information, not audio and video and other rich media, RealNetworks, Inc. foresaw the need for specific solutions that could handle the creation, delivery and consumption of media via the Internet. That led RealNetworks, Inc. to invent and release the RealPlayer and RealAudio in 1995.

RealNetworks, Inc. develops end-to-end solutions that allow everyone — from Fortune 500 companies with locations worldwide to individuals at their desktops or other Internet-enabled devices — to create, send and receive audio, video and other multimedia services over the Internet. With the introduction of the Helix Initiative and the Helix Servers and RealProducer product families from RealNetworks, Inc., they continue to lead the way in delivering the highest quality Internet media experience delivering any media format, from any point of origin, across any network transport, running any OS to any person on any Internet-enabled device anywhere in the world.

Helix DRM [41] replaces the company's Media Commerce Suite (MCS) [42], which provided DRM only for RealAudio and RealVideo formats, and is part of RealNetworks ambitious open-source push for adoption among content producers and consumer electronics manufacturers

Helix DRM 10

Helix DRM is the first multiformat digital rights management platform for secure delivery of media to any device. It is a comprehensive and flexible platform for the secure media content delivery of standards-based as well as leading Internet formats, including RealAudio, RealVideo, MP3, MPEG-4*, AAC*, H.263 and AMR*. Helix DRM makes it possible to deliver these formats not only to PCs but also to a wide array of non-PC devices, including mobile devices and home appliances.

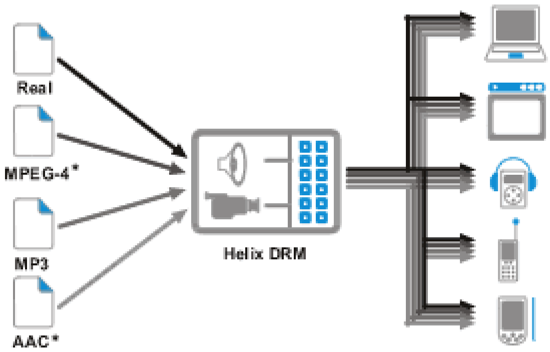


Figure 17 – Helix DRM overview

Helix DRM includes a set of products and services enabling business models through secure rights managed distribution of movies, music and other digital content to millions of media player users worldwide.

It empowers rights holders to create a range of robust business models for distributing rich media to the broadest worldwide audience, and will create many new ways for consumers to access and enjoy digital media.
Helix DRM provides secure media packaging, license generation and high-quality content delivery to a trusted media player base across all major platforms to multiple devices. It extends the RealPlayer and Helix Platform open architecture to accommodate the incorporation of a wide range of rights management systems. It easily integrates into all types of existing infrastructures and back-end systems, supporting a broad set of business models including purchase, rental, video on-demand, and subscription services.

Consumers want to access their favorite audio and video content via the Internet and play it on a variety of home and mobile devices. Content owners want to meet this demand in a way that protects their copyrighted material. Until now, that has required them to use a different digital rights management (DRM) system for each major format they want to support, or to avoid making their content available at all for certain devices.

Helix DRM is a complete, end-to-end secure digital delivery platform that is comprised of four major components Helix DRM Packager, Helix DRM License Server, elix DRM Client and Helix DRM Device Support .

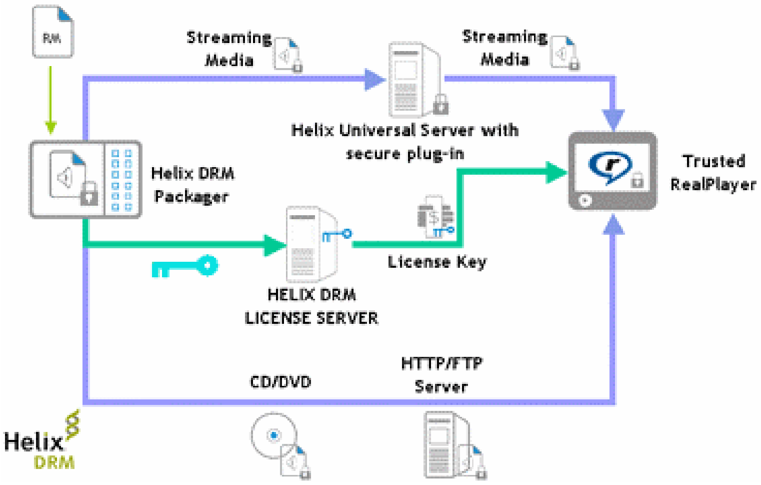


Figure 18 – Helix DRM

Helix DRM Features

General features

Helix DRM general features are:

- The content is separate from the rights. This means content owners can change the business rules associated with the content without re-encoding or re-packaging.
- Supports multiple usage rights: Content owners have the ability to issue licenses for playback of a specific duration, playback during a specific window of time, and to limit the number of plays for each media file distributed.
- Supports multiple business models: such as rental, subscription services, content syndication, content transaction and promotion, pay-per-view and video on demand.
- Supports multiple content delivery modes: Secure content can be distributed content via streaming (live or on-demand), downloads, physical media or peer-to-peer file sharing networks.

Back-end system interoperability

Helix DRM was built to integrate with existing systems such as clearinghouses, payment systems, retail storefronts, databases, customer relationship management software, and other essential commerce systems.
Real Networks currently offer two models for integration of Secure Receivers through their Helix Device DRM Community:

- Native Device Support: A Helix DRM Native Device is a network-connected device such as a PC, Set Top Box, Home Media Server, or Mobile Phone. The Native Device can connect directly to the Internet, and has local storage. Helix DRM Native Device support requires both the Helix DRM and the Helix DNA client to run natively on the consumer device. With the Helix DRM Native Device implementation, consumers are able to perform content acquisition, licensing, and playback all on the Native Device. The network connection allows for content retrieval for download and streamed presentations as well as request and receipt of content licenses from the Helix DRM License Server. The storage on the Native Device holds the secure license store containing the content licenses that have been issued as well as any downloaded content. Playback of the media through the Helix DNA Client with the Helix DRM Client Plug-in provides high quality playback of the widest variety of content formats as well as enforcement of the business rules defined by the content owner.
- Secure Receiver Device Support: Helix DRM for Secure Receiver Devices enables Consumer Electronics Manufacturers to enable Industry Supported, Standards Based, Industry Leading secure media experiences such as subscription & purchase business models on a wide range of consumer, mobile, and portable devices as well as connected home appliances. A Secure Receiver Device is a device that requires a connection to a Native Device for the transfer of content and enforcement of the business rules defined by the content owner.

Helix DRM Components

Helix DRM Packager

The Helix DRM Packager uses strong encryption algorithms and secure container technology to prevent unauthorized use of content and to prepare content for distribution via streaming, download or other delivery methods. The packaged media content and the associated business rules for unlocking and using that content are stored separately, so that multiple sets of business rules can be applied to a single file over time. The Helix DRM Packager can support a wide range of media formats and can deliver secure live content when used in conjunction with the RealProducer.

Helix DRM License Server

The Helix DRM License Server is a scalable, flexible server that allows retailers, Internet music and movie services, and enterprises to manage, authorize, and report content transactions. The Helix DRM License Server verifies content licensing requests, issues content licenses to trusted, authenticated Helix DRM end-user clients, such as RealPlayer, and provides auditing information to facilitate royalty payments. The content owner, in the event of a security Player breach, can also revoke licenses.

Helix DRM Client

The Helix DRM client enables download and streaming playback of secure formats in a tamper-resistant environment based on the usage rules specified by the content owners. Client applications, such as the RealPlayer can be built on top of the Helix DRM client.

Helix DRM Device

The evolution of digital media is moving quickly beyond the PC. Consumers throughout the world are moving to take their music with them wherever they go. As connected and non-connected audio and video devices become more prevalent, it becomes critical to enable new business models that take advantage of consumer demand. Content rights holders are quickly seeking proven, reliable methods to deliver their content securely beyond the PC. To secure the media content that is on consumer devices, Helix DRM for

Devices enables chip and consumer electronic device manufacturers to include DRM on their consumer devices and to keep content secure beyond the PC. As part of Real strategic initiatives to deliver media anytime, anywhere, and on any device, the flexibility of Helix DRM means that content rights holders can determine the rules for which they want to deliver content to devices. It also means that device manufacturers can provide a complete DRM solution on their devices that will meet the needs of consumers.

Helix DRM 10 for Devices works in two different ways-Primary Device and Secondary Device -thus giving content rights holders and device manufacturers maximum flexibility over their business models.

- **Helix DRM 10 Primary Device**: it is a network-connected device such as a PC, Set Top Box, Home Media Server, or Mobile Phone. The Primary Device can connect directly to the Internet, and has local storage. Helix DRM 10 Primary Device support requires both the Helix DRM 10 and the Helix DNA client to run natively on the consumer device. In the Helix DRM 10 Primary Device implementation, the consumer acquires the secure media file from a media server, and then attempts to play the encrypted file on their Helix DRM 10 Primary Device. Because a license key is required to decrypt the secure media file, the Helix DRM 10 client running on the Primary Device makes a license request and the Helix DRM 10 License Server generates a license with the license key and the associated business rules defined by the content owner.
- **Helix DRM for Secondary Devices**: it enables Consumer Electronics Manufacturers to enable Industry Supported, Standards Based, Industry Leading secure media experiences such as subscription & purchase business models on a wide range of consumer, mobile, and portable devices as well as connected home appliances. A Secondary Device is a device that requires a connection to a Primary Device for the transfer of data and rules for that data. There are two models for integration with Secondary Devices through our private Helix Device DRM Community:
    - Secure Memory Device - With the Secure Memory protocol, manufacturers implement device compatibility with stored secure content and enable the transfer of secure content to the device from an application with native Helix DRM 10 support, such as the RealPlayer 10. By enabling the Secure Memory protocol, CE manufacturers will be able to offer secure download services such as the RealPlayer 10 Music Store to their consumers.
    - Secure Streaming Device - With the Secure Steaming device protocol, device manufacturers allow secure content to be streamed to their devices over a home network. By enabling the Secure Streaming protocol, CE manufacturers can make their devices compatible with premium content services to help drive demand for their device technology.

RealNetworks does plan to enable Secure Streaming via Digital Transmission Copy Protection over Internet Protocol (DTCP-IP).

4.1.9 Trymedia ActiveMark DRM SYSTEM

ActiveMARK DRM [43] is also the only single protection solution that works on both physical media (CD/DVD) and via download. ActiveMARK DRM ensures that content is used only as intended. Rather than preventing copies from being made by consumers, ActiveMARK is designed to enable copies to revert-to-trial mode (if required), creating additional sales without incremental cost. Additionally, consumers may make backup copies of their content without exposing any additional security risk.
ActiveMARK DRM performs the following client-side operations:

- Business model support
- Rules enforcement
- User interfacing
- License management
- Security

Business model support

ActiveMARK DRK support different business models, the most common are:

- CD Activation: Product is purchased offline and is then activated to gain access to copying/backup features
- Try Before You Buy: Consumers can try your products, with limitations, before making a purchase decision
- Rent and Buy: Consumers can pay to try your products (rent) before deciding to buy
- Buy without Trial: Consumers can/must purchase your content without being able to run a trial period
- Subscription: Consumers pay a monthly fee for access to a range of content. Access is turned on or off based on the subscription status
- Beta/Demo Uses (non-commercial): Content cannot be purchased, but can be used in trial mode up to a specified limit (time, date, etc.)

Rules enforcement

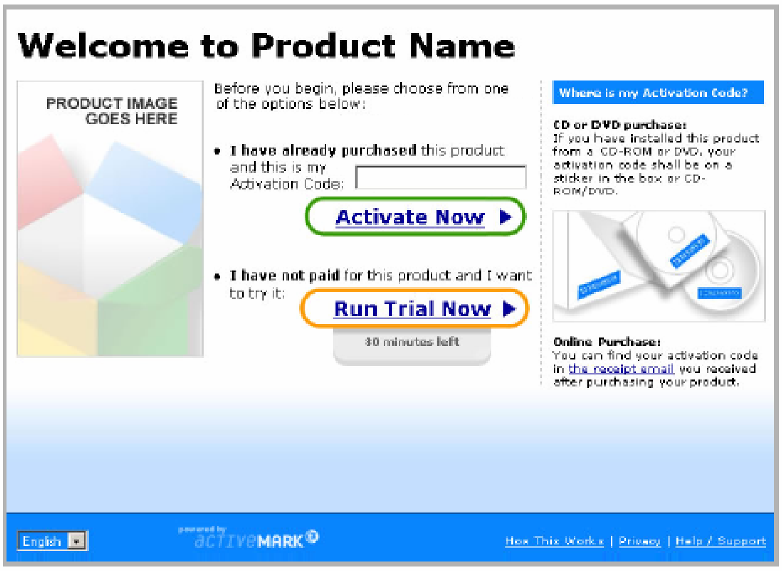
Rules enforcement articulates how the above business models are implemented. The content owner specifies rules that grant different access controls to his/her content during its lifecycle. Supported rules are:

- Time-based: Allow a trial version to be used during a specified time period
- Use-based: Allow a trial version to be used a certain number of times
- Date-based: Allow a trial version to be used until a certain data
- Space-based Allow a trial version to be played through a certain number of levels or disable the print function
- Complex: Allow a trial version to be used a certain number of times unless the date is before to certain date, and then allow a certain time of use
- Authentication-based: If user has a valid subscription, then allow use for a certain time period and check validity again

User interfacing

It is handled via in-product HTML pages. Pages are typically displayed at launch, close and trial expiration with an emphasis on encouraging consumers to purchase.

Interfaces are packaged within the product itself, allowing users to render and conduct basic operations without needing an Internet connection. These branded pages present product information along with buttons to share it with friends, play trials, reactivate (when licenses have been lost), activate and purchase the product.



Welcome to Product Name

Before you begin, please choose from one of the options below:

- **I have already purchased** this product and this is my Activation Code:

- **I have not paid** for this product and I want to try it:

CD or DVD purchase: If you have installed this product from a CD-ROM or DVD, your activation code shall be on a sticker in the box or CD-ROM/DVD.

Online Purchase: You can find your activation code in the receipt email you received after purchasing your product.

Figure 19 – User interface example

License management

In this process licenses are created, stored and validated on the user's computer.
Typically, a license is requested once a consumer clicks on "BUY" or "ACTIVATE" and payment/authentication is validated. Licenses are created based on the configuration of the user's computer, and allow a customizable level of tolerance for configuration changes before requiring reactivation. These licenses are issued by ActiveMARK License Web Services and are downloaded in the background to the end-user's computer without user intervention.
Once a valid license is installed, it is validated each time the product is executed without requiring a persistent Internet connection. License information can be stored remotely allowing users to retrieve licenses as needed after reformats or to activate products on multiple machines based on the specified rules.

Security

Trymedia follows the established principles of prevention, detection, response and healing to ensure maximum efficacy in the security process throughout the lifecycle:

- Prevention: It is possibly the most important aspect of effective content protection. In addition to the sophisticated design of the ActiveMARK security system, extensive anti-hacking, code obfuscation, cryptographic machine-binding, tamper-proofing, 128-bit encryption and intrusion-detection methodologies are deployed to keep content safe as it travels around the network.
- Detection: It is the process of identifying threats against ActiveMARK and the content protected by it. The earlier Trymedia finds a threat, the more time there is to respond to it. For this reason, the Trymedia R&D team is constantly monitoring underground message boards and hacker communities to identify the most pressing dangers.
- Response: Once a real threat has been identified and isolated, a response is prepared. These responses can take the form of an update to the ActiveMARK Technology (i.e. a new release of the ActiveMARK Packaging Tools), or simply a change in trial criteria. In any case, the objective is to balance the customer experience against the potential risk resulting from an attack.

Healing: Once a response has been agreed upon, the product is prepared for the healing cycle. A benefit of the ActiveMARK Technology is the ability to update content with the latest version on the server immediately upon repackaging, although this does not change any files that are being passed from person to person. Once a new version of a file is available, the old version ceases to be available immediately on the Trymedia Network, thus reducing the risk from multiple, older versions being made available.

4.1.10 Apple iTunes

Refer to DE 4.5.1.3 for the description of this DRM system.

4.1.11 CORAL

Refer to DE 4.5.1.3 for the description of this DRM system.

4.1.12 DreaM

Refer to DE 4.5.1.3 for the description of this DRM system.

4.1.13 Marlin

Refer to DE 4.5.1.3 for the description of this DRM system.

4.1.14 OpenSDRM

Refer to DE 4.5.1.3 for the description of this DRM system.

4.2 AXMEDIS users/tools registration, certification, verification and object usage issues

4.2.1 Why AXMEDIS DRM cannot be the same as Windows Media DRM: privacy and security issues

The AXMEDIS project is an European project which gets financial resources from European Community (EC). Therefore it has to present some innovation aspects and take into account the sum of laws adopted by all European Countries and in particular the aspect that concerns privacy.
It has to be considered that cannot be an unique subject that retain all information about users, objects, tools and action performed over objects. In this view the DRM provided by Microsoft with Windows Media DRM is in fault because it concentrates lot of information inside an unique point (the Microsoft Passport). This can be seen as a privacy violation in many European countries according to the related privacy laws.
Moreover Windows Media DRM system retains some protection information (such as license) inside the tool, on the device used to consume the multimedia object. This can be a dangerous issue since it can be cracked in an easy way (it partially happened: see http://nanocrew.net/2005/08/31/reversing-nsc/).

According to what has been written above, the AXMEDIS DRM model cannot be similar to Windows Media (also because AXMEDIS has to provide many innovation aspects to be approved by EC and receive funding). This implies that registration, certification and accounting information has to be distributed among several subjects. This is the way the system has been designed as different entities, including Distributors, AXMEDIS Certifier and Supervisor (AXCS), AXMEDIS Protection Manager and Support (PMS) and so on, each with its own task and responsibility.

In the AXMEDIS current situation each subject retains a subset of user data needed to perform its own task:

- **Distributor**. It retains all data about its own users and the pertinent relation AXUID←→user data.
- **AXCS**. It retains data about users (limited to AXUID) user-id, object-id and the association users←→objects realized as a relation which take into account information about actions performed on an object by a user. It has to be underlined that AXCS doesn't get information about user, but only User-ID (AXUID) and other fewer data: information about users has to be retained by distributors, each for its own users.
- **PMS**: It retains and provides licenses for the whole system.
- **Tool**: It contains certificates which include AXUID: this information is stored in a secure cache.

Moreover the AXMEDIS system has been designed to support multi-channel distribution models and the opportunity of reusing already deployed objects (according to the owner rights and licenses). This implies that in this kind of complex architecture every subject has to provide information to the whole system and has to integrate itself with other subjects providing required information in order to make possible many kinds of business models: each subject has to collaborate with other subjects to make the whole system works.

Since AXCS is a distributed authority,[1] data stored in its archives are available to several distributors, for control and reporting purposes. Therefore objects can be licensed to be used in different distribution channels with no adding work efforts for the distributor. A final user can be managed as the same entity even if he performs object consumption across different devices located in different distribution channels. In this way an object licensed to the user A and distributed by X for PC can be licensed also for mobile adaptation and played by the same user A in mobile distribution channel.
It has to be underlined that not all the licenses are generated on the basis of final users identity, but different business models can be realized by licensing the consumption of contents on a specific tool or a group of tools/devices belonging to a specific domain.

[1] It has to be recalled that AXCS is not an unique entity but there will be a set of AXCSs. In the first prototype only an AXCS will be presented, but in the final step of the project there will be a network of AXCSs.

License information and protection data must be decoupled and stored among different locations. In this way it can be prevented many inconvenient such as registration trials from crackers: if all information would be stored in the same location a brute-force attack trying some random generated ID could be successful (this is not our situation!).

First of all the following main aspects have to be taken into account:

- **Integration**. In order to implement a full integrated system there can be mostly two ways.
    - The first way forces every subject in the system to use the same exact software as the others to perform its own tasks. In this way the previous software used by every subject has to be dismissed and data has to be transformed according to new software specifications. Moreover a software change is a process that have an huge impact in a factory (also in terms of money burning) and can induce unforeseen costs and compatibility problems. This is not the AXMEDIS solution.
    - The second way forces every subject in the system to implement or acquire interfaces in order to perform a full integration with the other subjects in the system. Interfaces are needed to implement a full integration in terms of communications and (especially) data. This is a lower cost and more efficient solution. This is adopted by AXMEDIS.
- **Innovation**. The AXMEDIS project is an European project. It gets financial resources from EC because it has to provide some innovations aspect concerning the multimedia contents distribution. Then AXMEDIS cannot offer solutions that don’t have innovation features (both for distribution and for protection) and cannot implement already seen mechanisms without introducing innovational aspects.
- **MPEG-21 compatibility**. The AXMEDIS system has to be MPEG-21 compliant. This implies that an Event Report mechanism has to be implemented and used. This implies that subjects that own rights over AXMEDIS objects has to use the CAMART interface in order to get accounting information. It has to be underlined that even if a subject (like a distributor) is not directly interested in accounting information (because it has a business model that doesn’t need it) there could be some other subjects that own rights (or that needs accounting information such as collecting societies) over the same objects distributed by that distributor. This means that this kind of distributors doesn’t need an interface with CAMART (in order to get accounting information) but Event Reporting is still needed by the system in order to provide usage information to the other pertinent (and authorized) subjects.
- **DRM interoperability**. Two main conceptualisation has to be underlined: content protection and licenses generation. Objects are protected once for all the users and all the distribution channels. Instead, licenses are generated “ad personam”, i.e. licenses are generated specifically for the recipient that could be a user, a domain and so on. It has to be underlined that the license model is not only mpeg-21 compliant but can be also made a license transcoding to other license models such as ODRL and others.

4.2.2 Basic concepts

Before we continue with the analysis of the current architecture, with the analysis of the distributor requests and with the description of the proposed solutions, it is necessary to introduce some terms used in the following scenarios:

- **AXUID**: it is the AXmedis User IDentifier; this is actually an UUID (a sequence of 32 hexadecimal numbers, see http://www.opengroup.org/onlinepubs/9629399/apdxa.htm for further details on UUID format) preceded by a three-letter prefix (used to distinguish different types of users) and an underscore character. An example of AXUID is BUS_d0719d28-e695-4db7-841c-f078ae7fdfb6.
- **UID Certificate or AXPIC (Axmedis Personal Identity Card)**: it is a sort of AXMEDIS Passport, a certificate in the X.509v3 format containing various data (see the document AXMEDIS-DE3-1-2A-AXFW-Spec-(General-and-Model)-Part-A-v2-0-closed.doc at page 47 for more details) among which the AXUID assigned to the user owning the certificate, his/her public key, an encrypted signature, a validity period, etc.

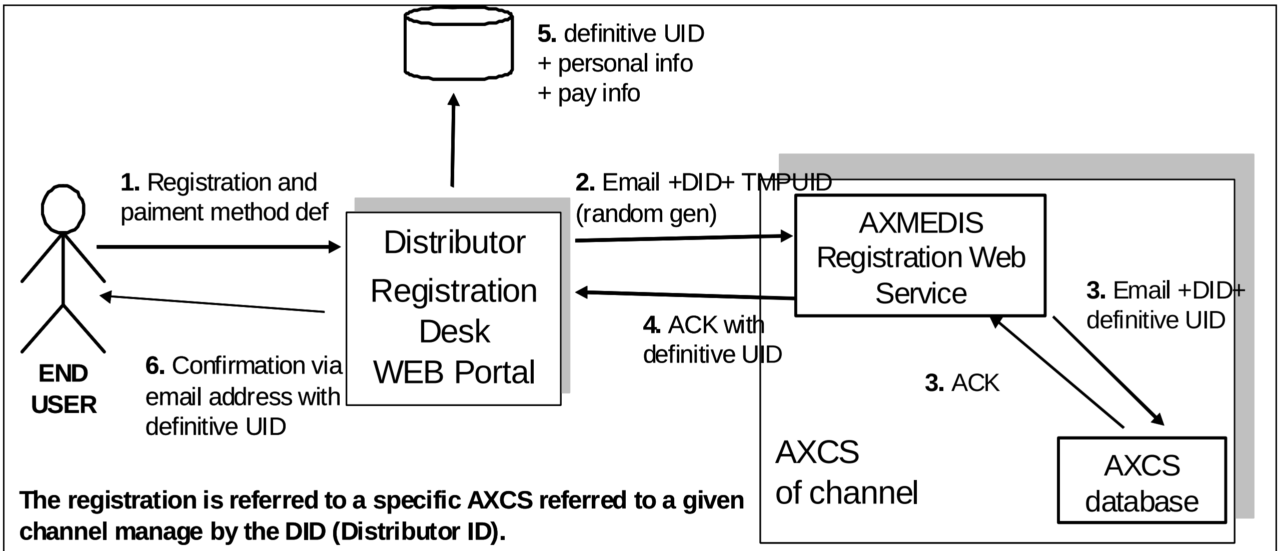
- **AXRTID**: it is the AXmedis Registered Tool IDentifier. It has to be noted that a tool has to be registered in AXMEDIS by the producer in order to receive an AXRTID identifying that particular tool. For instance, if producer DUMMY would produce a tool called DUMMYPlayer to consume AXMEDIS objects, DUMMY would have to register DUMMYPlayer in order to receive an AXRTID used to distinguish the DUMMYPlayer from all the other tools. Please note that an AXRTID is used to distinguish the tool among all those which are AXMEDIS compliant tools, but it does not allow to identify a particular copy of a tool, i.e. the DUMMYPlayer of user A from the one of user B (the AXTID is used for this purpose).
- **AXTID**: it is the AXmedis Tool IDentifier. This id is assigned to a particular copy of a tool already registered in AXMEDIS (therefore it has its AXRTID) upon the certification process. The certification allows to associate that particular copy of the tool to the machine on which it is installed. Continuing with the previous example, the DUMMYPlayer of user A would receive an AXTID different from the one received by DUMMYPlayer of user B, while they would have the same AXRTID.
- **AXDID**: it is the AXmedis Distributor IDentifier; this is actually an UUID and is the equivalent for distributors of the AXUID for users.
- **AXLID**: it is the AXmedis License IDentifier. This is actually an unique identifier used to univocally distinguish licenses produced by the PMS.
- **ActionLog**: it is a structure containing data about actions performed by users on objects for which they have acquired a valid license. Each ActionLog contains data about one action, such as the AXUID, ADID, AXLID, action performed, etc.

In the following scenarios it is often used an abbreviated form of these acronyms (i.e. the acronym without the prefix “AX”) in order to keep short the descriptions of data exchanged among the various entities and thus making the diagrams more readable. So, for instance, it is used the form UID in place of AXUID, or RTID in place of AXRTID and so on.

4.2.3 Current scenarios and architecture

Here is presented a summarize of the actual scenarios representing the current situation.
The diagrams in the following sections concerns registration, verification, content consumption and license production.

4.2.3.1 End User registration in a distribution channel scenario



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

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

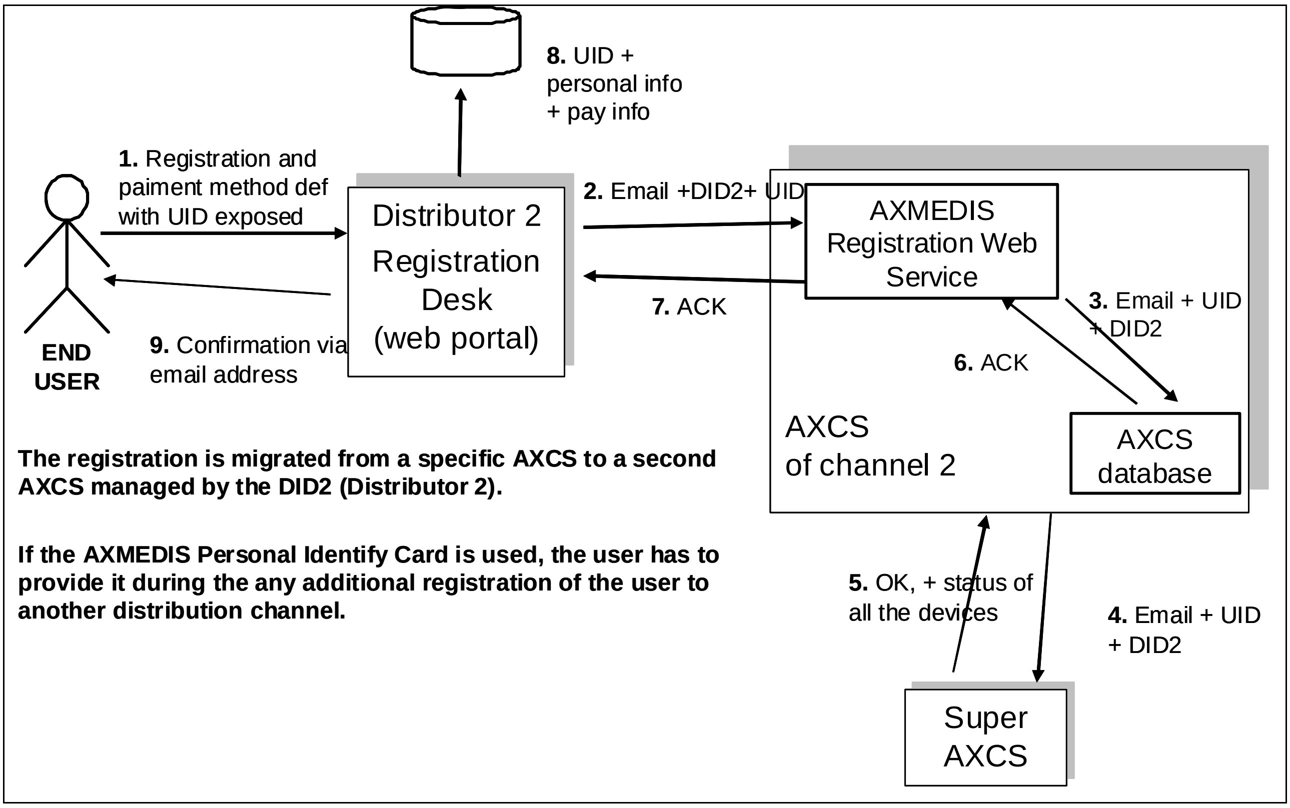
Scenario "Pros":

- The distributor plays the typical role of authenticating user identity in different ways (pop authentication, certificates, etc.). After the user has been identified, this warranty is transferred in the AXMEDIS system.
- The AXUID is always in the hand of distributor and the latter can use the AXUID in order to generate licenses for its user, e.g. after purchasing in (pre)pay-per-view or in subscription.

Scenario "Cons":

- The distributor has to implement a client to use AXCS Registration Web Service
- The distributor has the duty of registering a user to AXMEDIS if it wants to sell AXMEDIS contents to that user
- The distributor has to store and manage additional information with respect to user data

4.2.3.2 End User registration in a different distribution channel scenario


8. UID + personal info + pay info
1. Registration and paiment method def with UID exposed
Distributor 2 Registration Desk (web portal)
2. Email +DID2+ UID
AXMEDIS Registration Web Service
7. ACK
3. Email + UID + DID2
6. ACK
END USER
9. Confirmation via email address
AXCS of channel 2
AXCS database
5. OK, + status of all the devices
4. Email + UID + DID2
Super AXCS

The registration is migrated from a specific AXCS to a second AXCS managed by the DID2 (Distributor 2).

If the AXMEDIS Personal Identify Card is used, the user has to provide it during the any additional registration of the user to another distribution channel.

Scenario “Pros”:

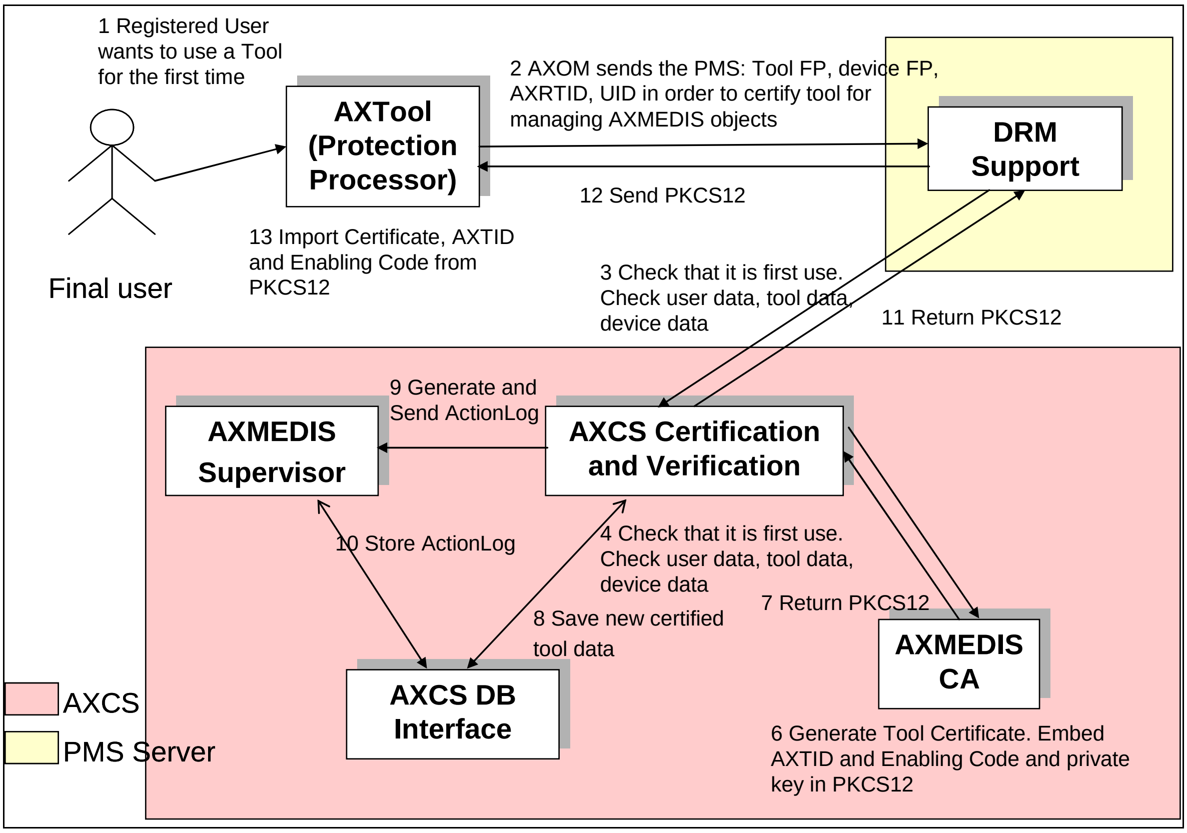
- The distributor has no obligation of registering a new user in AXMEDIS, it can sell objects to already registered user
- The provided AXUID is checked against information stored in some AXCS

Scenario “Cons”:

- The distributor has to implement a client to use AXCS Registration Web Service
- The distributor has to store and manage additional information with respect to user data
- The distributor has to setup some https web pages in order to create a secure connection on the basis of a user UID certificate and to get the AXUID information; in this web pages the distributor has to manage different situations (user already registered in AXMEDIS or not) since distributor doesn’t know this information. In this case only the user can provide it and the distributor needs to know that in order to decide whether to establish the secure connection with the user or not.

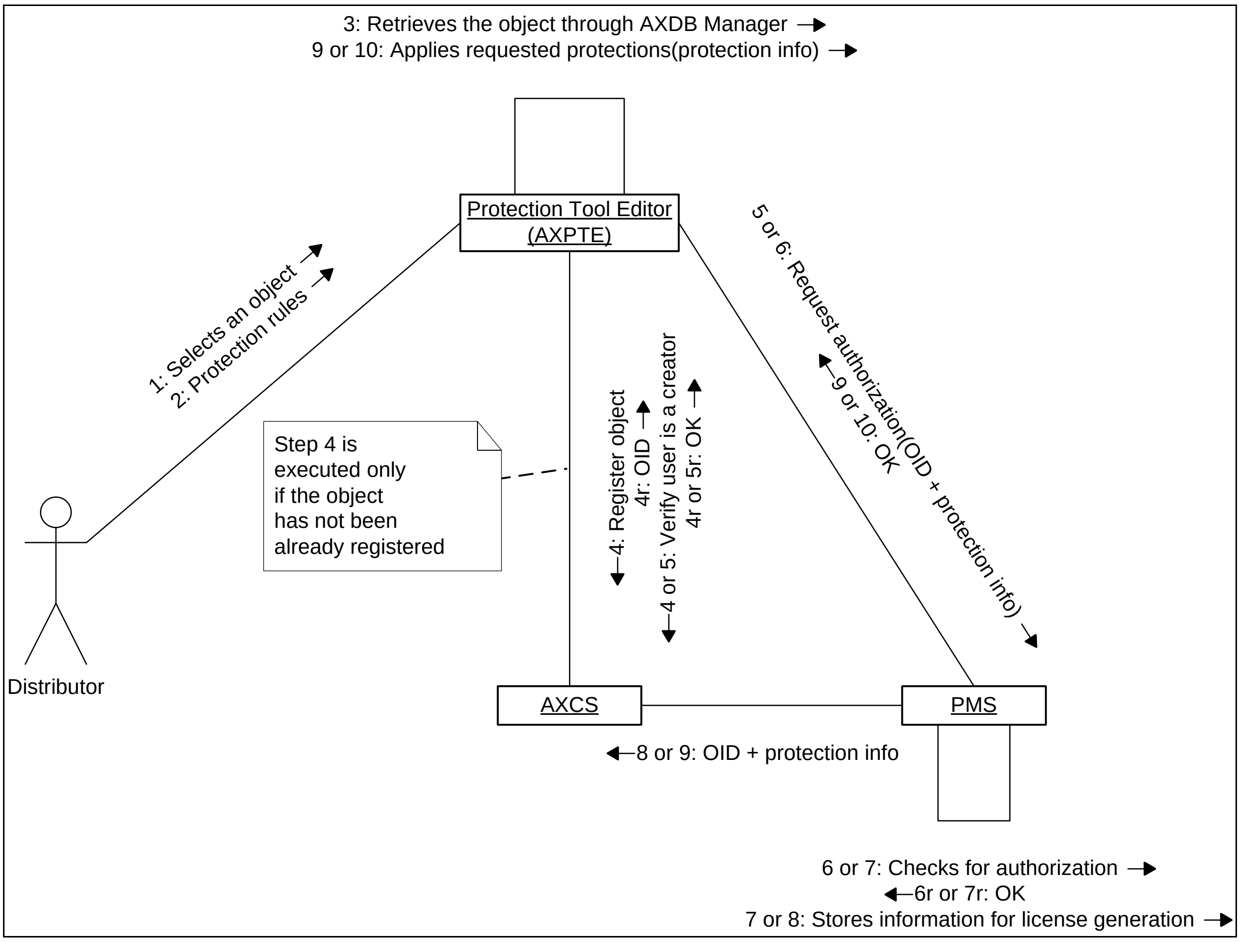
Please note that this mechanism makes it possible to register an user in AXMEDIS only once and this means an user is associated with only one AXUID in the whole system. So, there is no need to repeat the registration process even if the user wants to do a transaction with a distributor other than the one that registered him in AXMEDIS.

4.2.3.3 Certification of Tool by User scenario


1 Registered User wants to use a Tool for the first time
Final user
AXTool (Protection Processor)
2 AXOM sends the PMS: Tool FP, device FP, AXRTID, UID in order to certify tool for managing AXMEDIS objects
DRM Support
12 Send PKCS12
13 Import Certificate, AXTID and Enabling Code from PKCS12
3 Check that it is first use. Check user data, tool data, device data
11 Return PKCS12
9 Generate and Send ActionLog
AXMEDIS Supervisor
AXCS Certification and Verification
10 Store ActionLog
4 Check that it is first use. Check user data, tool data, device data
7 Return PKCS12
8 Save new certified tool data
AXMEDIS CA
AXCS DB Interface
6 Generate Tool Certificate. Embed AXTID and Enabling Code and private key in PKCS12
AXCS
PMS Server

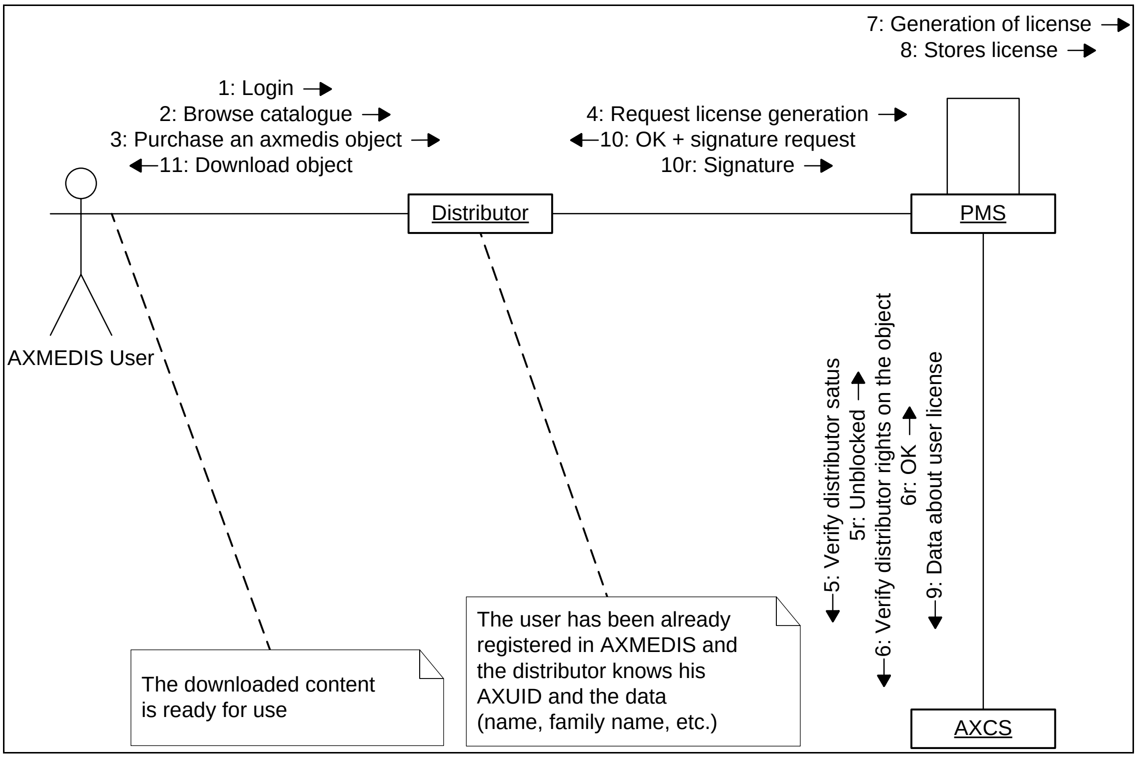
This scenario is reported only for information about the AXMEDIS security mechanisms. The distributors have not to deal with these aspects, but they only benefit the advantages.

4.2.3.4 Objects protection


3: Retrieves the object through AXDB Manager →
9 or 10: Applies requested protections(protection info) →
Protection Tool Editor (AXPTE)
1: Selects an object →
2: Protection rules →
5 or 6: Request authorization(OID + protection info) →
←9 or 10: OK
←4: Register object
4r: OID →
←4 or 5: Verify user is a creator
4r or 5r: OK →
Step 4 is executed only if the object has not been already registered
Distributor
AXCS
PMS
←8 or 9: OID + protection info
6 or 7: Checks for authorization →
←6r or 7r: OK
7 or 8: Stores information for license generation →

This scenario is reported only for information about the AXMEDIS protection mechanisms. Even if distributors are involved, the object protection mechanism is not treated in this document.

4.2.3.5 License generation


7: Generation of license
8: Stores license
1: Login
2: Browse catalogue
3: Purchase an axmedis object
11: Download object
4: Request license generation
10: OK + signature request
10r: Signature
Distributor
PMS
AXMEDIS User
5: Verify distributor satus
5r: Unblocked
6: Verify distributor rights on the object
6r: OK
9: Data about user license
The downloaded content is ready for use
The user has been already registered in AXMEDIS and the distributor knows his AXUID and the data (name, family name, etc.)
AXCS

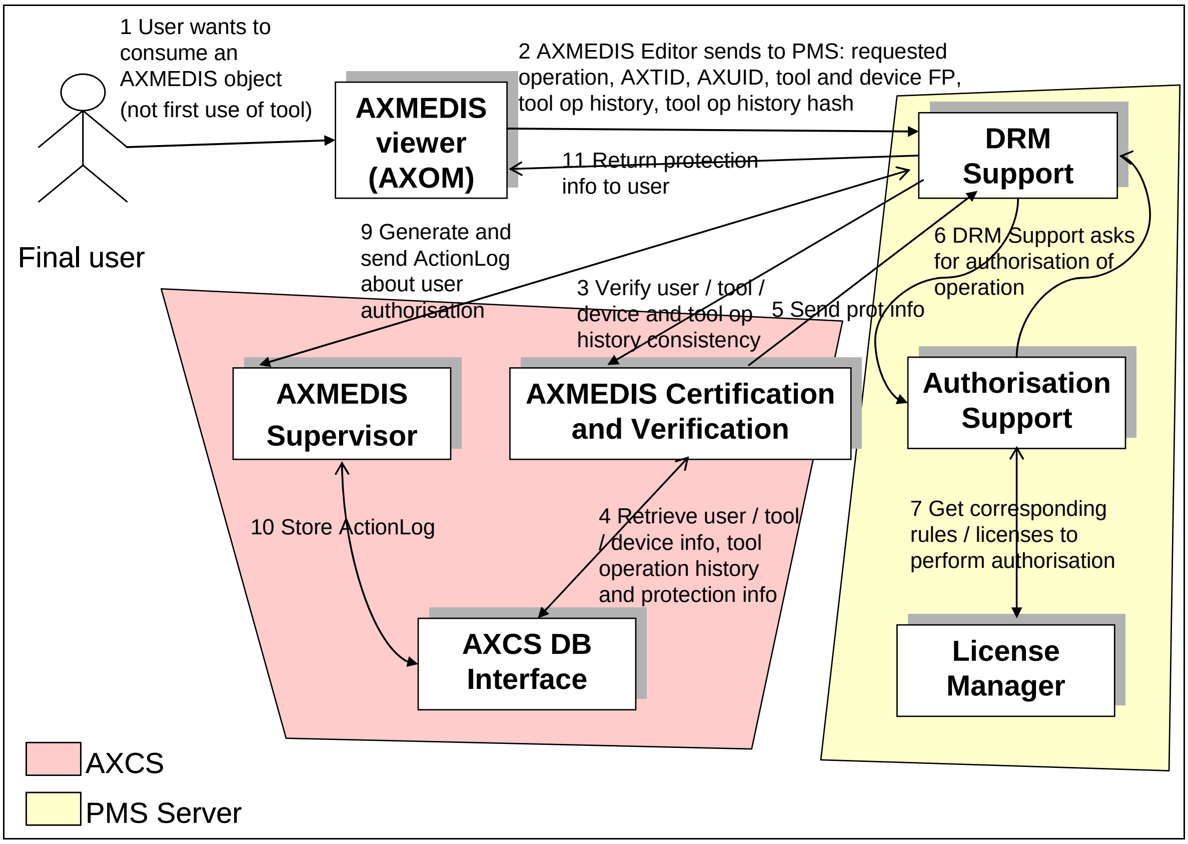
The License generation scenario “Pros”:

- The distributor can put in his catalogue AXMEDIS and non AXMEDIS objects
- The distributor is responsible only for issuing license generation providing the needed data, such as AXUID, AXTID, AXOID, Domain, etc. None of these data are mandatory and different kinds of license can be generated on the basis of this information.
- The distributor has not to deal DRM enforcement technology, since it is provided by the AXMEDIS system
- The distributor can issue a license generation for an object only if it owns the distribution rights of that object
- The distributor, being the real issuer of the license, is requested to sign it

Scenario “Cons”:

- The distributor has to deal with PMS License Generator Web Service
- The distributor has to store and manage relevant information for license production. The more complex are the licenses, the more large amount of data is needed.

4.2.3.6 Successful consumption of a protected AXMEDIS object


1 User wants to consume an AXMEDIS object
(not first use of tool)
Final user
AXMEDIS viewer (AXOM)
2 AXMEDIS Editor sends to PMS: requested operation, AXTID, AXUID, tool and device FP, tool op history, tool op history hash
DRM Support
11 Return protection info to user
9 Generate and send ActionLog about user authorisation
3 Verify user / tool / device and tool op history consistency
5 Send prot info
6 DRM Support asks for authorisation of operation
AXMEDIS Supervisor
AXMEDIS Certification and Verification
Authorisation Support
10 Store ActionLog
4 Retrieve user / tool / device info, tool operation history and protection info
7 Get corresponding rules / licenses to perform authorisation
AXCS DB Interface
License Manager
AXCS
PMS Server

This scenario is reported only for information about the AXMEDIS usage authorization mechanisms. If the license has been already issued, the distributor is not interested in this process. It can anyway be aware of content consumption using reporting mechanisms.

4.2.4 What is wrong in alternative scenarios

During the last meeting, distributors and technical designers tried to modify actual scenarios so as fulfil distributor requests. In the technical meeting there was proposed some architectural modification expressed by the following scenarios. After a careful analysis it was clear that some aspects had not taken into account.

Here is reported one of the proposed scenarios with the corresponding bad aspects that make them infeasible.

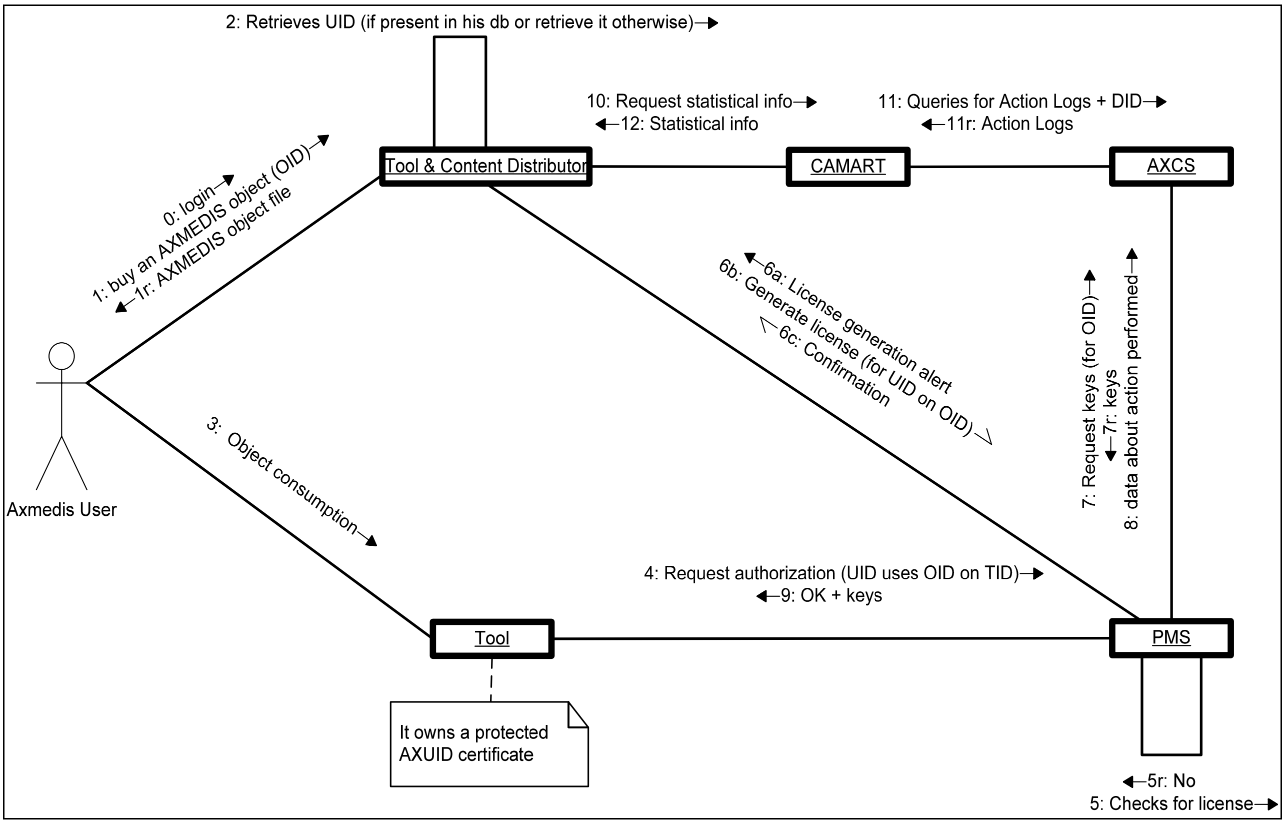

2: Retrieves UID (if present in his db or retrieve it otherwise)→
10: Request statistical info→
←12: Statistical info
11: Queries for Action Logs + DID→
←11r: Action Logs
Tool & Content Distributor
CAMART
AXCS
0: login→
1: buy an AXMEDIS object (OID)→
←1r: AXMEDIS object file
←6a: License generation alert
6b: Generate license (for UID on OID)→
←6c: Confirmation
7: Request keys (for OID)→
←7r: keys
8: data about action performed→
Axmedis User
3: Object consumption→
4: Request authorization (UID uses OID on TID)→
←9: OK + keys
Tool
PMS
It owns a protected AXUID certificate
←5r: No
5: Checks for license→

Figure 20 – User uses content

The major bad aspect of the above scenario is when PMS contacts Distributor for License generation alert, the User has to sign the license: in this scenario the user cannot sign the license.

This is an important requirement imposed by Distributors (also for legal issues).

4.2.5 What can be done to fulfil distributor needs

During the last meeting, distributors expressed several needs that can be mostly summarized as follows:

1. All distributors don't like to register every user in the AXMEDIS system: they would like to register only users interested in AXMEDIS objects consumption

2. Some distributors don't like to register their own users at all and doesn't want to store AXUID←→User association in their own factory database. They don't want to develop or acquire a software interface to the AXMEDIS Registration Web Service

3. Some distributors would like to register users only when it is needed (first time AXMEDIS object usage)

4. Distributors would like to generate on-demand licenses

   (Note that trough the following paragraph of this section the above numbers will be used to refer the associated needs).

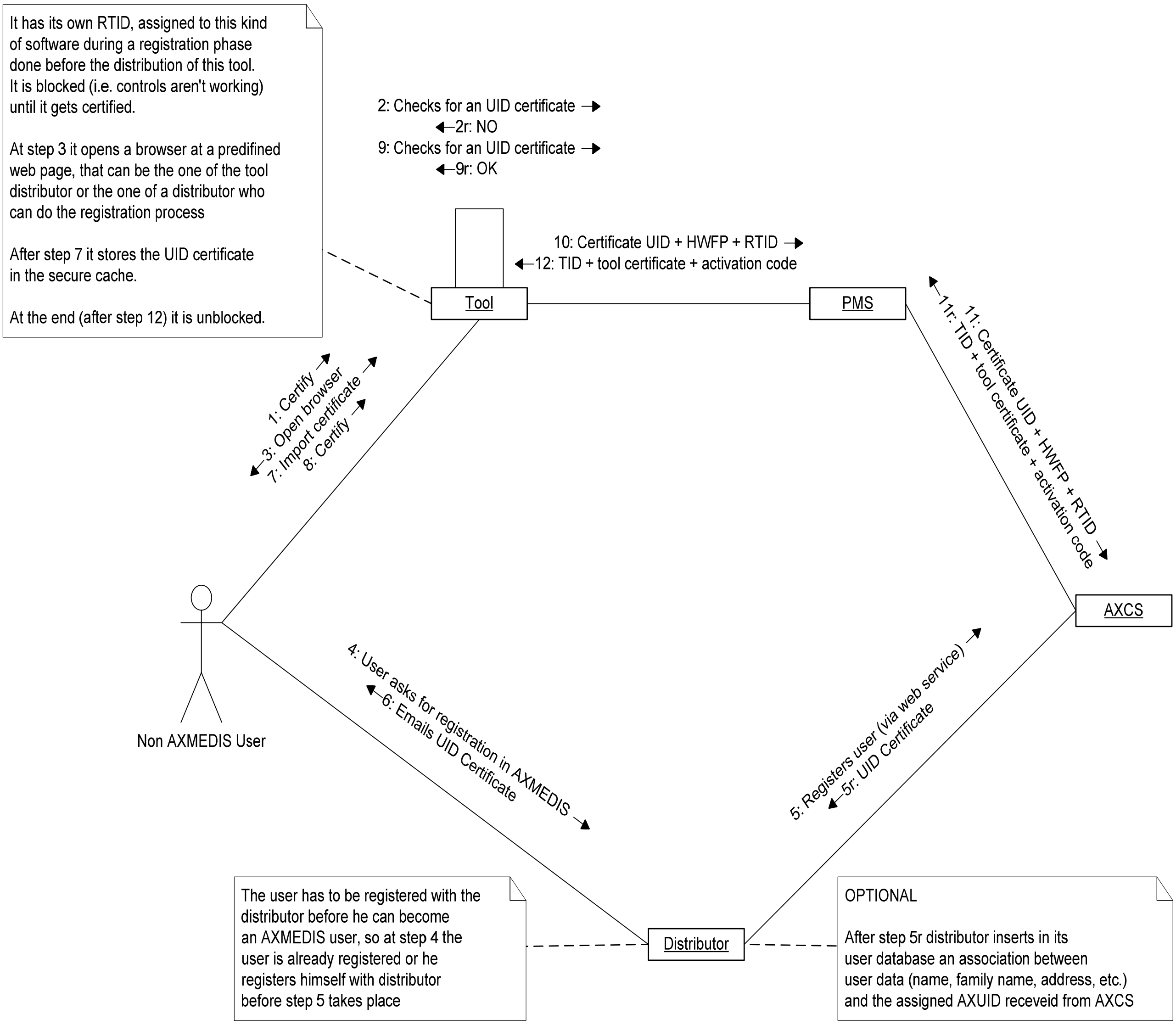
To satisfy request number 2, a new mechanism for registering users has been introduced, which lets users register in AXMEDIS themselves (see the following section 4.2.5.2 “Self user registration and tool certification scenario”). However, *request number 2 has an important implication: since some distributors don't want to store AXUID←→User association in their factory, a mechanism to retrieve user identity starting from his/her AXUID is needed, to be able to determine the user identity in case of trouble*s. How this can be achieved will be detailed in the following section 4.2.5.3 “Object consumption and on–demand license generation scenario” (and in particular in its sub-section “Important note”).

According to the reasons expressed in the previous sections, distributor requests can be almost satisfied. Let's consider the following scenarios represented as collaboration diagrams (please note that these scenarios are not intended to substitute those already presented in the section 4.2.3 “Current scenarios and architecture”, rather they are proposed as useful additions).

4.2.5.1 User registration through a distributor and tool certification scenario

Here are proposed two different versions, Version A and Version B.

Version A


2: Checks for an UID certificate →
←2r: NO
9: Checks for an UID certificate →
←9r: OK
10: Certificate UID + HWFP + RTID →
←12: TID + tool certificate + activation code
Tool
PMS
11: Certificate UID + HWFP + RTID →
←11r: TID + tool certificate + activation code
AXCS
1: Certify →
←3: Open browser
7: Import certificate →
8: Certify →
Non AXMEDIS User
4: User asks for registration in AXMEDIS →
←6: Emails UID Certificate
5: Registers user (via web service) →
←5r: UID Certificate
Distributor

It has its own RTID, assigned to this kind of software during a registration phase done before the distribution of this tool. It is blocked (i.e. controls aren't working) until it gets certified.

At step 3 it opens a browser at a predifined web page, that can be the one of the tool distributor or the one of a distributor who can do the registration process

After step 7 it stores the UID certificate in the secure cache.

At the end (after step 12) it is unblocked.

The user has to be registered with the distributor before he can become an AXMEDIS user, so at step 4 the user is already registered or he registers himself with distributor before step 5 takes place

OPTIONAL

After step 5r distributor inserts in its user database an association between user data (name, family name, address, etc.) and the assigned AXUID receveid from AXCS

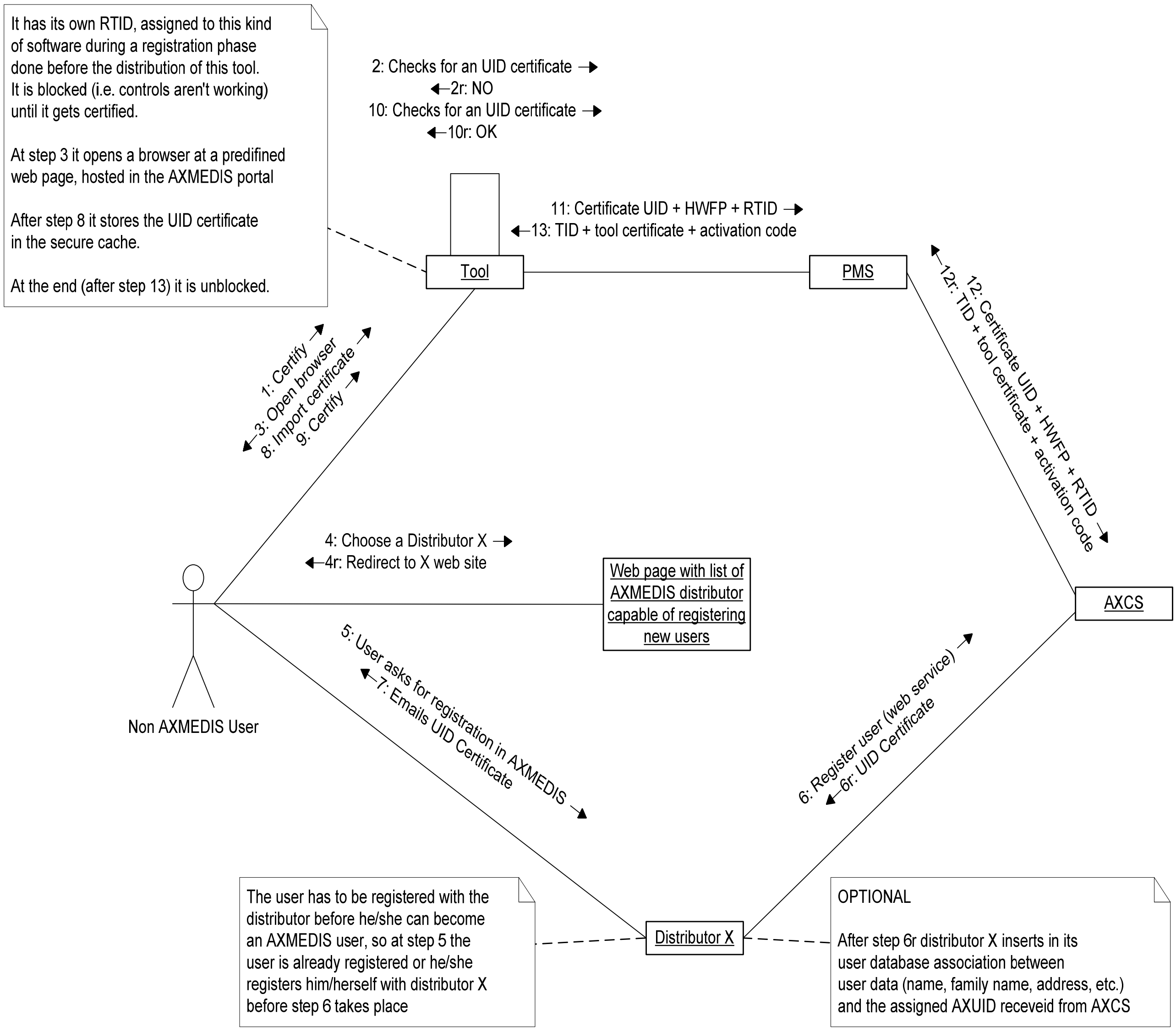
Key considerations about Version A scenario are:

A. It is assumed that the tool is capable of redirecting the user to a predefined web page and that the URL is stored directly inside the tool

B. The URL musts identify the web site of a distributor who is capable of registering new users in AXMEDIS: therefore this distributor has to interact with AXCS

C. The distributor can avoid to store the User←→AXUID association, because this information can be retrieved when needed directly from the user (see the following section 4.2.5.3 “Object consumption and on–demand license generation scenario” for further details)

Version B


It has its own RTID, assigned to this kind of software during a registration phase done before the distribution of this tool.
It is blocked (i.e. controls aren't working) until it gets certified.
At step 3 it opens a browser at a predifined web page, hosted in the AXMEDIS portal
After step 8 it stores the UID certificate in the secure cache.
At the end (after step 13) it is unblocked.
2: Checks for an UID certificate →
←2r: NO
10: Checks for an UID certificate →
←10r: OK
Tool
11: Certificate UID + HWFP + RTID →
←13: TID + tool certificate + activation code
PMS
12: Certificate UID + HWFP + RTID →
←12r: TID + tool certificate + activation code
AXCS
1: Certify →
←3: Open browser
8: Import certificate →
9: Certify →
4: Choose a Distributor X →
←4r: Redirect to X web site
Web page with list of AXMEDIS distributor capable of registering new users
Non AXMEDIS User
5: User asks for registration in AXMEDIS →
←7: Emails UID Certificate
6: Register user (web service) →
←6r: UID Certificate
Distributor X
The user has to be registered with the distributor before he/she can become an AXMEDIS user, so at step 5 the user is already registered or he/she registers him/herself with distributor X before step 6 takes place
OPTIONAL
After step 6r distributor X inserts in its user database association between user data (name, family name, address, etc.) and the assigned AXUID receveid from AXCS

Key considerations about Version B scenario are:

A. It is assumed that the tool is capable of redirecting the user to a predefined web page and that the URL is stored directly inside the tool

B. The web page is dynamically generated to reflect the most up-to-date situation. The distributor list is created on the basis of information retrieved from AXCS

C. The distributor can avoid to store the User←→AXUID association, because this information can be retrieved when needed directly from the user (see the following section 4.2.5.3 “Object consumption and on–demand license generation scenario” for further details)

Scenarios satisfied needs

It is worth to point out that the two proposed versions A and B of this scenario don’t exclude out each other. It is possible to have both of them with a solution like this: version B takes place if the tool considered is the “original” version, i.e. the one developed by the Consortium not yet tailored by any distributor (this can happen if the user has downloaded the tool directly from the AXMEDIS portal or he has received it in any other way, such as CD-ROM, etc.). If the tool has been tailored by a distributor, version A of the scenario takes place.

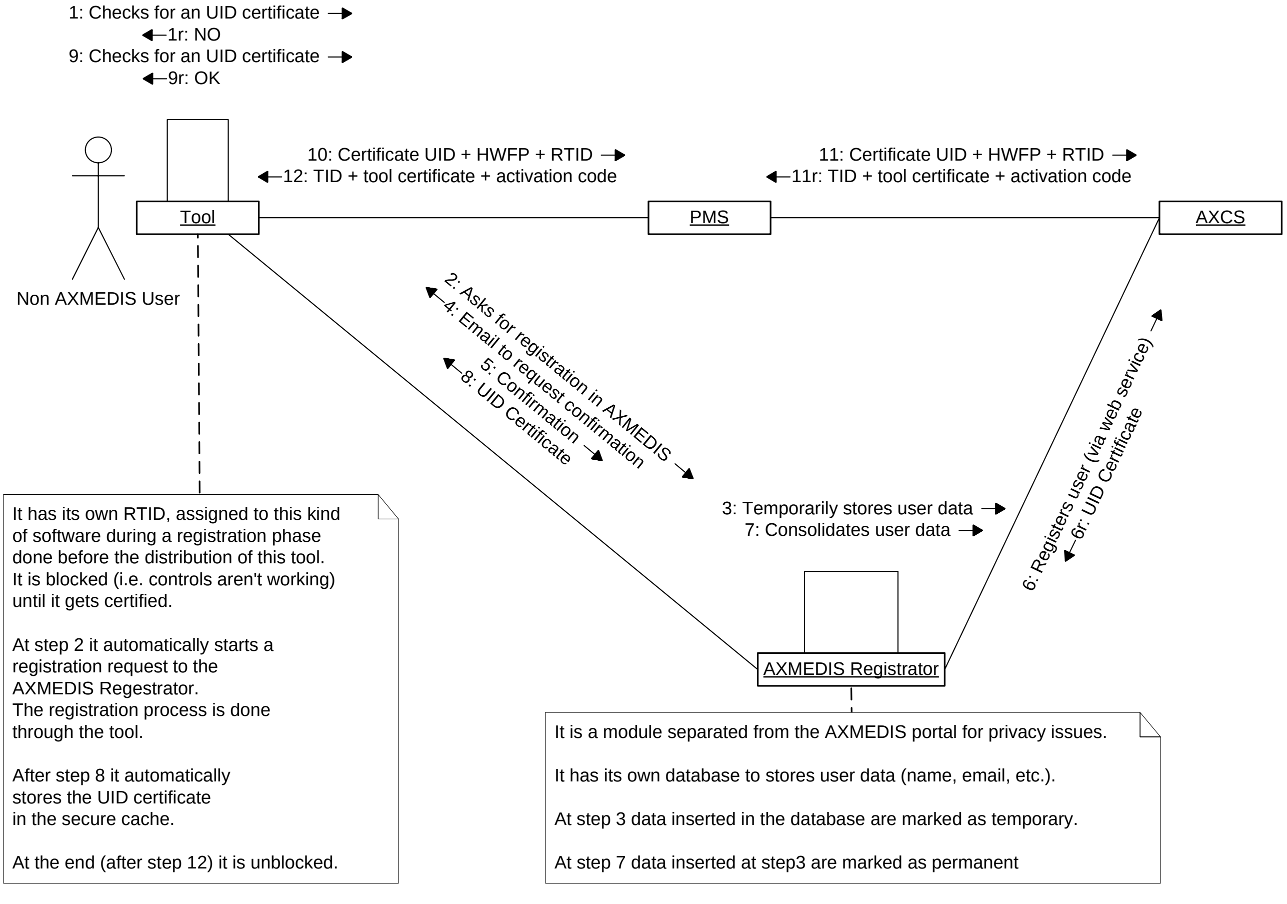
This scenario satisfies the following distributor requests:

1. because the user registration process is issued by the certification of an AXMEDIS compliant tool (so certainly the user is interested in using AXMEDIS objects)

2. because the user registration is delegated to a proper distributor either with whom one has an agreement (version A) or chosen by the user (version B)

3. because the user registration process takes place only when the user wants to use the tool and not before

4.2.5.2 Self user registration and tool certification scenario


1: Checks for an UID certificate →
←1r: NO
9: Checks for an UID certificate →
←9r: OK
10: Certificate UID + HWFP + RTID →
←12: TID + tool certificate + activation code
11: Certificate UID + HWFP + RTID →
←11r: TID + tool certificate + activation code
Non AXMEDIS User
Tool
PMS
AXCS
2: Asks for registration in AXMEDIS →
←4: Email to request confirmation
5: Confirmation →
←8: UID Certificate
3: Temporarily stores user data →
7: Consolidates user data →
6: Registers user (via web service) →
←6r: UID Certificate
AXMEDIS Registrator

It has its own RTID, assigned to this kind of software during a registration phase done before the distribution of this tool.
It is blocked (i.e. controls aren't working) until it gets certified.

At step 2 it automatically starts a registration request to the AXMEDIS Regestrator.
The registration process is done through the tool.

After step 8 it automatically stores the UID certificate in the secure cache.

At the end (after step 12) it is unblocked.

It is a module separated from the AXMEDIS portal for privacy issues.

It has its own database to stores user data (name, email, etc.).

At step 3 data inserted in the database are marked as temporary.

At step 7 data inserted at step3 are marked as permanent

Considerations about this scenario are:

A. In this scenario the AXUID generated is a valid (and therefore unique) AXUID, so it can be used normally by the user during any successive transaction (which obviously will be with a distributor, who can retrieve the AXUID as depicted in the scenario described in the following section 4.2.5.3 “Object consumption and on–demand license generation scenario”).

B. The tool contains information needed to contact both the AXMEDIS Registrator and a PMS. Please note that while the AXMEDIS Registrator is only an entity which simulates a real distributor in the registration process, the PMS depicted is a real PMS (one of the PMSs in the AXMEDIS system).

C. The communication between tool and AXMEDIS Registrator can be realized in various ways, for instance via Web Service (server on AXMEDIS Registrator and client on the tool), etc.

D. Nothing prevents the user from providing false personal data, anyway this is not an issue as this data will not be used in successive transactions. In fact the user will have to provide his/her personal data to a distributor at the moment of a successive transaction, for instance to purchase a license (as it can be seen at step 8 in the scenario described in the following section 4.2.5.3 “Object consumption and on–demand license generation scenario” when the user logs on the distributor site, so he/she has to be registered to the distributor) and therefore the distributor will have the chance to check the validity of user data against some authority (e.g. using the user credit card number). As at that time he will provide his/her AXUID to the distributor (needed for commanding license generation on PMS, see again the same scenario in the following section 4.2.5.3), the license will contain the AXUID, letting the user to be identified univocally.

E. The email sent at step 4 contains a code to be used to confirm the registration. In this automatically generated requests can be ignored. Only after the confirmation the user is actually registered on AXCS. If the registration is not confirmed in a predefined time, temporary data registered a t step 3 are cancelled and the registration is nullified.

F. The registration process is carried out directly through the tool, not through some external component (such a browser). This is to implement some controls such as a limit to the number of time an user can do the registration process, in order to prevent attacks to the AXMEDIS Registrator.

G. The self registration process can be done only a fixed number of time. In each attempt the tool checks the number of UID Certificates already installed and lets the process occur only if the limit has not yet been reached. At the same time, also the AXMEDIS Registrator perform this check: for each request it verifies how many user registrations that tool has already performed. If this number reaches the predefined limit, AXMEDIS Registrator does not allow a new registration.

This scenario satisfies the following distributor requests:

1. because the user is certainly interested in AXMEDIS object consumption, since he’s using the tool

2. because no distributor is involved in the registration process

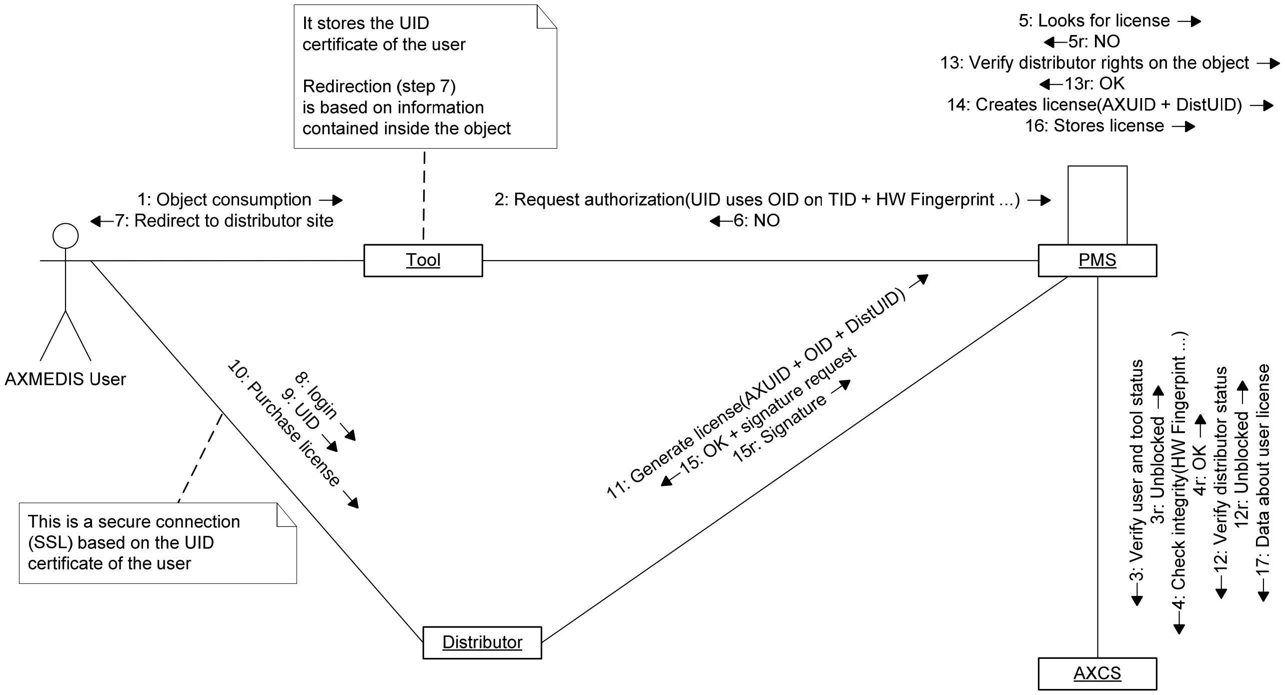
3. because the user registration process is done only at the first use of the tool and not before

4.2.5.3 Object consumption and on–demand license generation scenario

In this scenario we introduce a new term which is used only for convenience: DistUID.
DistUID is the user ID assigned by a distributor to a user who registers himself/herself to it (so it is a distributor specific way to identify the user).

Here are proposed two different versions, Version A and Version B.

Version A: single user license


It stores the UID certificate of the user

Redirection (step 7) is based on information contained inside the object
5: Looks for license →
←5r: NO
13: Verify distributor rights on the object →
←13r: OK
14: Creates license(AXUID + DistUID) →
16: Stores license →
1: Object consumption →
←7: Redirect to distributor site
2: Request authorization(UID uses OID on TID + HW Fingerprint ...) →
←6: NO
Tool
PMS
AXMEDIS User
8: login →
9: UID →
10: Purchase license →
11: Generate license(AXUID + OID + DistUID) →
←15: OK + signature request
15r: Signature →
This is a secure connection (SSL) based on the UID certificate of the user
Distributor
←3: Verify user and tool status
3r: Unblocked →
←4: Check integrity(HW Fingerpint ...)
4r: OK →
←12: Verify distributor status
12r: Unblocked →
←17: Data about user license
AXCS

To make it work, all the following initial condition must be satisfied:

A. User has to be an AXMEDIS User (already registered over the system)

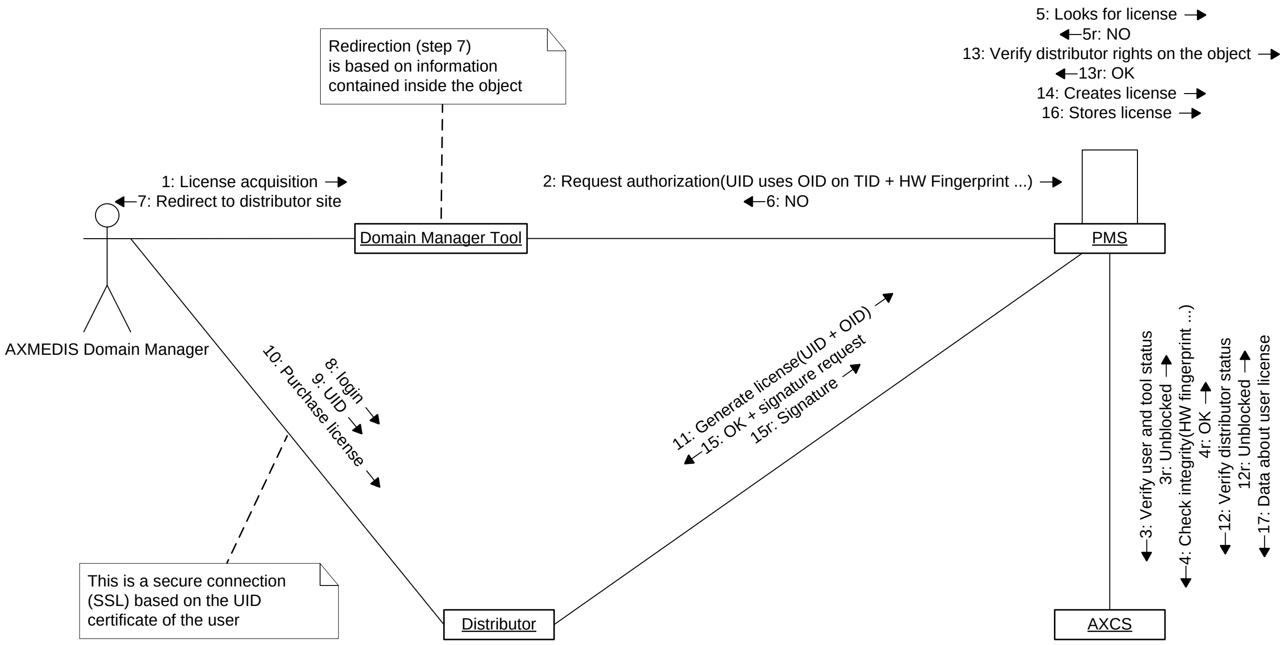
B. Tool has to be an AXMEDIS Tool (already certified over the system)

C. User got an object from a somewhat distribution channel (CD, DVD, internet, snail mail, and so on)

D. Object has to be tailored by the distributor including its own metadata such as particular object license acquisition web page (or other communication channel)

E. Tool has to be designed including license acquisition automatism (redirect to distributor object license acquisition web page or other communication channel)

Version B: user domain license


5: Looks for license
5r: NO
13: Verify distributor rights on the object
13r: OK
14: Creates license
16: Stores license
Redirection (step 7)
is based on information
contained inside the object
1: License acquisition
7: Redirect to distributor site
2: Request authorization(UID uses OID on TID + HW Fingerprint ...)
6: NO
Domain Manager Tool
PMS
AXMEDIS Domain Manager
8: login
9: UID
10: Purchase license
11: Generate license(UID + OID)
15: OK + signature request
15r: Signature
3: Verify user and tool status
3r: Unblocked
4: Check integrity(HW fingerprint ...)
4r: OK
12: Verify distributor status
12r: Unblocked
17: Data about user license
This is a secure connection
(SSL) based on the UID
certificate of the user
Distributor
AXCS

To make it work, all the following initial condition must be satisfied:

A. User has to be an AXMEDIS User (already registered over the system)

B. Tool has to be an AXMEDIS Tool (already certified over the system)

C. User got an object from a somewhat distribution channel (CD, DVD, internet, snail mail, and so on)

D. Object has to be tailored by the distributor including its own metadata such as particular object license acquisition web page (or other communication channel)

E. Tool has to be designed including license acquisition automatism (redirect to distributor object license acquisition web page or other communication channel)

Important note

Please note that it is necessary to store in the generated license the distributor specific ID assigned to the user (DistUID). As remarked at the beginning of this section 4.2.5, distributor request number 2 requires a mechanism to retrieve user identity starting from the AXUID, useful in case of troubles. Let’s suppose that a trouble with a user is discovered on the basis of some ActionLog. The ActionLog contains the AXUID of the user who performed the action, so the user can be univocally identified: anyway his/her identity is still unknown. This information can be simply retrieved if the license (univocally identified by the AXLID) stores the DistUID. In fact an ActionLog contains also the AXLID of the license which allowed the user to perform an action, so it is possible to retrieve that license from the correspondent PMS. Now, on the basis of the DistUID, it is possible to alert the proper distributor (an ActionLog contains also the pertinent AXDID), who knows the real identity of the user and therefore can take countermeasures.

This mechanism is fundamental to support the self user registration process (the scenario described in the section 4.2.5.2), in which the user is not granted by anyone in the AXMEDIS system.

Scenarios satisfied needs

Please note that this scenario is capable to support various business models. For instance, it is possible to make the user pay at the moment of the generation request (step 10) or to generate the license and then billing the user on a time based schema (in this case an interface with CAMART is requested for accounting purposes), etc.

This scenarios satisfies the following distributor requests:

2. because in this way distributors have not to deal with AXCS (so no interface to it is needed) and can avoid to store AXUID←→User association since they recover it directly from the user

4. because the license is generated on-demand

4.3 Tasks realised

As pointed out in the previous sections, full study of the DRM state of the art (related to content distribution via Internet) has been done and additional registration, certification, verification and object usage issues have been explored. In this sense, some tools have been implemented for the description of DRM systems that can be integrated with the current distribution systems.

The integration of these tools with the current platform has been addressed and all the described functionalities are currently integrated in the AXMEDIS framework, including end user registration, certification of tools, objects protection, license generation and consumption protected AXMEDIS objects.

4.4 Prototype description

Integration with the implemented tools has been done by means of web services, as described in the specification documents. Tools have been implemented using C++ and Java languages.

4.5 Integration of DRM models for Internet distribution

In previous sections, the scenarios for the integration between the different AXMEDIS tools and servers have been shown. In particular, it has been described when the Protection Management Support Server is contacted to make use of the licensing functionalities and also when the tools (through PMS Server) contact AXMEDIS Certifier and Supervisor in order to be certified.

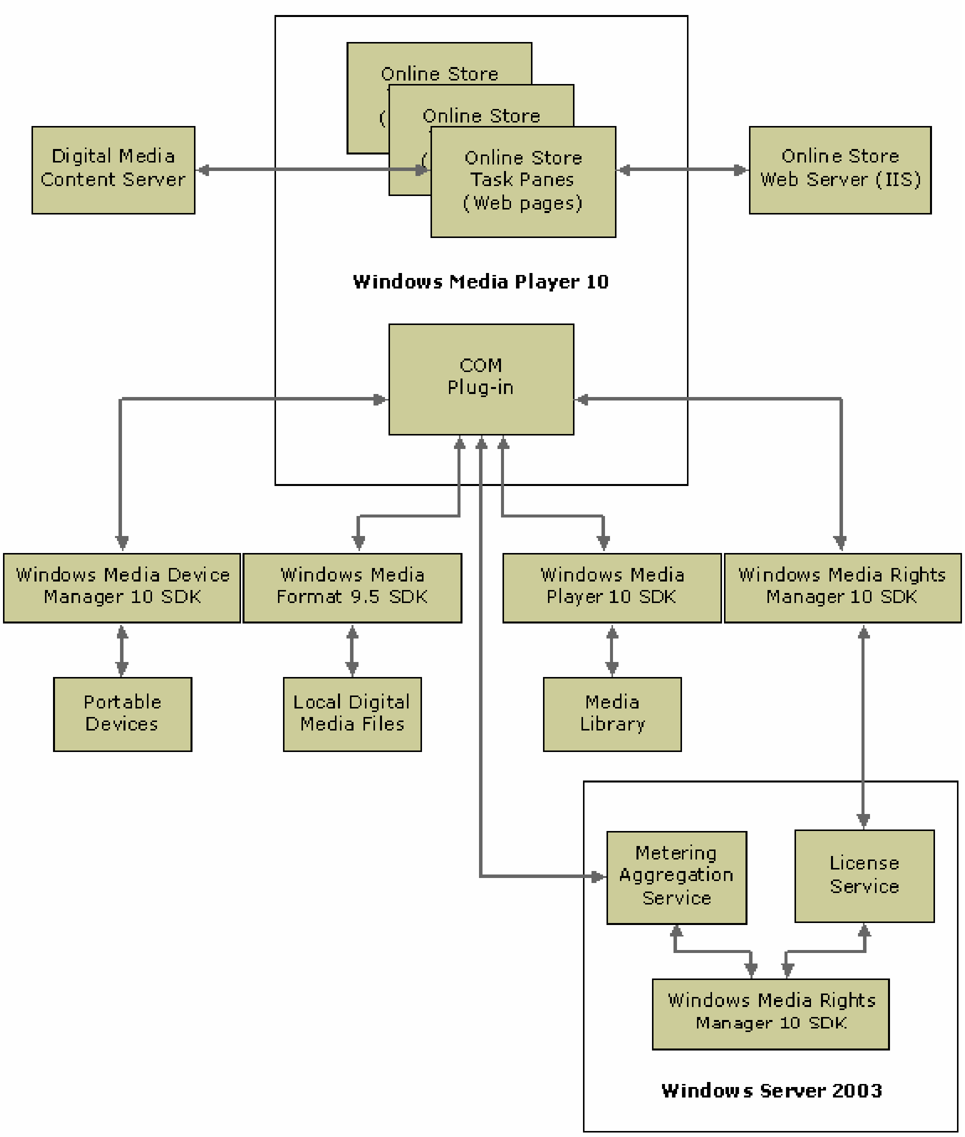
The design of the servers, which use Web Services technologies and provide secure connections to the users, make them very suitable for the Internet distribution. Security is provided by means of Client and Server certificates, that is, the Client needs and AXMEDIS certificate to be authenticated in front of the server. These certificates allow the establishment of an HTTPS connection (Secure HTTP, running over Secure Sockets Layer (SSL)).

The integration of the DRM models implemented in AXMEDIS for Internet distribution is direct, as the servers have to be connected to the Internet to provide the needed functionality to the different clients. From the client side, several modules are provided in order to facilitate communication with the servers. These modules are Protection Manager Support client and Protection Processor. These modules have to be integrated in the client side tools in order to be able to call the functionalities provided by the servers. Moreover, these modules keep track locally of some relevant information for allowing the user to work off-line, like the secure storage of licenses, protection information or action logs. Some of this information is only stored until a new connection to the servers is performed. At this moment, the information is sent and the server information is synchronised with the information provided by the client tool.

4.5.1 Integration of Windows Media

In order to tackle the integration of AXMEDIS DRM and another commercial tools that also use his own DRM we have chosen Microsoft Windows Media Rights Manager and Windows Media Player technologies.

Windows Media technologies include several modules that allow the management of contents, rights and some other distribution aspects. Next picture shows a possible architecture based on Windows Media Technologies for an online music store.



The integration of this kind of architecture can be taken from two different aspects. One is the integration of Windows Media Player for being able to play AXMEDIS objects and the other is the integration of a Windows Media DRM Server with AXMEDIS Protection Manager Support Server.

To integrate Windows Media Player with AXMEDIS objects we consider that a Windows Media plug-in has to be developed. This plug-in should contain at least an instance of the PMS Client in order to be able to authorise actions over AXMEDIS objects and to be able to decrypt the contents. The real problem is to ensure that a decrypted content must not be accessible for a user, it should be only accessible by the Windows Media Player. With some kind of content such as video or audio, one option could be to generate an internal streaming server and decrypt the content in real time. The AXMEDIS plug-in has also to access his own Secure Cache (like every PMS Client) to store keys, licenses and other relevant data.

The other possible integration is between Windows Media DRM Server and AXMEDIS Protection Manager Support Server. This kind of integration means that a Windows Media DRM Server must implement some module to make it capable to understand (or transform) AXMEDIS objects into Windows Media Objects. The idea is that Windows Media DRM Server can be also a client of AXMEDIS PMS. When Windows Media DRM Server obtains some AXMEDIS Object and its licenses it should be able to create an equivalent Windows Media Object and protect it with Windows DRM Licenses. Perhaps these transformations cannot be done because different semantics of each kind of licenses, but some transformations could be performed. Another issue is that if these object transformations between AXObject and Windows Media Objects are really legal.

Moreover, supposing that this object interaction is possible and legal another important issue exists in the supervision of the content usage. The PMS Server has to inform the AXCS about all the actions made by the final application in order to catch possible malicious usages. So, it has to be also integrated these interactions between Windows Media Server and AXMEDIS PMS Server.
This solution can be seen as a DRM Gateway between these different technologies, but taking into account the possible different incompatibilities. Probably the most usual, common and simple licenses can be interoperable, but the most complex licenses may contain rights and conditions that are not suitable in both formats, so no transformation could be done.

4.5.2 Integration of OMA tools

Regarding interoperability of DRM, we have been working on the interoperability of rights expression languages, especially MPEG-21 REL and OMA DRM REL. The work performed in this area is reported in De 4.7.1.2 and De 4.3.1.2.

It has included the creation of new DRM and UML models for OMA DRM REL, based on the previous experience on MPEG-21 REL and a preliminary study on the possibility of using these models for the translation of rights expressions between these two languages [91].

4.6 Licenses for supporting TISCALI requirements

This section describes a set of licenses that try to accomplish the specific requirements made by TISCALI on DE 2.1.1.2.1, Update of requirements. Some of the requested rules cannot be expressed by current version of MPEG-21 REL. It is foreseen that AXMEDIS project will create an extension of REL to provide support to some capabilities requested by legal users. Missing conditions for the TISCALI case could be also considered for the extension of REL task.

The requirements provided are:

- activation of rights at a fixed date
- counted number of plays (N variable)
- burn option on removable device (CD, DVD etc)
- counted number of devices
- license server unavailability URL to be hard coded in DRM file
- support for high volume video files (5-5GB)
- support high volume video streaming (up to 10Mb/s)

The following sections show the licenses that can be expressed accomplishing the requirements. The requirements not present is because they cannot be currently expressed with MPEG-21 REL.

4.6.1 Activation of rights at a fixed date

The following licenses describe the condition of allowing the exercise of rights at a fixed date. The corresponding distributor and final user licenses are provided.

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-
REL-R-NS ../schemas/rel-r.xsd
                             urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd
                            urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">
        <r:grantGroup>
                <r:grant>
                        <r:keyHolder>
                                <r:info>
                                        <dsig:KeyName>AXDID:TISCALI   </dsig:KeyName>
                                </r:info>
                        </r:keyHolder>
                        <r:issue/>
                        <r:grant>
                                <mx:play/>
                                <mx:diReference>
                                        <mx:identifier>urn:axmedis:AX01</mx:identifier>
                                </mx:diReference>
                                <r:validityInterval>
                                        <r:notBefore>2006-10-04T12:00:00</r:notBefore>
                                </r:validityInterval>
                        </r:grant>
                </r:grant>
        </r:grantGroup>
        <r:issuer>
                <r:keyHolder>
                        <r:info>
                                <dsig:KeyName>AXDID:MTLABS</dsig:KeyName>
                        </r:info>
                </r:keyHolder>
        </r:issuer>
</r:license>
```

Figure 21 – Distribution license for fixed date

The above figure shows an example of a license distributed by MTLABS (issuer) to TISCALI (principal). This license allows TISCALI to distribute the right play over the urn:axmedis:AX01 object on date 4/10/2006 at 12.

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-
REL-R-NS ../schemas/rel-r.xsd
                             urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd
                            urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">
        <r:grantGroup>
                <r:grant>
                        <r:keyHolder>
                                <r:info>
                                        <dsig:KeyName>AXDID:Peter</dsig:KeyName>
                                </r:info>
                        </r:keyHolder>
                        <mx:play/>
                        <mx:diReference>
                                        <mx:identifier>urn:axmedis:AX01</mx:identifier>
                                </mx:diReference>
                        <r:validityInterval>
                                <r:notBefore>2006-10-04T12:00:00</r:notBefore>
                        </r:validityInterval>
                </r:grant>
        </r:grantGroup>
        <r:issuer>
                <r:keyHolder>
```

```
                        <r:info>
                                <dsig:KeyName>AXDID:TISCALI</dsig:KeyName>
                        </r:info>
                </r:keyHolder>
        </r:issuer>
</r:license>
```

Figure 22 – Final user license for fixed date

The above figure shows an example of a final user license distributed by TISCALI (issuer) to AXDID:Peter (principal). This license allows Peter play the urn:axmedis:AX01 object on date 4/10/2006 at 12. The grant given to the final user must be the same as the one described in the distribution license.

4.6.2 Counted number of plays

The following licenses describe the condition of allowing the exercise of a right a number of times. The corresponding distributor and final user licenses are provided.

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-
REL-R-NS ../schemas/rel-r.xsd
                         urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd
                         urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">
        <r:grantGroup>
                <r:grant>
                        <r:keyHolder>
                                <r:info>
                                        <dsig:KeyName>AXDID:TISCALI   </dsig:KeyName>
                                </r:info>
                        </r:keyHolder>
                        <r:issue/>
                        <r:grant>
                                <mx:play/>
                                <mx:diReference>
                                        <mx:identifier>urn:axmedis:AX01</mx:identifier>
                                </mx:diReference>
                                <sx:exerciseLimit>
                                        <sx:count>5</sx:count>
                                </sx:exerciseLimit>
                        </r:grant>
                </r:grant>
        </r:grantGroup>
        <r:issuer>
                <r:keyHolder>
                        <r:info>
                                <dsig:KeyName>AXDID:MTLABS</dsig:KeyName>
                        </r:info>
                </r:keyHolder>
        </r:issuer>
</r:license>
```

Figure 23 – Distribution license for number of times

The above figure shows an example of a license distributed by MTLABS (issuer) to TISCALI (principal). This license allows TISCALI to distribute the right play over the urn:axmedis:AX01 object. Final user will be allowed to make 5 plays.

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-
REL-R-NS ../schemas/rel-r.xsd
                         urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd
                         urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">
        <r:grantGroup>
                <r:grant>
```

```
                        <r:keyHolder>
                                <r:info>
                                        <dsig:KeyName>AXDID:Alice</dsig:KeyName>
                                </r:info>
                        </r:keyHolder>
                        <mx:play/>
                        <mx:diReference>
                                <mx:identifier>urn:axmedis:AX01</mx:identifier>
                        </mx:diReference>
                        <sx:exerciseLimit>
                                <sx:count>5</sx:count>
                        </sx:exerciseLimit>
                </r:grant>
        </r:grantGroup>
        <r:issuer>
                <r:keyHolder>
                        <r:info>
                                <dsig:KeyName>AXDID:TISCALI   </dsig:KeyName>
                        </r:info>
                </r:keyHolder>
        </r:issuer>
</r:license>
```

Figure 24 – Final user license for number of times

The above figure shows an example of a final user license distributed by TISCALI (issuer) to AXDID:Alice (principal). This license allows Alice to play the urn:axmedis:AX01 object for five times. The grant given to the final user must be the same as the one described in the distribution license.

4.6.3 Counted number of devices

The following licenses describe the condition of allowing the exercise of a right on a number of devices. The corresponding distributor and final user licenses are provided.

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-
REL-R-NS ../schemas/rel-r.xsd
                         urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd
                         urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">
        <r:grantGroup>
                <r:grant>
                        <r:keyHolder>
                                <r:info>
                                        <dsig:KeyName>AXDID:TISCALI   </dsig:KeyName>
                                </r:info>
                        </r:keyHolder>
                        <r:issue/>
                        <r:grant>
                                <mx:play/>
                                <mx:diReference>
                                        <mx:identifier>urn:axmedis:AX01</mx:identifier>
                                </mx:diReference>
                                <r:allConditions>
                                        <mx:renderer licensePartId="trustedDevice1">
                                                <r:keyHolder>
                                                        <r:info>

        <dsig:KeyName>AXTID:DEVICE01</dsig:KeyName>
                                                        </r:info>
                                                </r:keyHolder>
                                        </mx:renderer>
                                        <mx:renderer>
                                                <r:keyHolder licensePartId="trustedDevice2">
                                                        <r:info>

        <dsig:KeyName>AXTID:DEVICE02</dsig:KeyName>
                                                        </r:info>
                                                </r:keyHolder>
                                        </mx:renderer>
                                </r:allConditions>
                        </r:grant>
                </r:grant>
        </r:grantGroup>
        <r:issuer>
                <r:keyHolder>
                        <r:info>
                                <dsig:KeyName>AXDID:MTLABS</dsig:KeyName>
                        </r:info>
                </r:keyHolder>
        </r:issuer>
</r:license>
```

Figure 25 – Distribution license for number of devices

The above figure shows an example of a license distributed by MTLABS (issuer) to TISCALI (principal). This license allows TISCALI to distribute the right play over the urn:axmedis:AX01 object. Final user will be allowed to make play only with two trusted devides AXTID:DEVICE01 and AXTID:DEVICE02.

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-
REL-R-NS ../schemas/rel-r.xsd
                         urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd
                         urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">
        <r:grantGroup>
                <r:grant>
                        <r:keyHolder>
                                <r:info>
                                        <dsig:KeyName>AXDID:Alice</dsig:KeyName>
                                </r:info>
                        </r:keyHolder>
                        <mx:play/>
                        <mx:diReference>
                                <mx:identifier>urn:axmedis:AX01</mx:identifier>
                        </mx:diReference>
                        <r:allConditions>
                                <mx:renderer licensePartId="trustedDevice1">
                                        <r:keyHolder>
                                                <r:info>
                                                        <dsig:KeyName>AXTID:DEVICE01</dsig:KeyName>
                                                </r:info>
                                        </r:keyHolder>
                                </mx:renderer>
                                <mx:renderer>
                                        <r:keyHolder licensePartId="trustedDevice2">
                                                <r:info>
                                                        <dsig:KeyName>AXTID:DEVICE02</dsig:KeyName>
                                                </r:info>
                                        </r:keyHolder>
                                </mx:renderer>
                        </r:allConditions>
                </r:grant>
        </r:grantGroup>
        <r:issuer>
                <r:keyHolder>
                        <r:info>
                                <dsig:KeyName>AXDID:TISCALI   </dsig:KeyName>
                        </r:info>
                </r:keyHolder>
        </r:issuer>
</r:license>
```

Figure 26 – Final user license for number of devices

The above figure shows an example of a final user license distributed by TISCALI (issuer) to AXDID:Alice (principal). This license allows Alice to play the urn:axmedis:AX01 object on trusted devices AXTID:DEVICE01 and AXTID:DEVICE02. The grant given to the final user must be the same as the one described in the distribution license.

5 Supporting AXMEDIS DRM in Windows Media (DSI)

This work has been performed to verify the possibility of using DRM model and format into a Windows Media delivering of content.

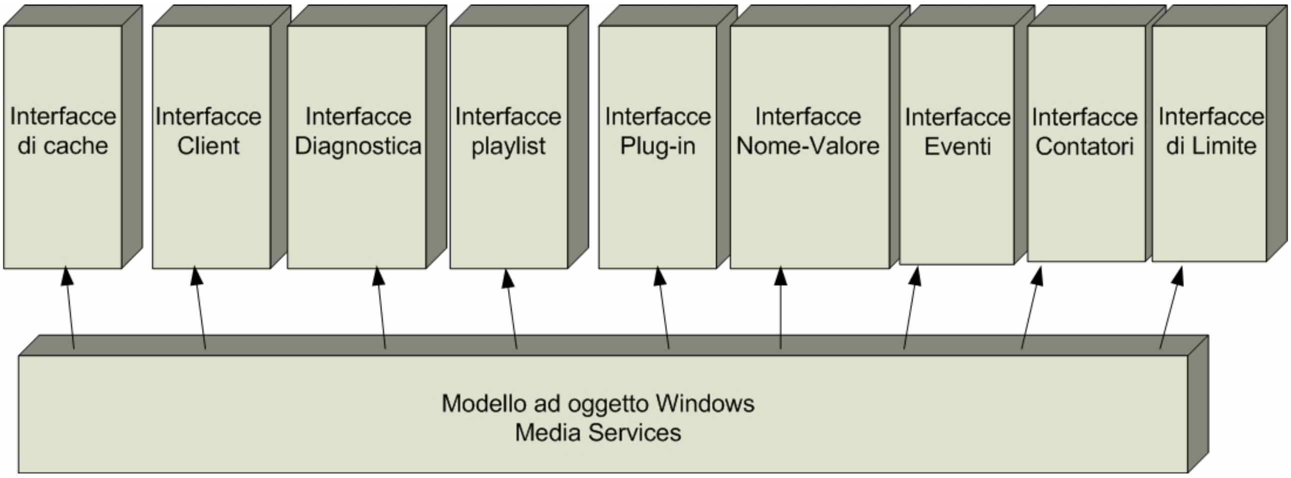
Details can be found in the Annex 1 of this document.

Windows Media DRM and architecture allow to customize several aspects of the delivering and into the players with specific plug ins of:

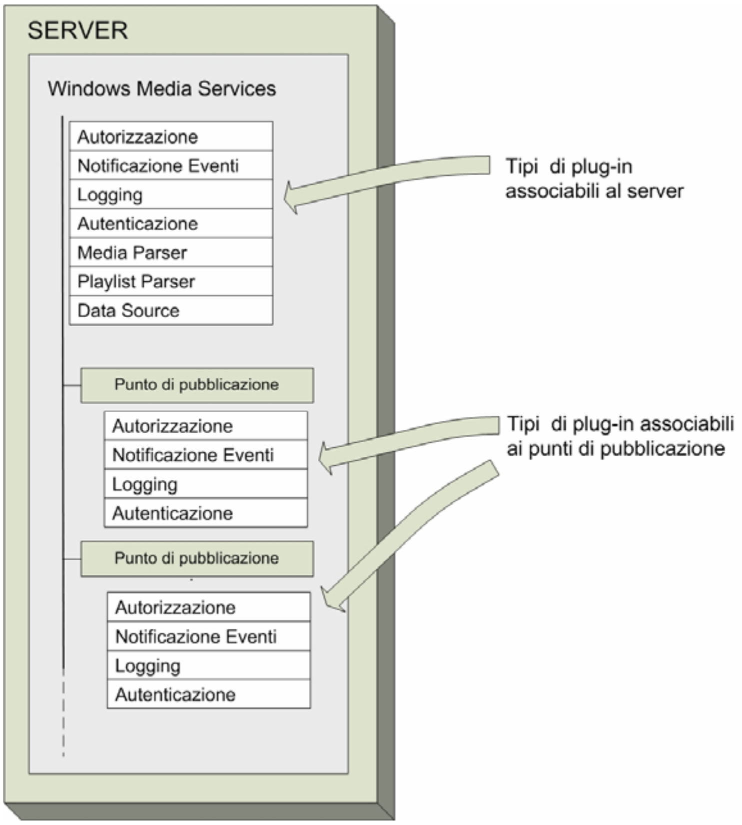
- Windows Media Services
- Windows Media Player
- Windows Media Encoder

5.1 Windows Media Services

It is based on COM components and it is based on points of publication for ondemand and broadcast.



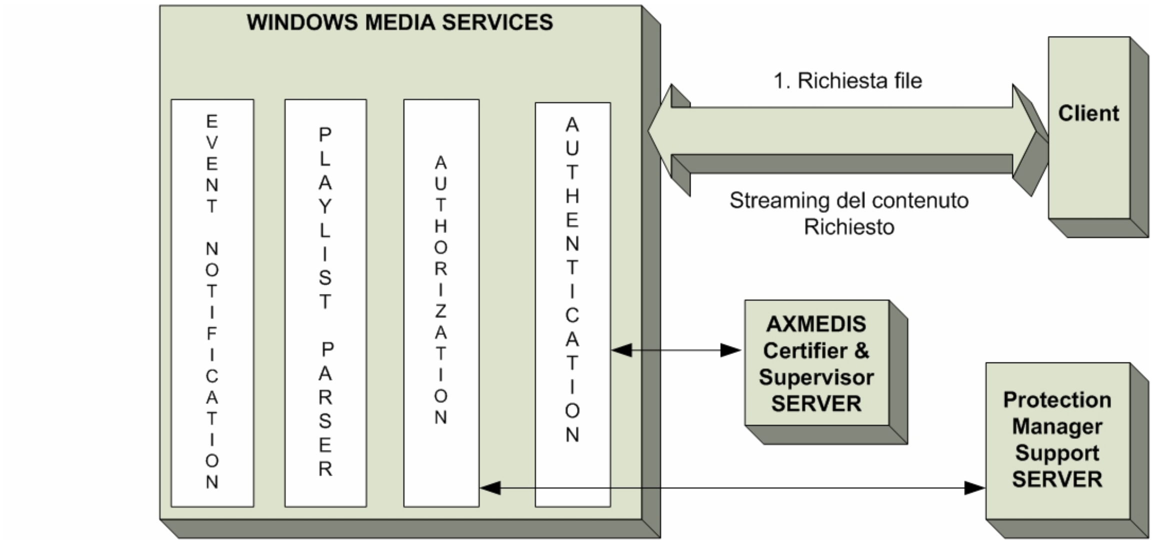
It is based on Authentication, authorisation, cache proxy, data source, event notification, logging and play list parser.



5.2 Streaming of AXMEDIS objects into Windows Media

Problems:

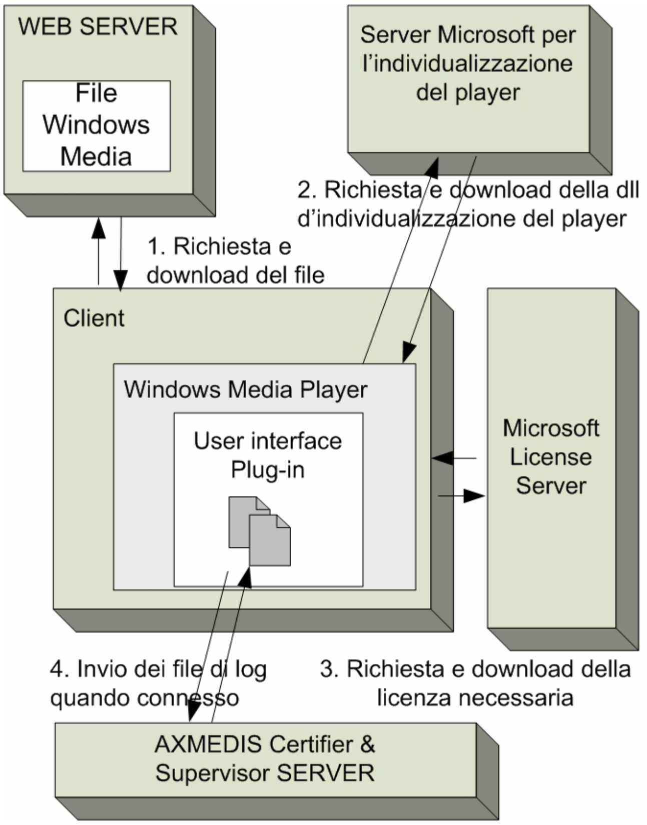
- Streaming of AXMEDIS objects
- Authentication of the client
- Authorization of the client



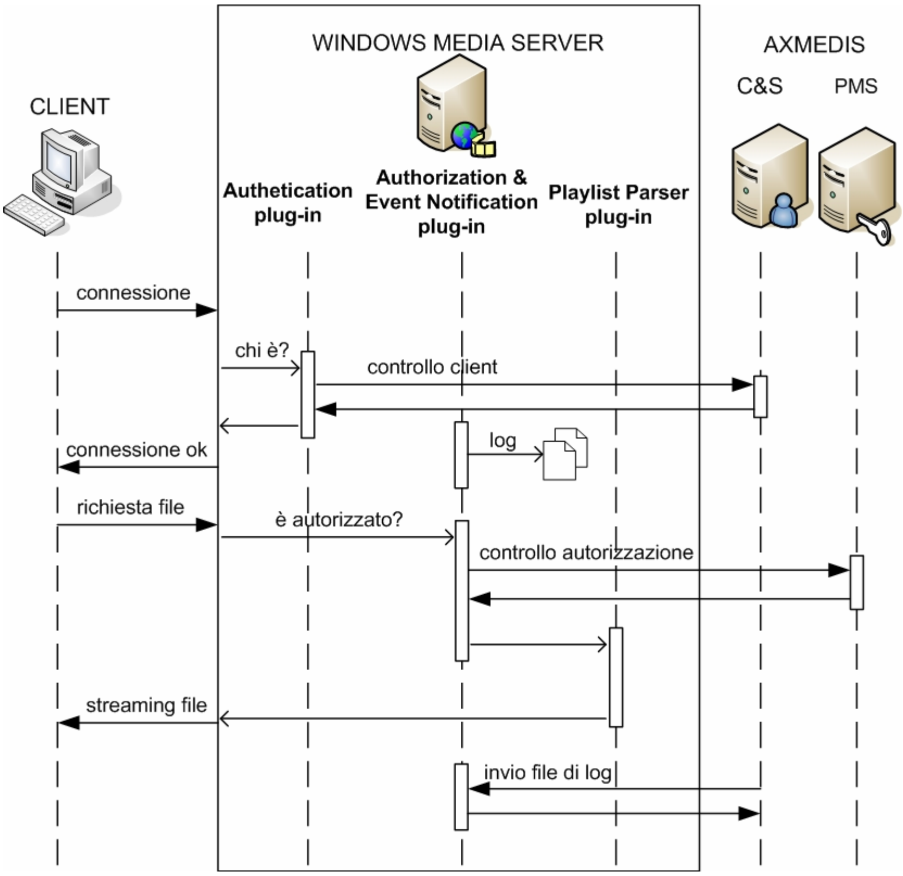
This can be realized by implementing a set of specific plugin for:

- Authentication
- Authorization
- Event notification
- Playlist parsing

While the AXMEDIS object has to pass in the Windows Media stream as converted in a Windows Media Object.

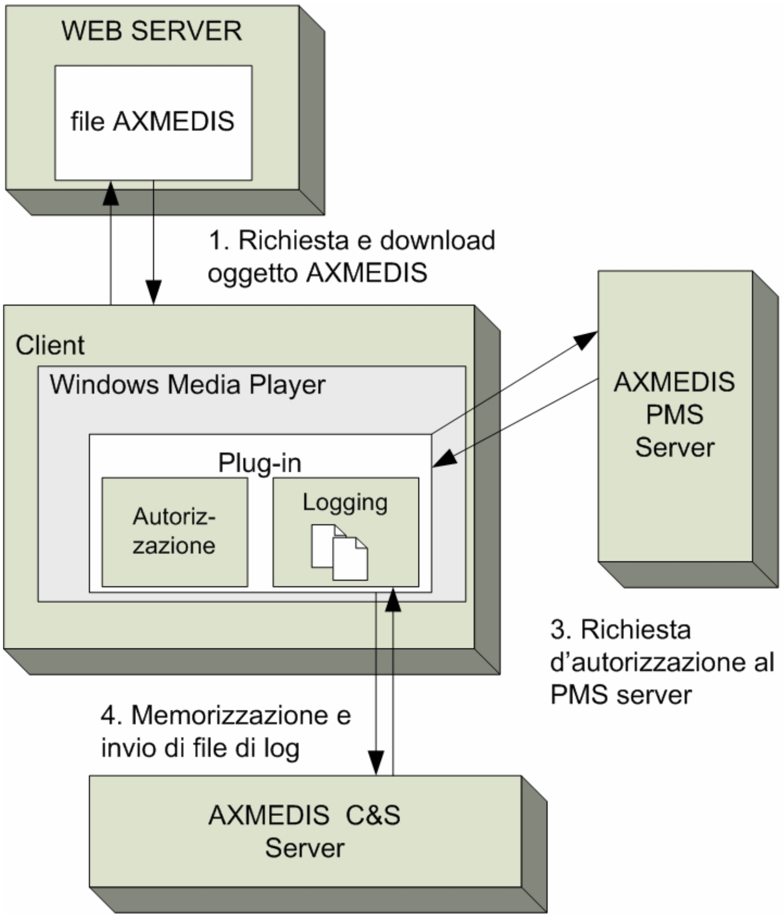

WEB SERVER
File Windows Media
Server Microsoft per l'individualizzazione del player
2. Richiesta e download della dll d'individualizzazione del player
1. Richiesta e download del file
Client
Windows Media Player
User interface Plug-in
Microsoft License Server
4. Invio dei file di log quando connesso
3. Richiesta e download della licenza necessaria
AXMEDIS Certifier & Supervisor SERVER

The Windows Media DRM cannot be customized. The license server has to be WM. This means that AXMEDIS/MPEG-21 licenses have to be translated into WM licenses and posted on the server. Action Log can be passes to the AXCS, and also the other aspects.



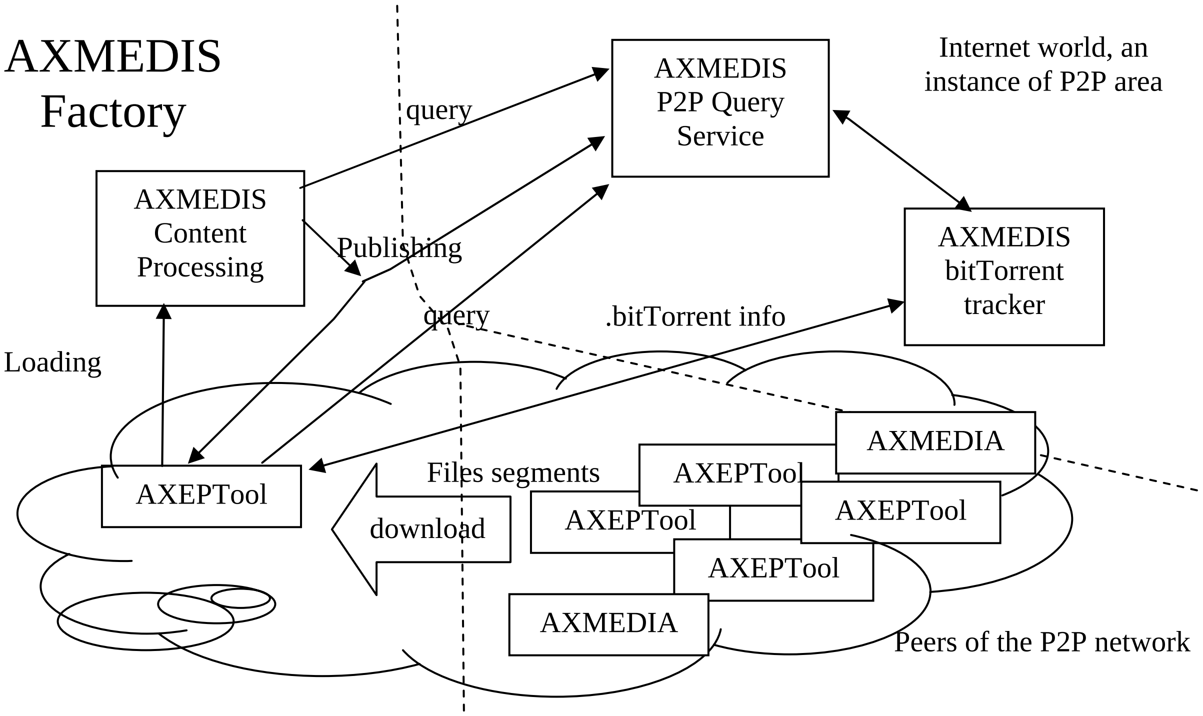
5.3 Reproduction of AXMEDIS objects into Windows Media

In this case, the AXMEDIS object is delivered from an AXMEDIS server to a computer endowed of a Windows Media Player. An AXMEDIS plug in can be realized into the Windows Media Player to interact to the AXMEDIS PMS and AXCS for the DRM management.


WEB SERVER
file AXMEDIS
1. Richiesta e download oggetto AXMEDIS
Client
Windows Media Player
Plug-in
Autoriz-zazione
Logging
AXMEDIS PMS Server
3. Richiesta d'autorizzazione al PMS server
4. Memorizzazione e invio di file di log
AXMEDIS C&S Server

6 P2P support for B2C application and distribution (DSI)

The solution identified by CRS4 has been completely supersede by a new solution based on BitTorrent.
The general solution and architecture of AXMEDIS framework tools integrated with P2P tools is reported in the previous section. Technically it can be depicted as follows.



The P2P Clients can be both: AXEPTool and/or AXMEDIA P2P tools.

In this architecture for the B2B level we have:

- **AXMEDIS factory** (under responsibility of DSI), a place in which the AXMEDIS objects are created and/or are used to create other more complex objects, or are distributed towards other distribution channels, for example, broadcast,
- **AXMEDIS P2P Query Service** (under responsibility of EXITECH): a server located to cover a P2P community in which the database of metadata are located and that collect also the .bitTorrent information for the related AXMEDIS objects. AXMEDIS P2P Query Service may be connected with other AXMEDIS P2P Query Services to create a higher level of P2P sharing of content.
- **AXMEDIS BitTorrent Tracker**: a server derived from BitTorrent Tracker technology and solution to host BitTorrent information, update them, and provide them to AXMEDIS P2P clients according to the BitTorrent protocol. In a global geographic system, many AXMEDIS BitTorrent Trackers may survive to provide services.
- **.bitTorrent info:** BitTorrent information file containing the information created by a bitTorrent Maker processing the file to be shared into the P2P area.
- **AXEPTool**: a P2P client tool derived from a BitTorrent Client for B2B P2P sharing of AXMEDIS objects.
- **AXMEDIA**: a P2P client tool derived from a BitTorrent Client for C2C AXMEDIS object sharing.

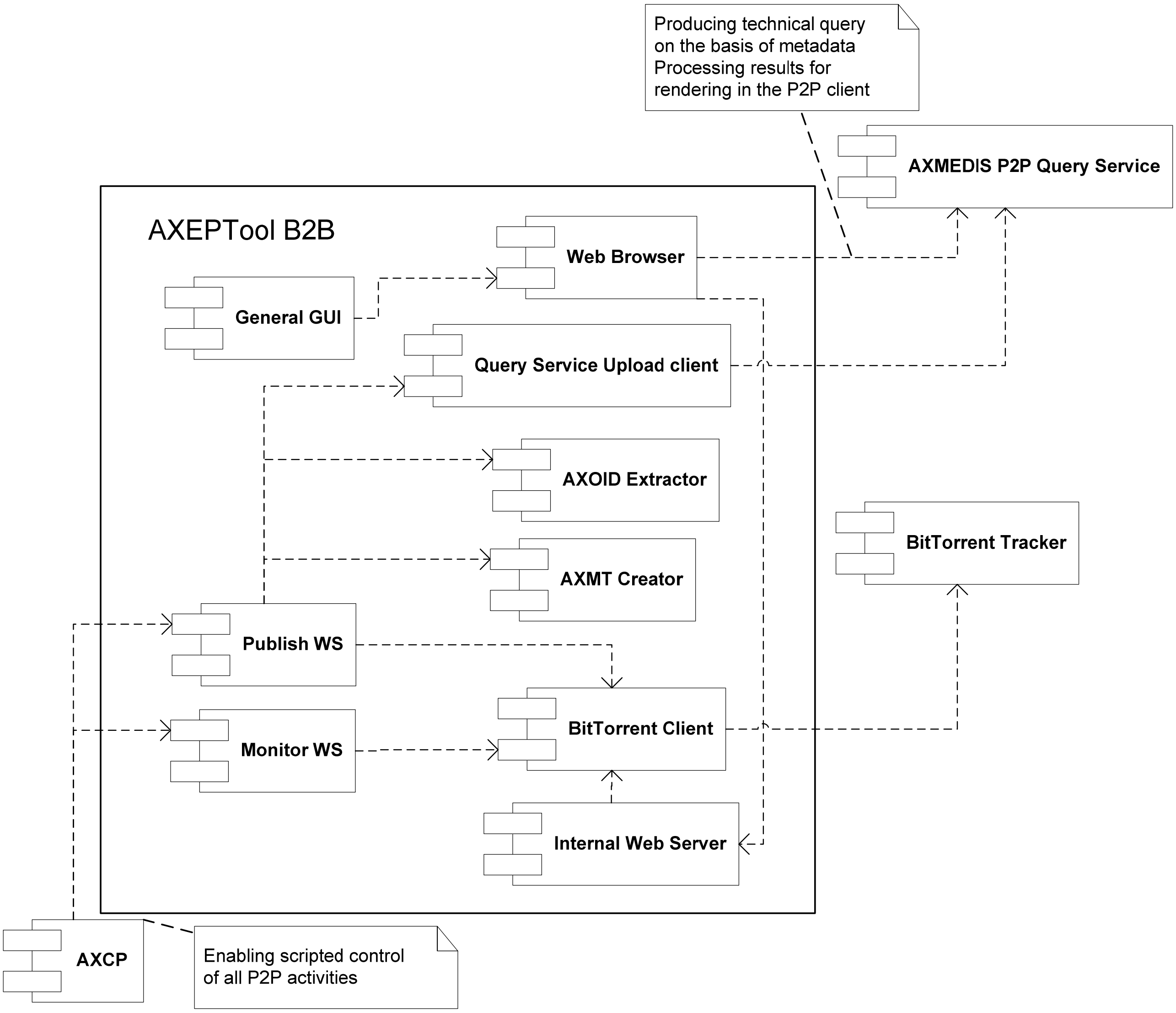
In sharing files, the AXEPTool may use the files contained into the local repositories of the AXMEDIA tools and vice versa. So that the P2P network is practically defined by the “AXMEDIS P2P Query Service” and by clients that get connection to the AXMEDIS BitTorrent Tracker to get the specific BitTorrent information associated to a download.

AXEPTool is a P2P client tool derived from bitTorrent Client for B2B P2P sharing and thus also for interacting with the AXMEDIS Factory and specifically with the AXCP tool.

The AXMEDIS AXEPTool P2P Client is going to be realized with .bitTorrent Azureus client (see the list in the previous section), which has as major features those presented in the previous table that for the purpose can be summarized as:

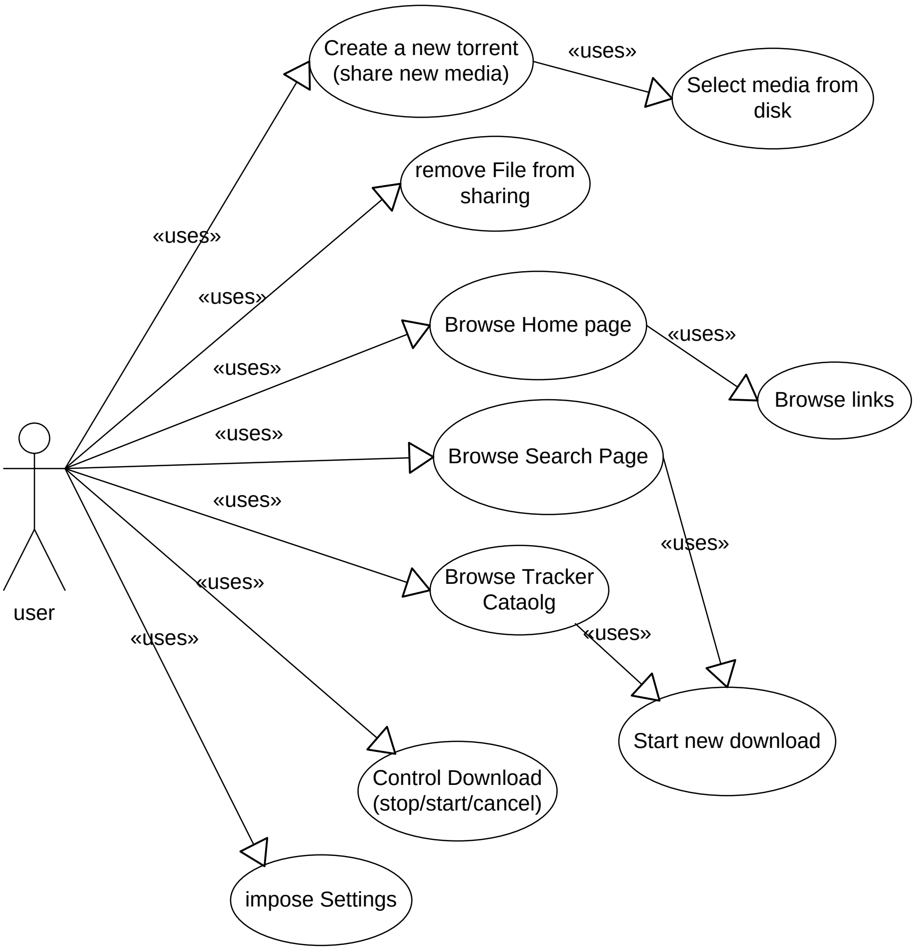
- Developed in Java
- Support protected communication with the Tracker
- Presents a monitor interface

It seems that the present Plugin technology of Azureus does not allow to do much and for sure not to control downloads and other facts that we need to enforce. So that the work is in effect to the modification of the P2P client.



The AXEPTool is a special P2P BitTorrent Client Node, suitable to cover the above mentioned requirements and roles in the P2P Network for B2B. It is used by content producers, distributors, integrators, aggregators, and expert users, etc., for automating activities of publishing, downloading and monitoring of the P2P network. Therefore, it accepts formal requests of publishing (file sharing), downloading, and monitoring via Web Services. In the AXMEDIS solution, the AXCP GRID Node may use those Web Services from the Java

Script code which can be executed by any GRID node of the AXCP GRID. In addition, specific tools may be created for interacting with the AXEPTool via Web Services. In more details, the Web Services exposed by the AXEPTool P2P Clients, the AXQuery Support and the information provided by the AXTrack allow the AXCP GRID to automate procedures publishing, downloading and monitoring the AXMEDIS P2P Network.



The AXMEDIS AXEPTool P2P Client:

- Includes an Integrated Web Browser and an Internal Web Server in order to easily communicate with the Web Portal realized by P2P Query Service.
- Provides a **P2PPublisher WS** for receiving objects from another tool (for example from the AXMEDIS AXCP Engine) to be published on the P2P network and thus for posting the BitTorrent information file and/or making accessible the full version of the file for other peers. It provides a functionality via WEB service to **bitTorrentURL publishObject(localFileName or filepath, AXOID, trackerURL)** publication of an object, its metadata have been already posted on the AXMEDIS P2P Query Service Server; the contract consist in posting/saving the AXMEDIS object passed into the **Published Object repository** of the P2P client tool and create a bittorrent file, thus posting bittorrent file on the selected tracker URL. If you have already published an object, and you call this functionality with a local file name it tries to post the correspondent .torrent file on the Tracker. Using filepath instead provides object publishing too on the Tracker's catalogue.
- The objects (AXMEDIS or not) can also be published manually from the user interface of the AXEPTool.
- Include two different repository one for Downloaded (or downloading) and one for Published objects. The bittorrent will automatically make available a downloaded media (or media) part, the distinction is only for clearly distinguish which content has been introduced in the P2P network from the client itself.

- Provides a **Monitoring Tools User Interface** for:
  - Observing the status of the plan to Downloaded files, % of completion, who is involved, download rate, etc.
  - Estimation of the time to completion
  - All the examples related to the Azureus Client have been depicted in the previous sections.
- Provides a **P2PMonitoring WS** with the following functionalities:
  - P2PObjectList getPublishedObjectsList(), returns a list of objects which are present in the directory of Published objects. The list depicts all media (both AXMEDIS or general), including filename, AXOID (if any) and infoHash.
  - PublishStatus getPublicationStatus(AXOIDFileName selectedMedia), provides the current status of a given object which is in the Published Object area of the P2P environment and selected by the proper identification (AXOID or localFileName). It returns the amount of bytes, seeds and peers for each segment, and the list of contacting peer for a given media.
  - result removePublishedObject(AXOIDFileName selectedMedia), deletes the object specified by the ID from the database/directory of Published objects. This is something that should not to be done in P2P systems, but when they are used for B2B the directory could reach the limit of their size and housekeeping is needed.
  - result downloadObject(AXOID), starts the download of the object especified by the AXOID in the P2P environment. The AXOID field has to be added into the Tracker database.
  - result downloadObjectURL(bittorrentURI), starts the download of the object specified by the torrent file URL in the P2P environment.
  - DownloadStatus monitorDownloadingStatus(AXOIDFileName selectedMedia), provides the current status a downloading objects in the P2P environment. It returns the current percentage of download, estimated time to completion (if possible), peers and seeds.
  - P2PObjectList getDownloadingObjectsList(), returns the list of objects in the directory of Downloaded objects.
  - result controlDownloadingObject(AXOIDFileName selectedMedia, boolean start/stop), start/stops the download of the object specified by identification (AXOID or localFileName).
  - result removeDownloadedObject(AXOIDFileName selectedMedia), deletes the object specified by the ID from the directory of Downloaded objects. This is something that should not to be done in P2P systems, but in some cases the directory could reach the limit of their size and housekeeping is needed.
  - mediaURL getDownloadedObjectURI (AXOIDFileName selectedMedia), retrieves the object URI from which the downloaded at 100% object can be copied into the AXMEDIS database by the AXCP. This action will be performed by the AXCP directly accessing to that URL.
  - ObjectStatus getStatus (AXOIDFileName selectedMedia), retrieves the status of an object in the AXEPTool's transfer list. This can be Downloading, Paused, Checking, Seeding, Error, Queued.
  - Sysinfo getSystemInformations(), retrieves the main system information and profile of the AXEPTool: CPU type, Operating System, RAM, AXEPTool's configuration file in XML format, User Name, AXEPTool's version, free HD space where AXEPTool is running

AXMEDIS P2P Query Service is a server that provides information about the objects indexed into the AXDB and at disposal of a community. Several Communities can be created, for example one for each distributor, by using and installing a different AXMEDIS P2P Query Service SERVER. A query service that could be a simple redirection of the AXMEDIS database query on other AXMEDIS factories (see http://www.axmedis.org/documenti/view_documenti.php?doc_id=1728). For more details on the structure of the AXMEDIS query support see: http://www.axmedis.org/documenti/view_documenti.php?doc_id=1932

AXMEDIS P2P Query Service Server includes

- **AXDB: AXMEDIS Database and its services** which is practically an AXMEDIS database that is mainly used to:
  - store the metadata of the objects. The objects that will be received are stored, but they are mainly stub objects without real resources inside in order to save time and bandwith

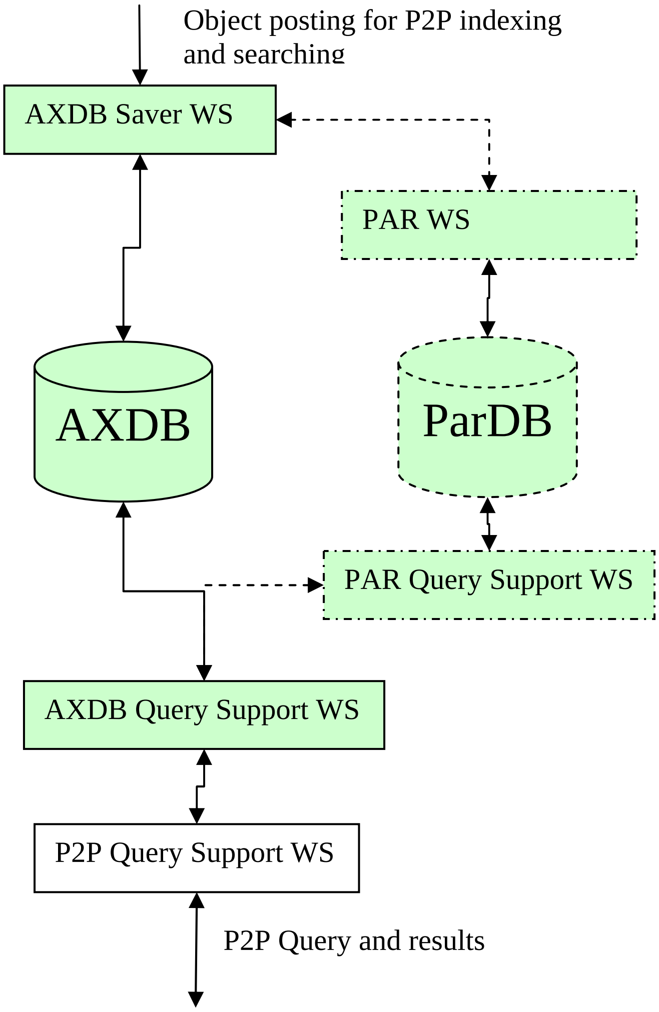
  - publishing/loading a new AXMEDIS object metadata and additional information
  - deleting/un-publishing of an AXMEDIS object.
- **AXMEDIS P2P Query Service Web Service** allows to establish a channel for:
  - receiving a query and providing a result in terms of AXOID, metadata. Torrent files will be recovered from the tracker from the AXOID.
- **AXMEDIS Web Query User Interface** allows to common users to make queries on the portal directly, the results are those described above. This is a simple and direct interface to make queries, probably used only by final users, unless a more suitable interface is implemented in the P2P tool. From that moment this will became an administrative interface only.

Please note that the **AXMEDIS P2P Query Service Web Service** can be used by:

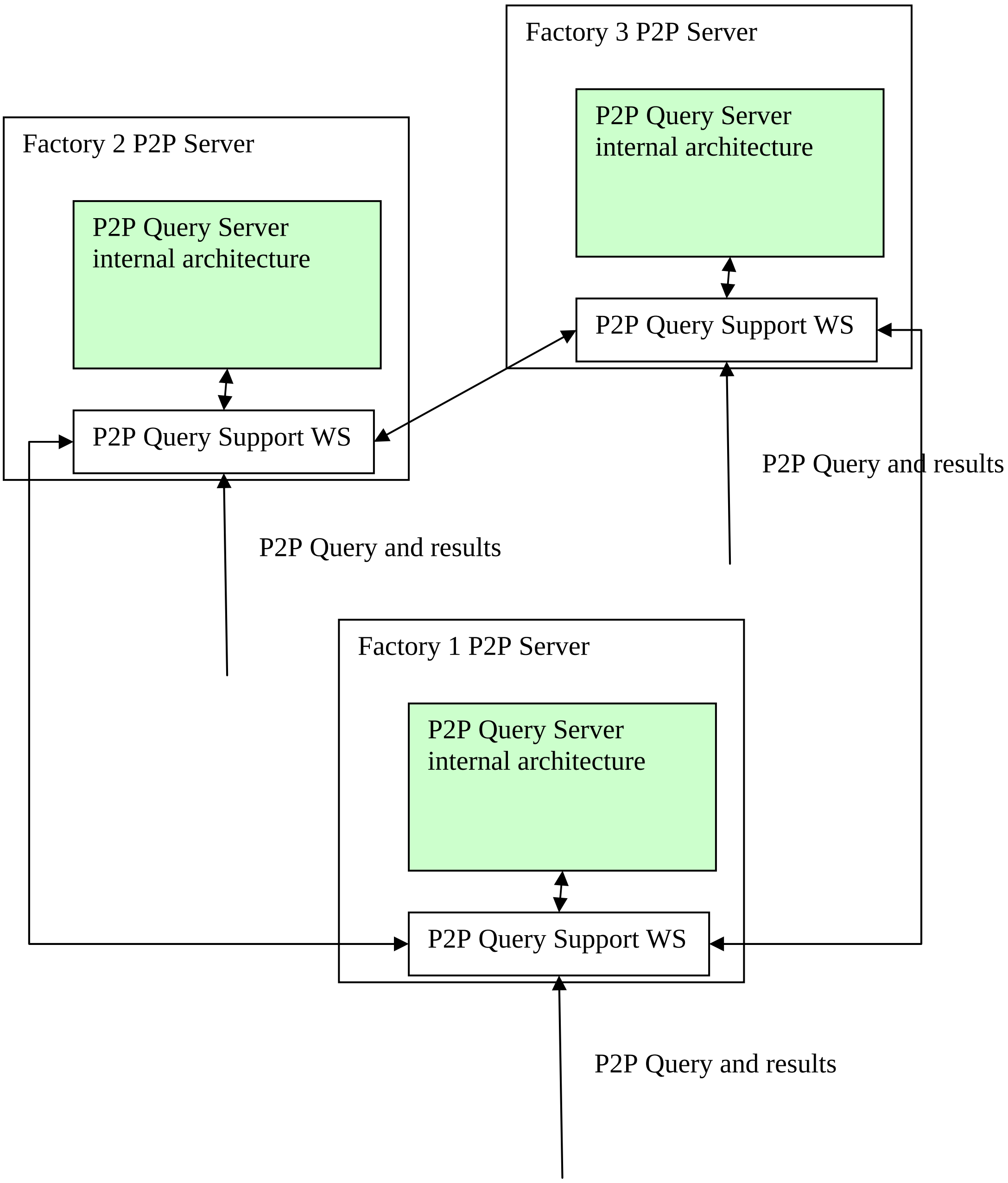
- the AXCP for automating the search for new objects in the P2P network
- the AXMEDIS query support for making a query on the Virtual Network of P2P tools of the several factories.

The architecture of the P2P Query Web Service is aimed to minimize code duplication maximizing at the same time the reuse of stable and already tested code. This architecture establishes the basis for a future scalability of the system on the basis of the needs that will be identified during the testing of the P2P system.

The component in green are reused for the Query Support architecture, while the white are developed for P2P. The green component with dashed lines can be considered optional.

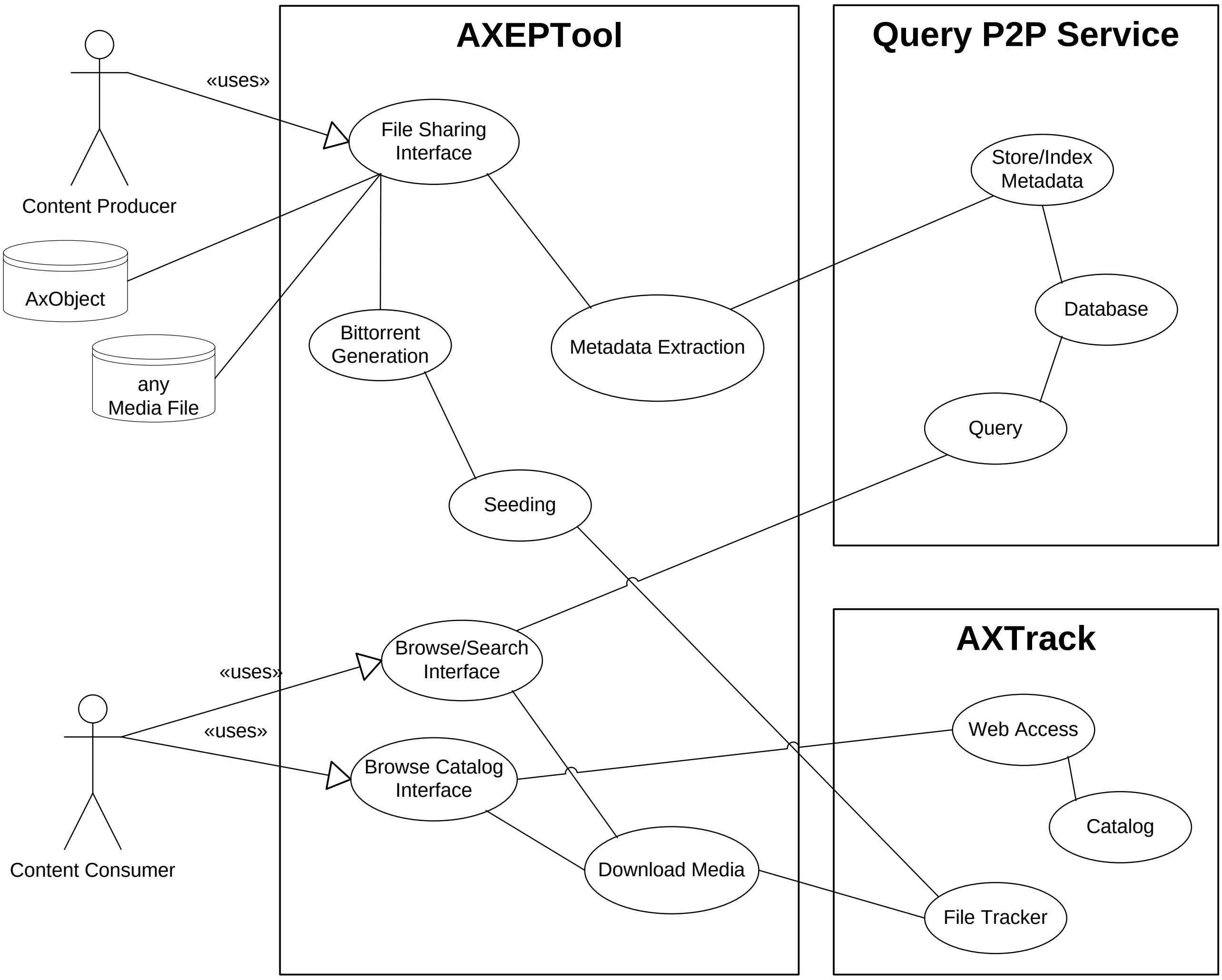


The posting of an object to the AXDB is done by the submission of a simplified object without the resources since it is only necessary to index metadata of the object and not to store the entire object in the AXDB.
Moreover it is possible to save also bittorrent information by adding a custom metadata field that can be recovered by using the query mechanism together with the AXOID.
The P2P Query Support WS is the interface between the Main Query Support defined in DE 3.1.2.2.9 and the P2P Query server. This layer and the AXDB Query Support WS layer can be merged together in order to maximize performance if needed, but leaving them separate allows the scalability of the architecture toward a more general approach of querying multiple node of the P2P architecture allowing for example to implement the architecture depicted below.



In this model the query on each factory can propagate to automatically to other factories and the results are collected together by the P2P Query Support WS layer of each factory that became then a complex component replicating the functions of the MainQuerySupport at the P2P level.

Here is the general use-case covering the exchange of files between a content producer and a customer. In the typical scenario, there is one content producer, but there are many customers.



The content producer connects to a file sharing interface and use it to put the file on the peer to peer network. First metadata and bittorrent info are generated locally from the file. Those metadata are then uploaded to one or more AXMEDIS P2P Query Service, where they are indexed for subsequent query, and to one or more AXTrack Web Service as seed notification. The File Tracker starts keeping track of the files.

After the tracker is ready to track the files, the AXEPTool could automatically start a seeding process (not mandatory, but configurable option). The seeding process will then register itself to the tracker, so that potential customers will know that the content producer has one full copy of the file.

The content download can start by operating in two different manners:

1. the search approach and the browse approach. The first one is based on the Query P2P Service and the information it stores.
2. the second it is simply a list which exhaustively reports all the content which is downloadable by the AxTrack instance.

The downloading process will connect to the tracker sending the information that he has an empty file list. The tracker will then inform it of the places where it can retrieve the file parts.

In the above example, there is only one seed and one downloader, so the answer of the tracker will be to grab it from the content producer. But, as soon as the customer has downloaded at least one small file part, the tracker will redirect other willing customers to both the content producer and the customer.

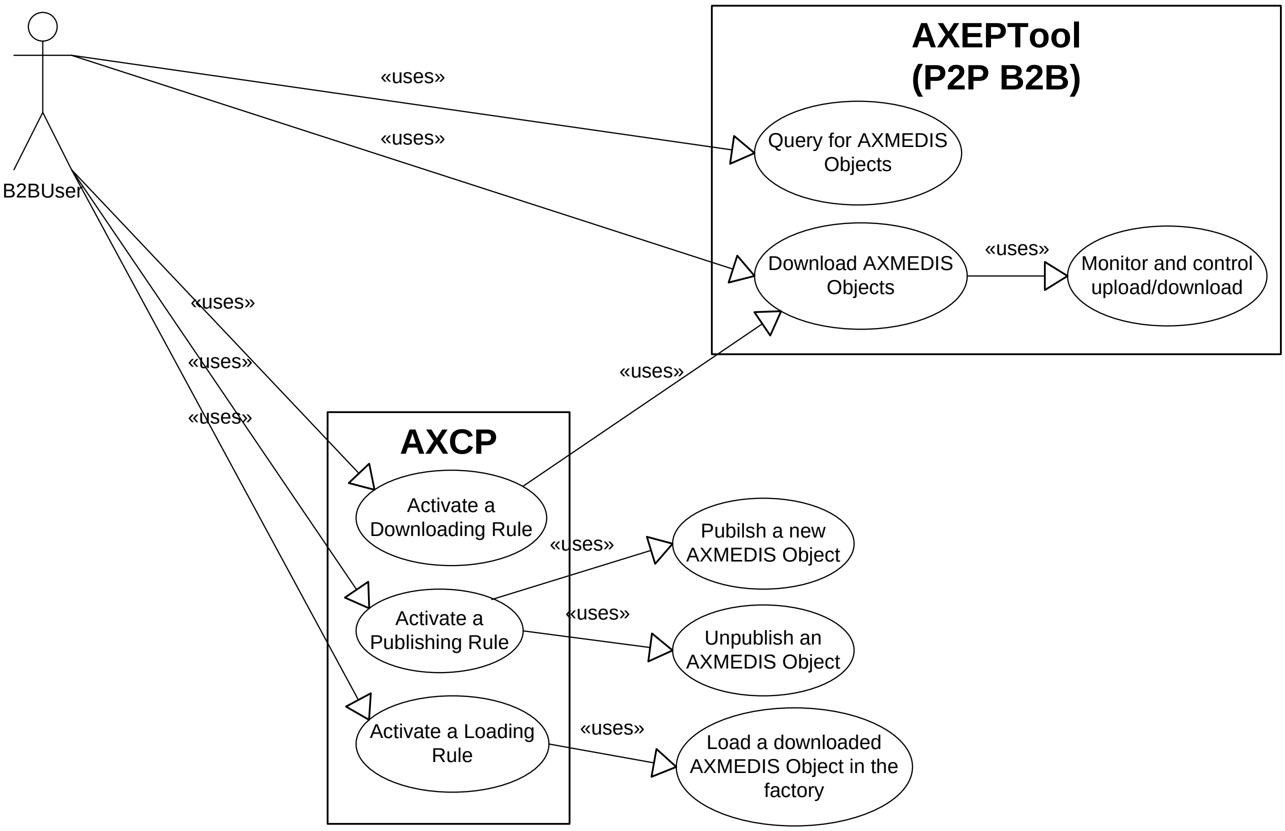
The traditional bittorrent protocol take places at this phase, whit the sole variation of being able to select to filter media in AXMEDIS form and/or traditional formats (e.g., avi, mp3, zip).

6.1 Usage walk-through

In this section, the main use cases have been depicted in order to clarify functionalities to be exposed by AXEPTool (P2P B2B client).
In the following use-case diagram the main B2B user action has been included.

The use cases of main interest are those considered for AXEPTool:

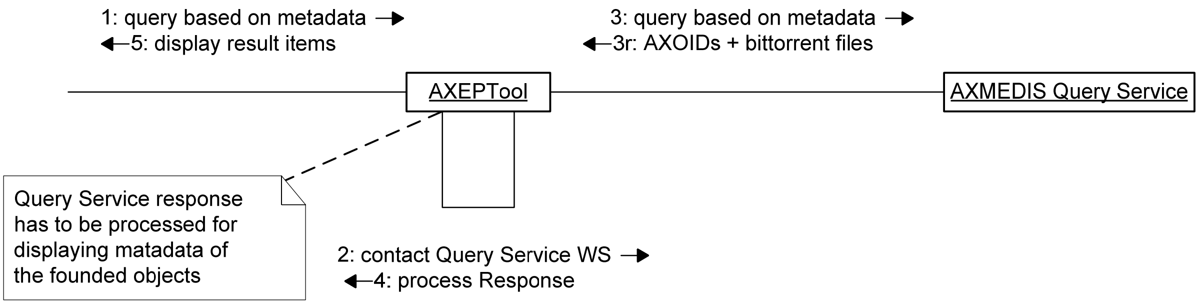
- The user can query for content directly on the client interface (even embedding a web-page rendering);
- The user can command to initiate the download process for a given query result; this use case include the typical action of monitoring and controlling the active downloading process.


B2BUser
«uses»
«uses»
«uses»
«uses»
«uses»
«uses»
AXEPTool
(P2P B2B)
Query for AXMEDIS Objects
Download AXMEDIS Objects
«uses»
Monitor and control upload/download
AXCP
Activate a Downloading Rule
Activate a Publishing Rule
Activate a Loading Rule
«uses»
«uses»
«uses»
Pubilsh a new AXMEDIS Object
Unpublish an AXMEDIS Object
Load a downloaded AXMEDIS Object in the factory

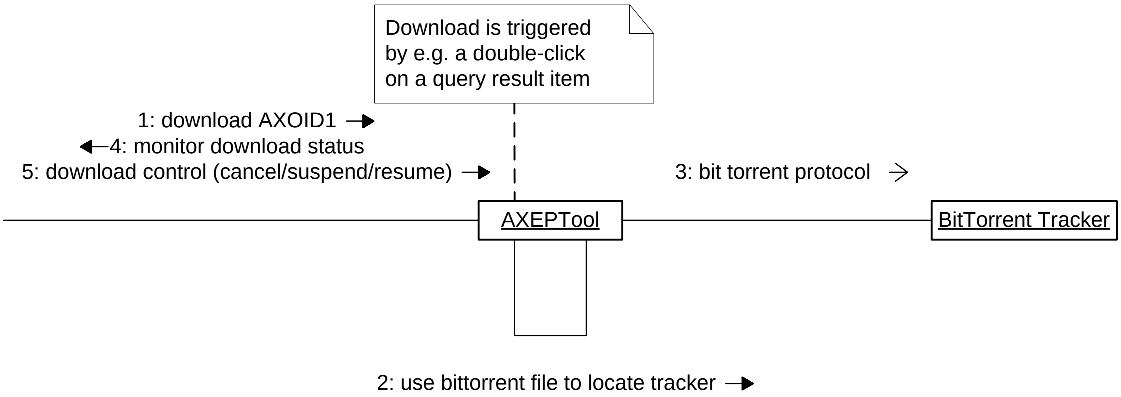
The other use cases are reported for two main reasons:

- The capability of controlling the AXEPTool (by a well-known WS interface) remotely. This action is particularly needed in B2B context since download processes can be managed in an automatic manner. This highlights that the downloading technology is reused, while it is accessed by a different way w.r.t. to the GUI.
- The role of the AXCP in completing the full P2P distribution life-cycle. A B2B user can actually create procedures for publish/unpublishing AXMEDIS objects by using scripting language. It can also “script” (i.e. program) what to import from the downloaded objects in the factory database for further usage.

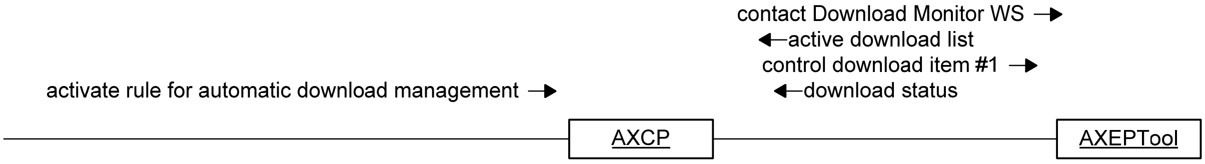
In the following basic collaboration diagram are reported in order to show AXEPTool P2P client functionalities for query and download.


1: query based on metadata →
←5: display result items
3: query based on metadata →
←3r: AXOIDs + bittorrent files
AXEPTool
AXMEDIS Query Service
Query Service response has to be processed for displaying matadata of the founded objects
2: contact Query Service WS →
←4: process Response

The first diagram shows the interaction between AXEPTool and AXMEDIS Query Service. The action is simply to redirect the user query to the Query Service collecting in the response information regarding the matching objects. The WS response is processed, since all the present metadata have to be attached to the query result items.
Please note that in the response also bittorrent metainfo files are returned.


Download is triggered by e.g. a double-click on a query result item
1: download AXOID1 →
←4: monitor download status
5: download control (cancel/suspend/resume) →
3: bit torrent protocol →
AXEPTool
BitTorrent Tracker
2: use bittorrent file to locate tracker →

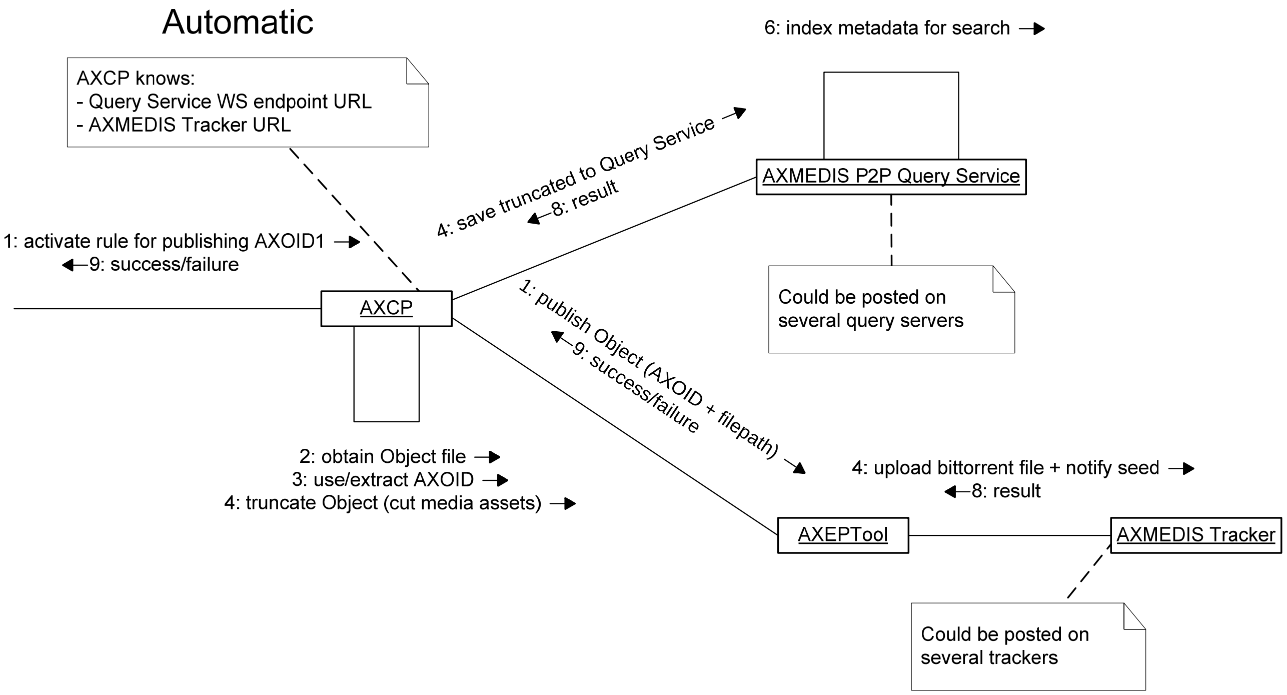
In the second diagram the typical action, being consequent to a query, has been presented.
In fact the user can decide of starting the download of an object. In this case the interaction is based on bittorrent protocol established among AXEPTool and the interested BitTorrent Tracker.


contact Download Monitor WS →
←active download list
control download item #1 →
←download status
activate rule for automatic download management →
AXCP
AXEPTool

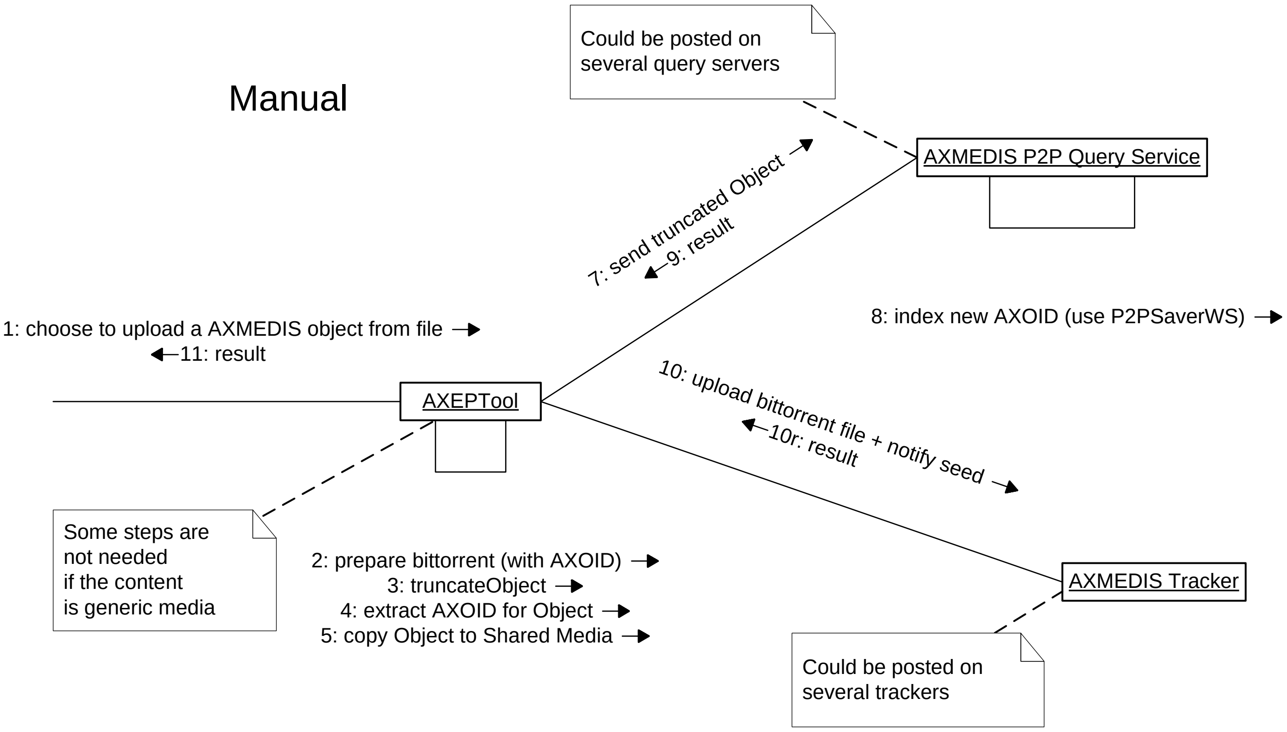
In the last diagram the AXCP is put as an intermediate in the AXEPTool control. In this case it allows the script function to operate as the user on the GUI (i.e. looking what is downloading and controlling those processes).

6.1.1 Usage walk-through for AXMEDIS Query Service and AXCP (EXITECH)

In this diagram a possible solution for avoiding posting of metadata by the B2B user is presented.
This solution can reuse the knowledge of the AXMEDIS infrastructure regarding an object metadata, since any object that can be distributed/published have been registered to AXCS (with metadata).
The solution of using the information located into the AXCS can be useful especially if the AXCS and the AXMEDIS P2P Query Service are located and maintained by the same organization. That can be AXMEDIS.ORG or by the channel distributor.


Automatic
AXCP knows:
- Query Service WS endpoint URL
- AXMEDIS Tracker URL
6: index metadata for search →
4: save truncated to Query Service →
←8: result
AXMEDIS P2P Query Service
1: activate rule for publishing AXOID1 →
←9: success/failure
AXCP
Could be posted on several query servers
1: publish Object (AXOID + filepath) →
←9: success/failure
2: obtain Object file →
3: use/extract AXOID →
4: truncate Object (cut media assets) →
4: upload bittorrent file + notify seed →
←8: result
AXEPTool
AXMEDIS Tracker
Could be posted on several trackers

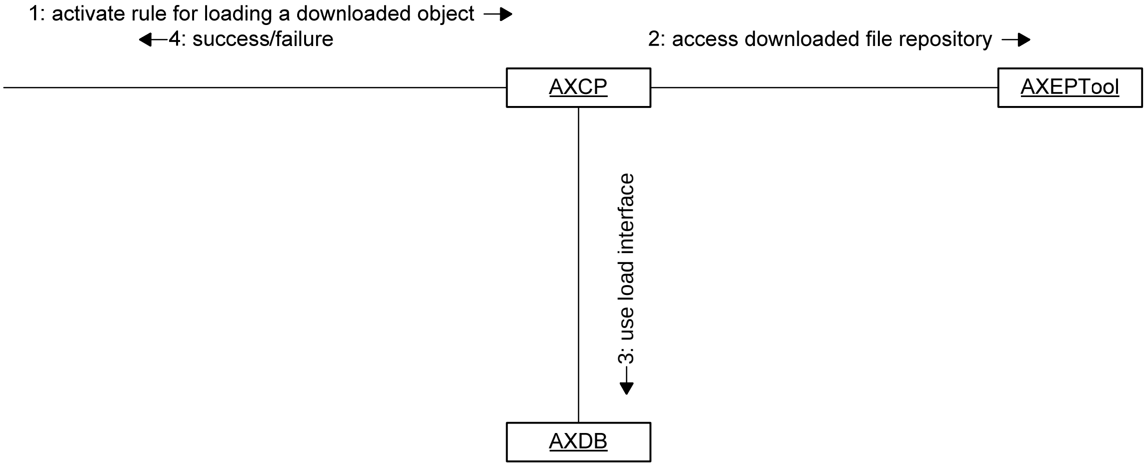
Please note that publishing task is totally carried out by AXCP and its functionalities of accessing WS. AXCP has also to compute bittorrent file.


Could be posted on several query servers
Manual
AXMEDIS P2P Query Service
7: send truncated Object →
←9: result
8: index new AXOID (use P2PSaverWS) →
1: choose to upload a AXMEDIS object from file →
←11: result
AXEPTool
10: upload bittorrent file + notify seed →
←10r: result
Some steps are not needed if the content is generic media
2: prepare bittorrent (with AXOID) →
3: truncateObject →
4: extract AXOID for Object →
5: copy Object to Shared Media →
AXMEDIS Tracker
Could be posted on several trackers

In the manual version some content manipulations which has been performed by AXCP, has to be done by AXEPTool, the AXOID has to be extracted for augmenting bittorrent, and the AXMT file (truncated object) has to be posted for future queries.

Another diagram has been produced to explain how import of newly downloaded object can be realized by only using AXCP functionality.

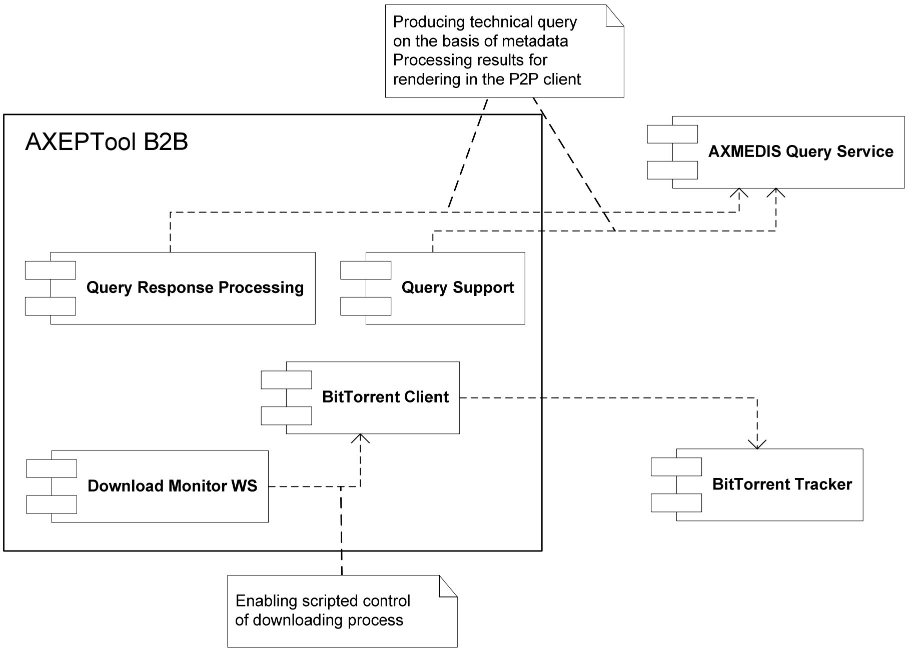
The latter has to access downloaded file repository of AXEPTool, and to use AXDB loading service.


1: activate rule for loading a downloaded object →
←4: success/failure
2: access downloaded file repository →
AXCP
AXEPTool
3: use load interface
AXDB

6.2 Note on the AXEPTool implementation

The following diagram put in evidence the main parts of the P2P B2B client solution. The relationships have been depicted for indicating major dependencies.

Please note the separation among query and download functionalities and the capability of remotely control (via WS) the P2P client. Additional technology is needed to integrate query/response mechanism in the AXEPTool.


Producing technical query on the basis of metadata Processing results for rendering in the P2P client
AXEPTool B2B
AXMEDIS Query Service
Query Response Processing
Query Support
BitTorrent Client
Download Monitor WS
BitTorrent Tracker
Enabling scripted control of downloading process

6.3 The AXMEDIA tool

AXMEDIA is a Java P2P BitTorrent Client Node for final users content sharing and B2C (Business to Consumer) content distribution. Internet distributors may use the P2P Network to reduce costs for scaling the

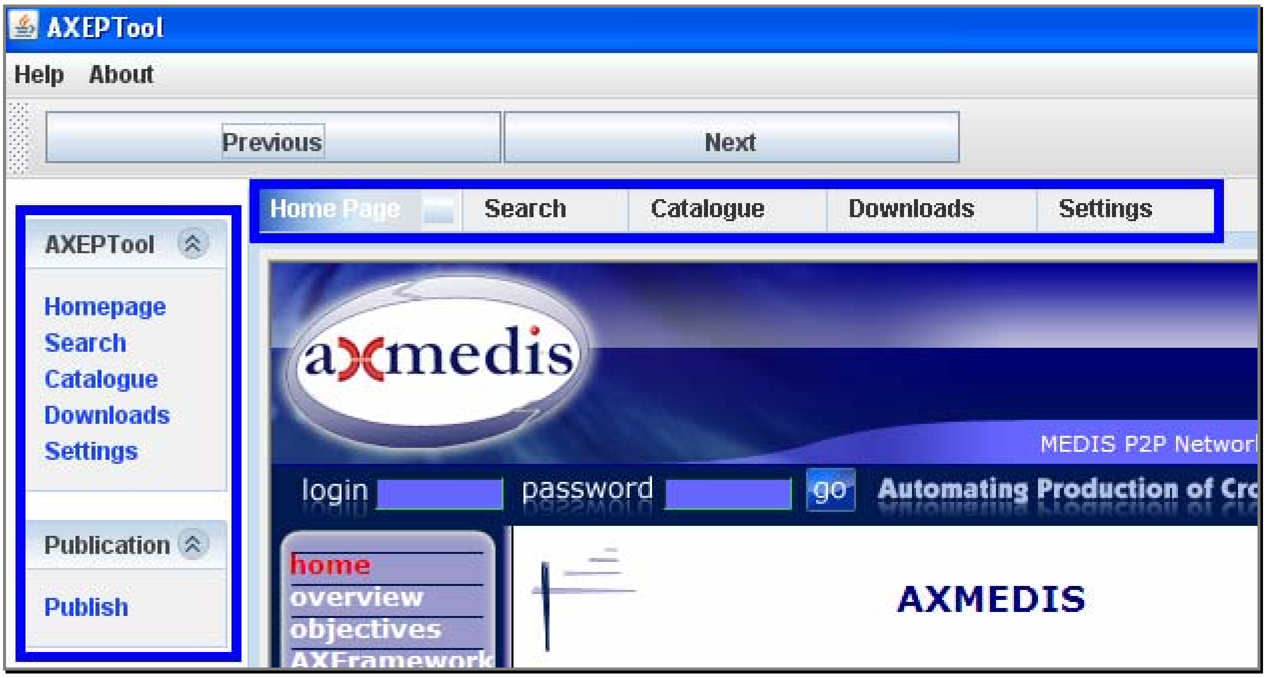
infrastructure for content distribution. AXMEDIA does not include some of the AXEPTool's features regarding publication, downloading and monitoring of the activities. Thus, it is easy to use, lighter to be executed and more easily acceptable by final users.
AXMEDIA does not include an integrated WebService and can manage a less number of connections (4 up/4 down) than the AXEPTool (50 up/10 down). AMXEDIA doesn't provide a settings page and doesn't include a Web Service server.
It is capable to access to the AXMEDIS P2P Query Service Server by means of a Web Service client.
Compared to the AXEPTool, AXMEDIA presents:

- a nicer user interface
- a simplified query user interface
- a simplified monitoring tool and user interface

6.3.1 Adaptation of the AXEPTool for B2C environment

In this section an AXETPool/AXMedia feature comparison table is presented to put in evidence the changes applied to the AXEPTool to obtain the AXMedia tool.

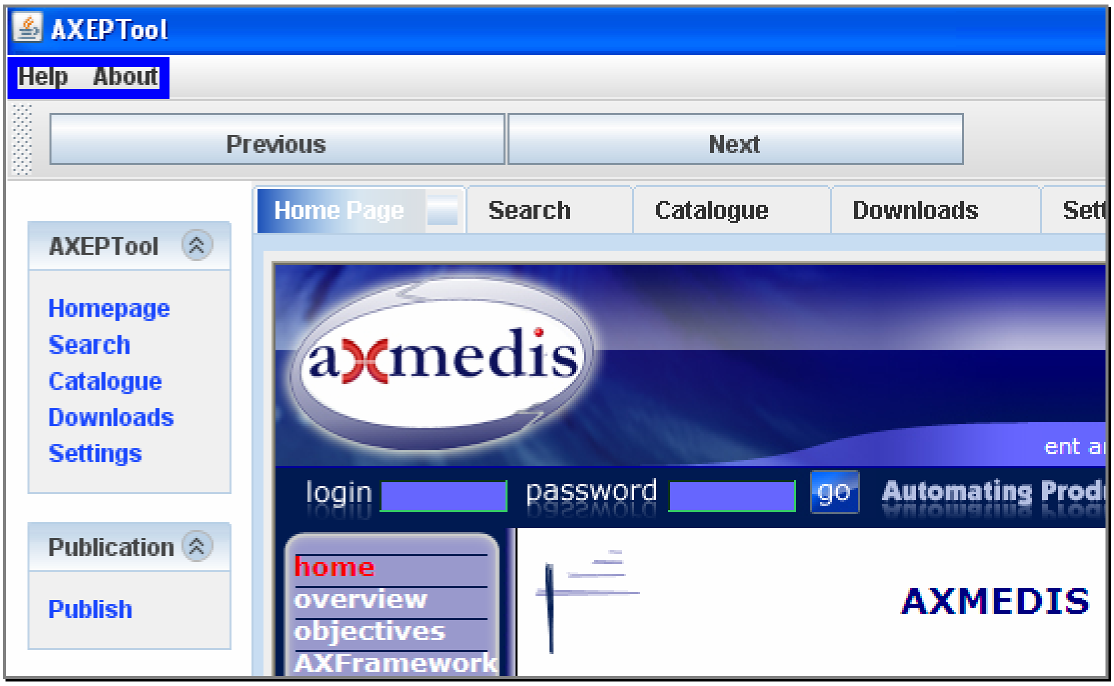
Menu on left and windows tabs:



ID	AXEPTool Feature	AXMEDIA Feature
1	Menu on left with *Homepage*, it leads to open the Home Page Tab, reporting the AXMEDIS home page as defined into the configuration file	Same as in the AXEPTool
2	Menu on left with *Search*, it leads to open the Search Tab, reporting the Query Portal page as defined into the configuration file	Same as in the AXEPTool
3	Menu on left with *Catalogue*, it leads to open the Catalogue Tab, reporting the AXTrack Catalogue page as defined in the configuration file	Same as in the AXEPTool, it is probable that this feature is too complex and not simple to be understood by the users.
4	Menu on left with *Downloads*, it leads to open the Downloads Tab, reporting the objects' transfer list	Same as in the AXEPTool

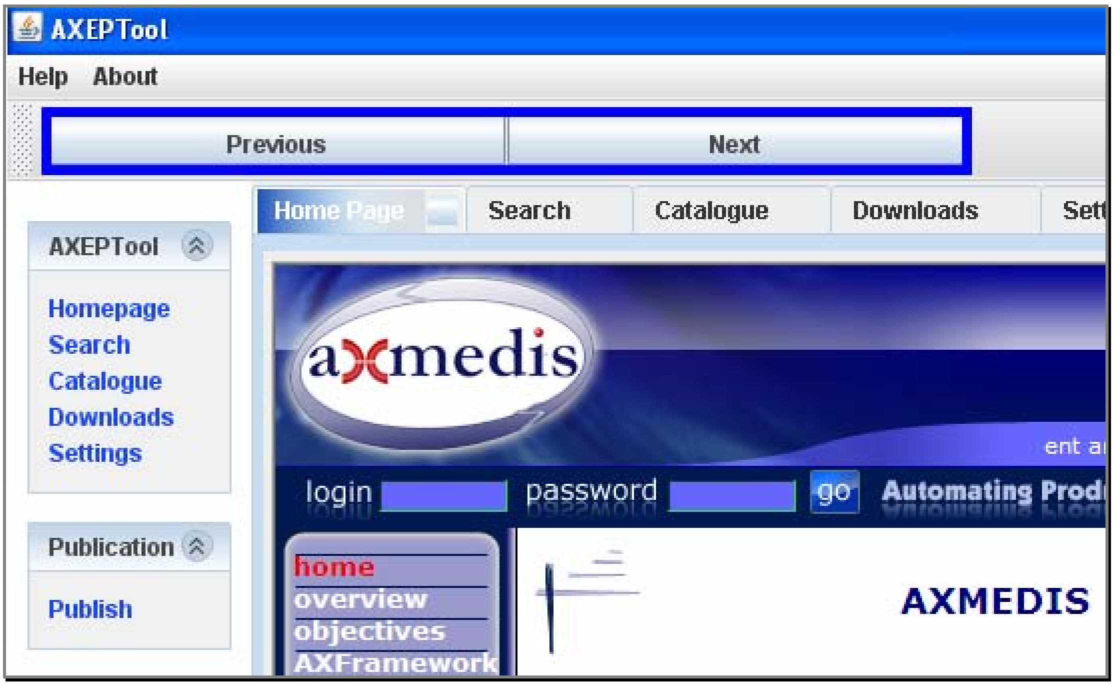
5	Menu on left with *Settings*, it leads to open the Settings Tab, that allows to edit the Settings reported in the configuration file too	Feature not requested in AXMEDIA, the form to change the setting has a sense only if some setting are still needed in AXMEDIA tool
6	Menu on left with *Publish*, it allows to select and publish an object through a file selector dialog window	Same as in the AXEPTool, feature not requested in AXMEDIA

Menu on top:



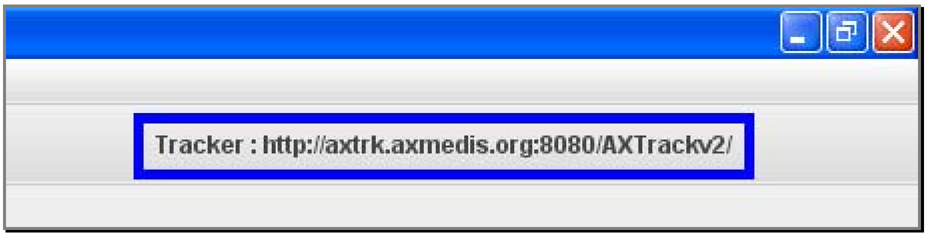
ID	AXEPTool Feature	AXMEDIA Feature
7	Menu on top with *Help*, it leads to open a temporary popup which reports the AXMEDIS portal link:	Same as in the AXEPTool but the link has to be clickable
8	Menu on top with *About*, it leads to open a temporary popup which reports the AXMEDIS portal link and the current application version.	Same as in the AXEPTool but the link has to be clickable

Browsing buttons:



ID	AXEPTool Feature	AXMEDIA Feature
9	Browsing Left Button *Previous* under top menu, it backs to the previous cached page if clicked when the Home Page tab is selected.	Same as in the AXEPTool but the button has to work with the Search page too
10	Browsing Right Button *Next* under top menu, it forwards to the previous cached page if clicked when the Home Page tab is selected	Same as in the AXEPTool but the button has to work with the Search page too

Tracker URL:


Tracker : http://axtrk.axmedis.org:8080/AXTrackv2/

ID	AXEPTool Feature	AXMEDIA Feature
11	Tracker URL reported on the top right of the AXEPTool window	Feature not requested in AXMEDIA. This means that the AXMEDIA does not have to present the tracker url

Search Page:

ID	AXEPTool Feature	AXMEDIA Feature
12	It shows the Query support portal page which allows to make queries for objects metadata	Same as in the AXEPTool

Catalog Page:

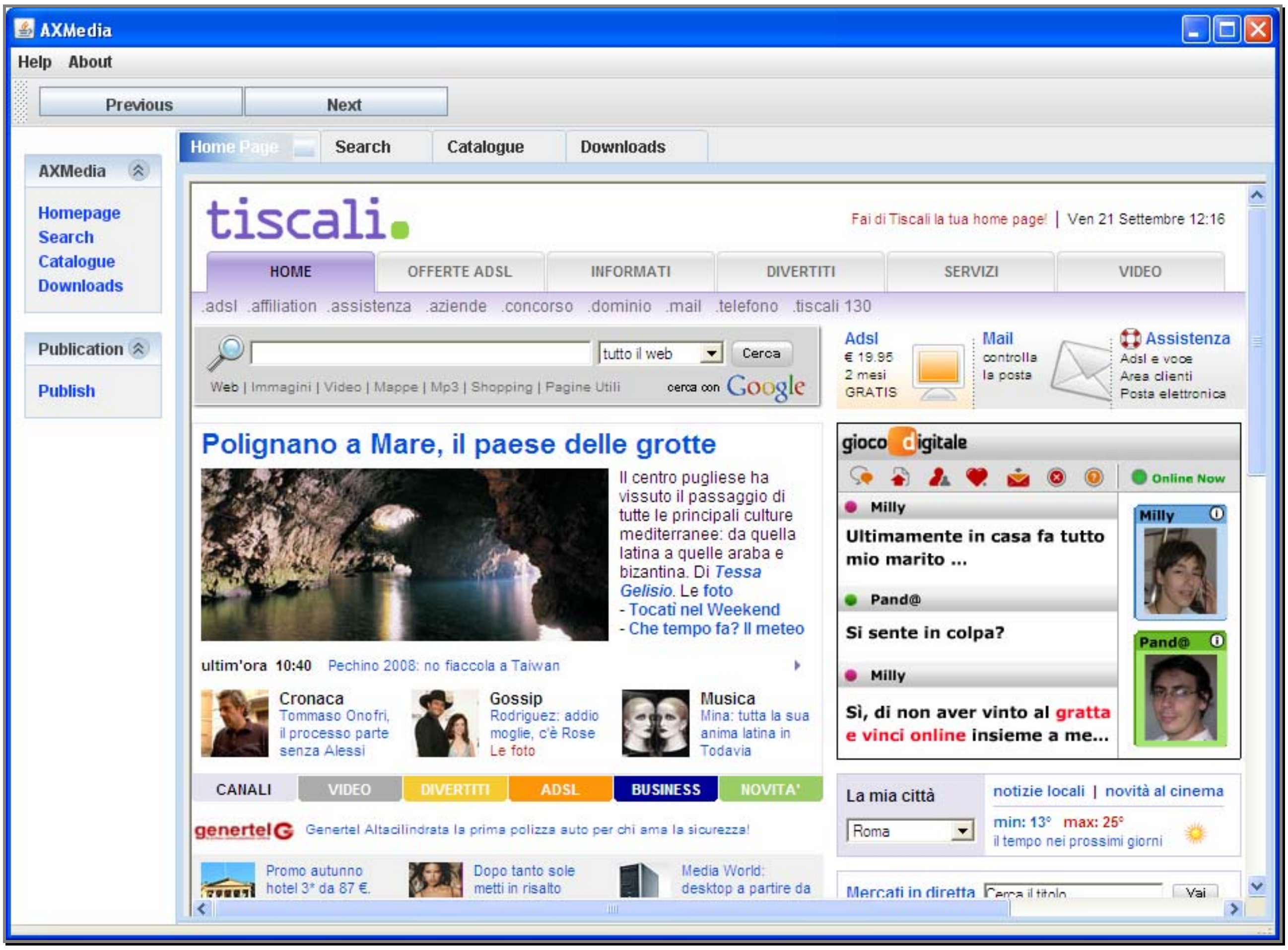
ID	AXEPTool Feature	AXMEDIA Feature
13	It shows the list of AXMEDIS and NON AXMEDIS objects published on the AXTrack, each of them with its file name, file size and AXOID (if present)	Same as in the AXEPTool but it has to list only AXMEDIS objects and doesn't have to report the AXOID field, probably the access to catalogue is not needed.

6.3.2 AXMEDIA GUI

From the Menu Help button it is possible to access to the AXMEDIS portal and from the About button the software version is showed. In both cases the user can access to the AXMEDIS web portal (http://www.axmedis.org) through a clickable hyperlink.

From the AXMedia left menu, the user is able to perform these actions:

- browse the AXMEDIS portal Home Page, thanks to the AXMedia integrated browser
- search for shared objects in the P2P network by using the AXMEDIS P2P Query Portal
- search for shared objects in the P2P network with by accessing the AXTrack's catalogue page
- start/stop downloads from the Download page
- publish objects in the P2P network



Polignano a Mare, il paese delle grotte

Il centro pugliese ha vissuto il passaggio di tutte le principali culture mediterranee: da quella latina a quelle araba e bizantina. Di Tessa Gelisio. Le foto

- Tocati nel Weekend

- Che tempo fa? Il meteo

ultim'ora 10:40 Pechino 2008: no fiaccola a Taiwan

Cronaca Tommaso Onofri, il processo parte senza Alessi

Gossip Rodriguez: addio moglie, c'è Rose Le foto

Musica Mina: tutta la sua anima latina in Todavia

Milly Ultimamente in casa fa tutto mio marito ...

Pand@ Si sente in colpa?

Milly Sì, di non aver vinto al gratta e vinci online insieme a me...

genertel Genertel Altacilindrata la prima polizza auto per chi ama la sicurezza!

Promo autunno hotel 3* da 87 €.

Dopo tanto sole metti in risalto

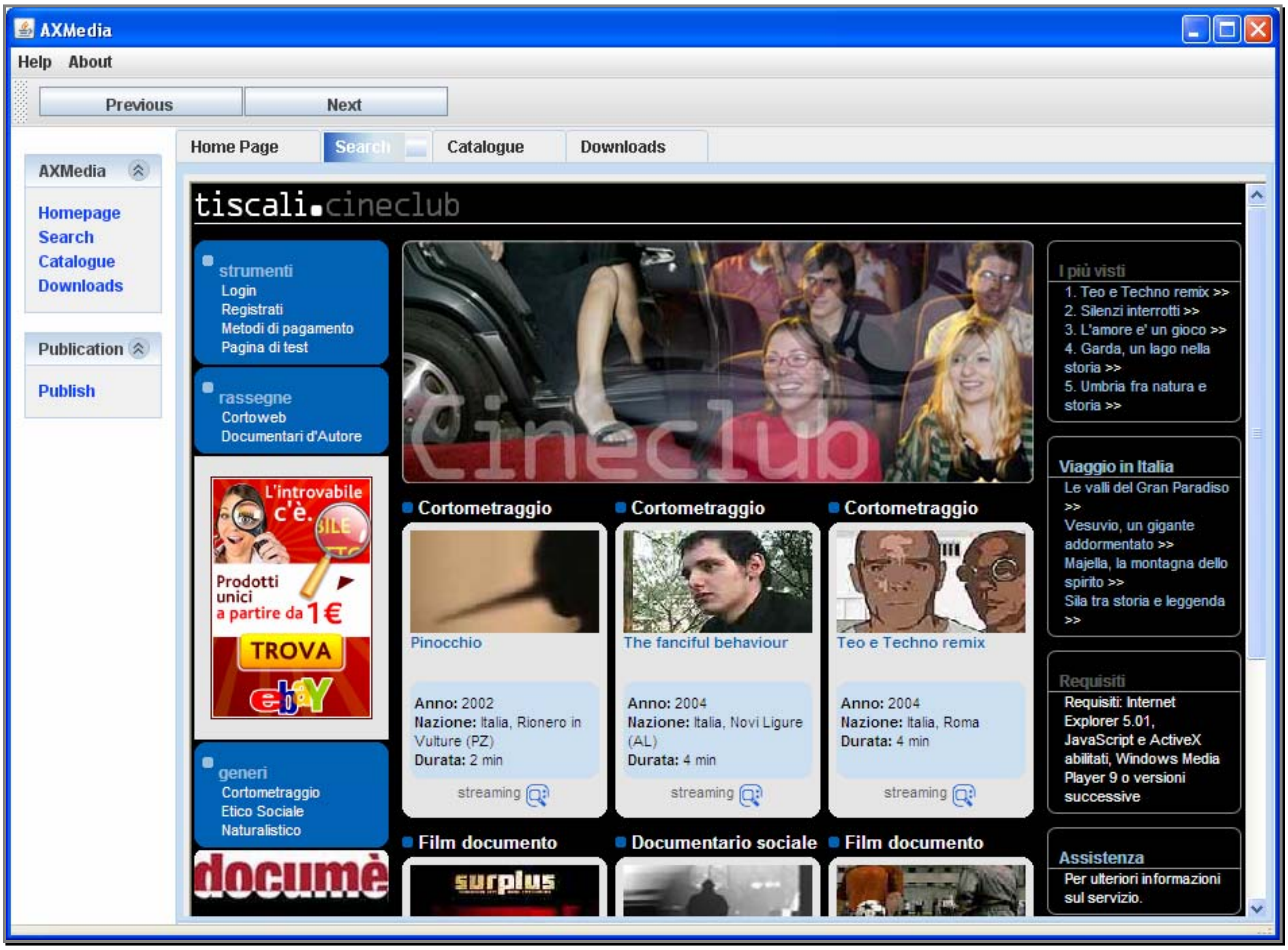
Media World: desktop a partire da

The tool bar allows the user to perform actions like those you can find in a web browser:

- Go to Previous page
- Go to Next page

In the Home Page tab the user can access the AXMEDIS portal homepage (http//www.axmedis.org).

In the Search tab the user can access to the P2P Query portal to search for objects, retrieve their metadata and start downloading.



Cortometraggio
Pinocchio
Anno: 2002
Nazione: Italia, Rionero in Vulture (PZ)
Durata: 2 min
streaming

Cortometraggio
The fanciful behaviour
Anno: 2004
Nazione: Italia, Novi Ligure (AL)
Durata: 4 min
streaming

Cortometraggio
Teo e Techno remix
Anno: 2004
Nazione: Italia, Roma
Durata: 4 min
streaming

Film documento

Documentario sociale

Film documento

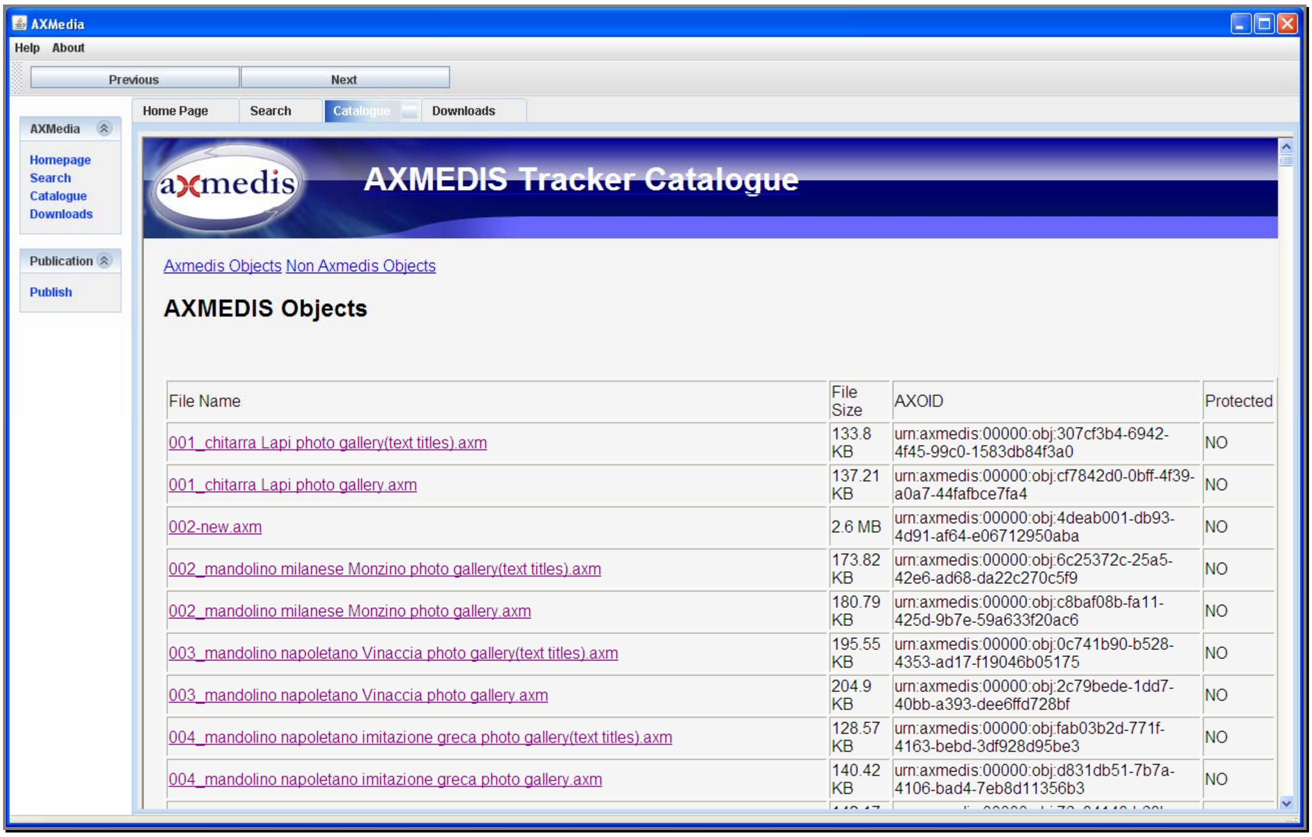
I più visti
1. Teo e Techno remix >>
2. Silenzi interrotti >>
3. L'amore e' un gioco >>
4. Garda, un lago nella storia >>
5. Umbria fra natura e storia >>

Viaggio in Italia
Le valli del Gran Paradiso >>
Vesuvio, un gigante addormentato >>
Majella, la montagna dello spirito >>
Sila tra storia e leggenda >>

Requisiti
Requisiti: Internet Explorer 5.01, JavaScript e ActiveX abilitati, Windows Media Player 9 o versioni successive

Assistenza
Per ulteriori informazioni sul servizio.

In the Catalogue tab the user can access to the AXTrack's P2P catalogue and start downloading the shared objects. The AXTrack's catalogue presents the same aspect than in the AXEPTool.



AXMEDIS Tracker Catalogue

Axmedis Objects Non Axmedis Objects

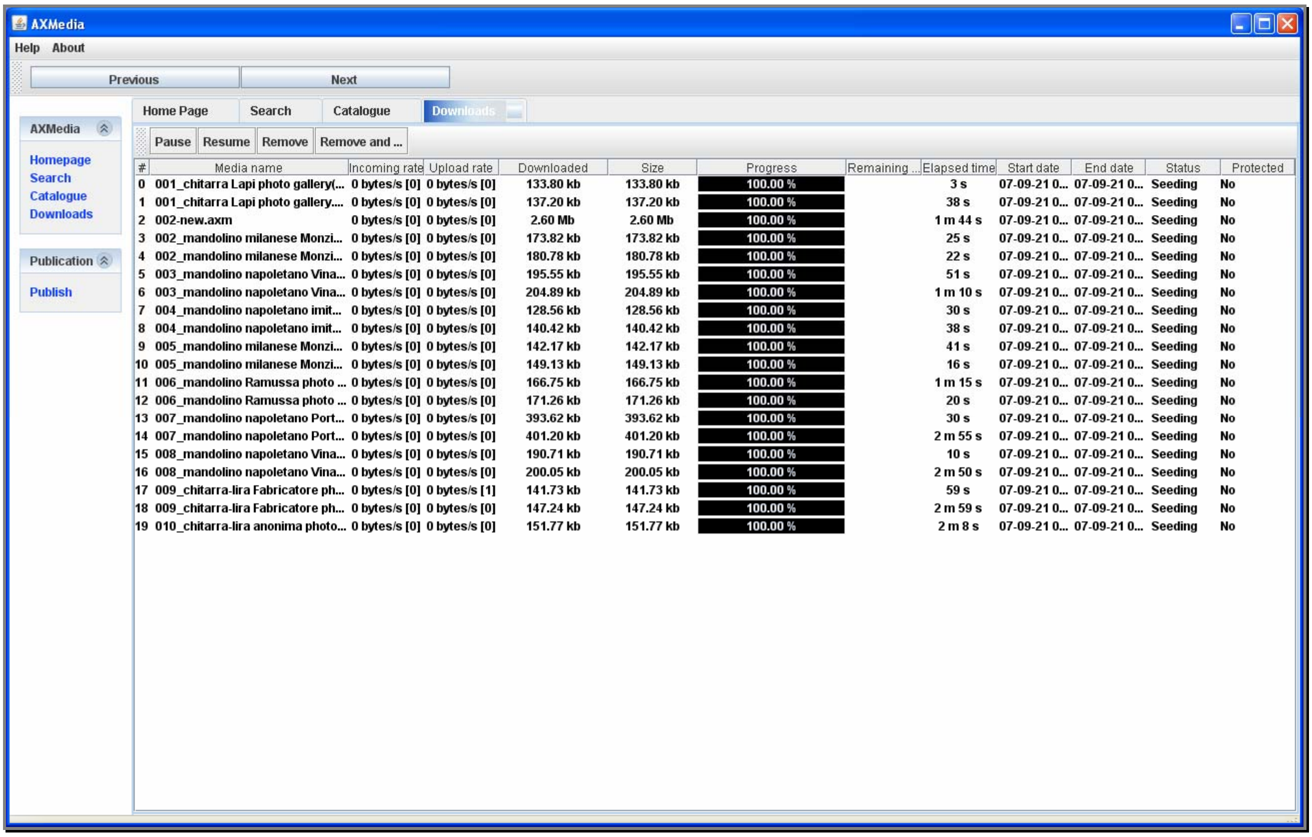
AXMEDIS Objects

File Name	File Size	AXOID	Protected
001_chitarra Lapi photo gallery(text titles).axm	133.8 KB	urn:axmedis:00000:obj:307cf3b4-6942-4f45-99c0-1583db84f3a0	NO
001_chitarra Lapi photo gallery.axm	137.21 KB	urn:axmedis:00000:obj:cf7842d0-0bff-4f39-a0a7-44fafbce7fa4	NO
002-new.axm	2.6 MB	urn:axmedis:00000:obj:4deab001-db93-4d91-af64-e06712950aba	NO
002_mandolino milanese Monzino photo gallery(text titles).axm	173.82 KB	urn:axmedis:00000:obj:6c25372c-25a5-42e6-ad68-da22c270c5f9	NO
002_mandolino milanese Monzino photo gallery.axm	180.79 KB	urn:axmedis:00000:obj:c8baf08b-fa11-425d-9b7e-59a633f20ac6	NO
003_mandolino napoletano Vinaccia photo gallery(text titles).axm	195.55 KB	urn:axmedis:00000:obj:0c741b90-b528-4353-ad17-f19046b05175	NO
003_mandolino napoletano Vinaccia photo gallery.axm	204.9 KB	urn:axmedis:00000:obj:2c79bede-1dd7-40bb-a393-dee6ffd728bf	NO
004_mandolino napoletano imitazione greca photo gallery(text titles).axm	128.57 KB	urn:axmedis:00000:obj:fab03b2d-771f-4163-bebd-3df928d95be3	NO
004_mandolino napoletano imitazione greca photo gallery.axm	140.42 KB	urn:axmedis:00000:obj:d831db51-7b7a-4106-bad4-7eb8d11356b3	NO

In the Downloads tab the user will get information on:

- number of the downloaded/downloading/published objects
- name of the downloaded/downloading/published objects
- download rate
- upload rate
- downloaded bytes
- object size
- percentage of the downloaded bytes
- remaining time to complete the download
- elapsed time since the download has started.
- start download/publish date
- end download/publish date
- object status (seeding, queued, error)
- protected flag (showing if an AXMEDIS object is protected or not)

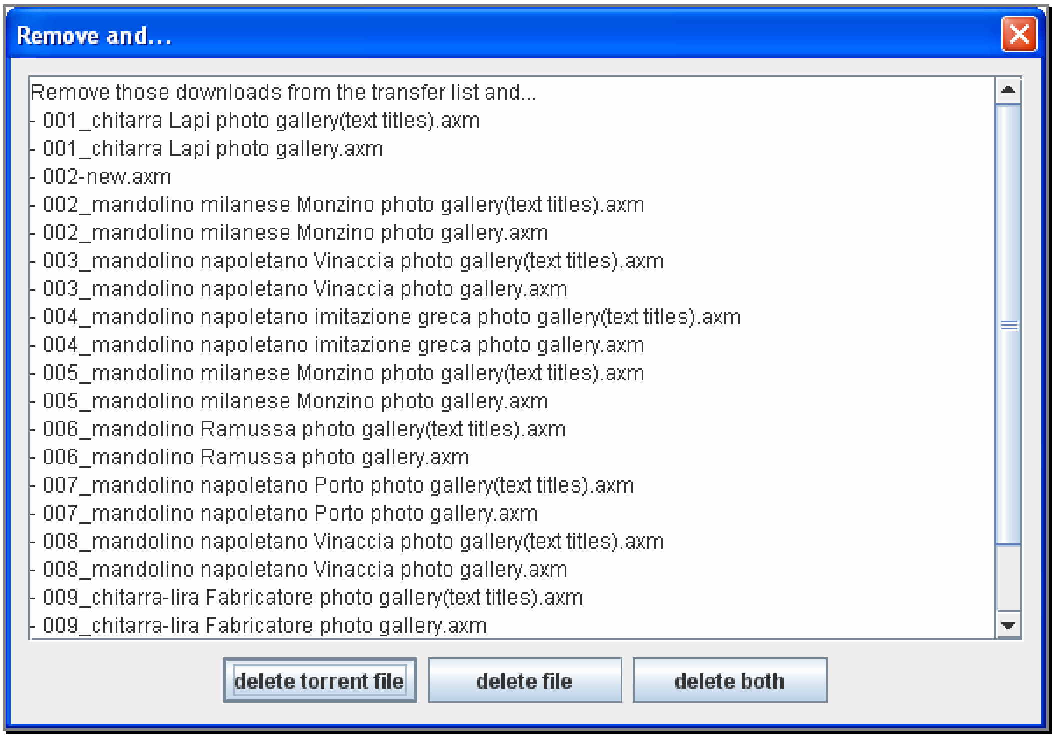
The Downloads tab doesn't include an AXOID field as you can see in the AXEPTool, since this is not necessary for the end user to access this information.



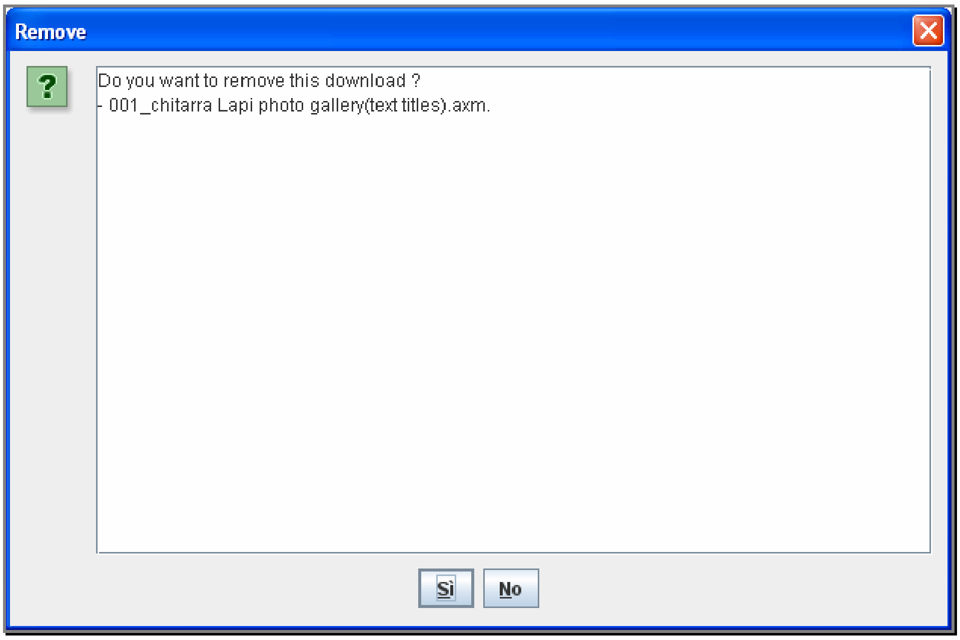
#	Media name	Incoming rate	Upload rate	Downloaded	Size	Progress	Remaining ...	Elapsed time	Start date	End date	Status	Protected
0	001_chitarra Lapi photo gallery(...	0 bytes/s [0]	0 bytes/s [0]	133.80 kb	133.80 kb	100.00 %		3 s	07-09-21 0...	07-09-21 0...	Seeding	No
1	001_chitarra Lapi photo gallery....	0 bytes/s [0]	0 bytes/s [0]	137.20 kb	137.20 kb	100.00 %		38 s	07-09-21 0...	07-09-21 0...	Seeding	No
2	002-new.axm	0 bytes/s [0]	0 bytes/s [0]	2.60 Mb	2.60 Mb	100.00 %		1 m 44 s	07-09-21 0...	07-09-21 0...	Seeding	No
3	002_mandolino milanese Monzi...	0 bytes/s [0]	0 bytes/s [0]	173.82 kb	173.82 kb	100.00 %		25 s	07-09-21 0...	07-09-21 0...	Seeding	No
4	002_mandolino milanese Monzi...	0 bytes/s [0]	0 bytes/s [0]	180.78 kb	180.78 kb	100.00 %		22 s	07-09-21 0...	07-09-21 0...	Seeding	No
5	003_mandolino napoletano Vina...	0 bytes/s [0]	0 bytes/s [0]	195.55 kb	195.55 kb	100.00 %		51 s	07-09-21 0...	07-09-21 0...	Seeding	No
6	003_mandolino napoletano Vina...	0 bytes/s [0]	0 bytes/s [0]	204.89 kb	204.89 kb	100.00 %		1 m 10 s	07-09-21 0...	07-09-21 0...	Seeding	No
7	004_mandolino napoletano imit...	0 bytes/s [0]	0 bytes/s [0]	128.56 kb	128.56 kb	100.00 %		30 s	07-09-21 0...	07-09-21 0...	Seeding	No
8	004_mandolino napoletano imit...	0 bytes/s [0]	0 bytes/s [0]	140.42 kb	140.42 kb	100.00 %		38 s	07-09-21 0...	07-09-21 0...	Seeding	No
9	005_mandolino milanese Monzi...	0 bytes/s [0]	0 bytes/s [0]	142.17 kb	142.17 kb	100.00 %		41 s	07-09-21 0...	07-09-21 0...	Seeding	No
10	005_mandolino milanese Monzi...	0 bytes/s [0]	0 bytes/s [0]	149.13 kb	149.13 kb	100.00 %		16 s	07-09-21 0...	07-09-21 0...	Seeding	No
11	006_mandolino Ramussa photo ...	0 bytes/s [0]	0 bytes/s [0]	166.75 kb	166.75 kb	100.00 %		1 m 15 s	07-09-21 0...	07-09-21 0...	Seeding	No
12	006_mandolino Ramussa photo ...	0 bytes/s [0]	0 bytes/s [0]	171.26 kb	171.26 kb	100.00 %		20 s	07-09-21 0...	07-09-21 0...	Seeding	No
13	007_mandolino napoletano Port...	0 bytes/s [0]	0 bytes/s [0]	393.62 kb	393.62 kb	100.00 %		30 s	07-09-21 0...	07-09-21 0...	Seeding	No
14	007_mandolino napoletano Port...	0 bytes/s [0]	0 bytes/s [0]	401.20 kb	401.20 kb	100.00 %		2 m 55 s	07-09-21 0...	07-09-21 0...	Seeding	No
15	008_mandolino napoletano Vina...	0 bytes/s [0]	0 bytes/s [0]	190.71 kb	190.71 kb	100.00 %		10 s	07-09-21 0...	07-09-21 0...	Seeding	No
16	008_mandolino napoletano Vina...	0 bytes/s [0]	0 bytes/s [0]	200.05 kb	200.05 kb	100.00 %		2 m 50 s	07-09-21 0...	07-09-21 0...	Seeding	No
17	009_chitarra-lira Fabricatore ph...	0 bytes/s [0]	0 bytes/s [1]	141.73 kb	141.73 kb	100.00 %		59 s	07-09-21 0...	07-09-21 0...	Seeding	No
18	009_chitarra-lira Fabricatore ph...	0 bytes/s [0]	0 bytes/s [0]	147.24 kb	147.24 kb	100.00 %		2 m 59 s	07-09-21 0...	07-09-21 0...	Seeding	No
19	010_chitarra-lira anonima photo...	0 bytes/s [0]	0 bytes/s [0]	151.77 kb	151.77 kb	100.00 %		2 m 8 s	07-09-21 0...	07-09-21 0...	Seeding	No

In the Downloads tab four buttons are available to perform various operations on the objects:

- Pause (pauses an object downloading)
- Resume (resumes an object downloading)
- Remove and…(deletes torrent file, delete file, delete both)
- Remove (removes an object from the transfer list without deleting it from the disk)



Remove those downloads from the transfer list and...
- 001_chitarra Lapi photo gallery(text titles).axm
- 001_chitarra Lapi photo gallery.axm
- 002-new.axm
- 002_mandolino milanese Monzino photo gallery(text titles).axm
- 002_mandolino milanese Monzino photo gallery.axm
- 003_mandolino napoletano Vinaccia photo gallery(text titles).axm
- 003_mandolino napoletano Vinaccia photo gallery.axm
- 004_mandolino napoletano imitazione greca photo gallery(text titles).axm
- 004_mandolino napoletano imitazione greca photo gallery.axm
- 005_mandolino milanese Monzino photo gallery(text titles).axm
- 005_mandolino milanese Monzino photo gallery.axm
- 006_mandolino Ramussa photo gallery(text titles).axm
- 006_mandolino Ramussa photo gallery.axm
- 007_mandolino napoletano Porto photo gallery(text titles).axm
- 007_mandolino napoletano Porto photo gallery.axm
- 008_mandolino napoletano Vinaccia photo gallery(text titles).axm
- 008_mandolino napoletano Vinaccia photo gallery.axm
- 009_chitarra-lira Fabricatore photo gallery(text titles).axm
- 009_chitarra-lira Fabricatore photo gallery.axm



Do you want to remove this download ?
- 001_chitarra Lapi photo gallery(text titles).axm.

The AXMEDIA tool doesn’t offer a Settings tab as you can see in the AXEPTool, since the end user doesn’t need to access advanced settings (proxy, AXTrack URL configuration, download directory, TCP port to use for bittorrent traffic, etc.).

6.3.3 Usability and performance issues

The model adopted for P2P Query after CRS4 abandoned the project is very different with respect to a distributed P2P query, since the current model has a query server over which all the objects are indexed and can be queried.

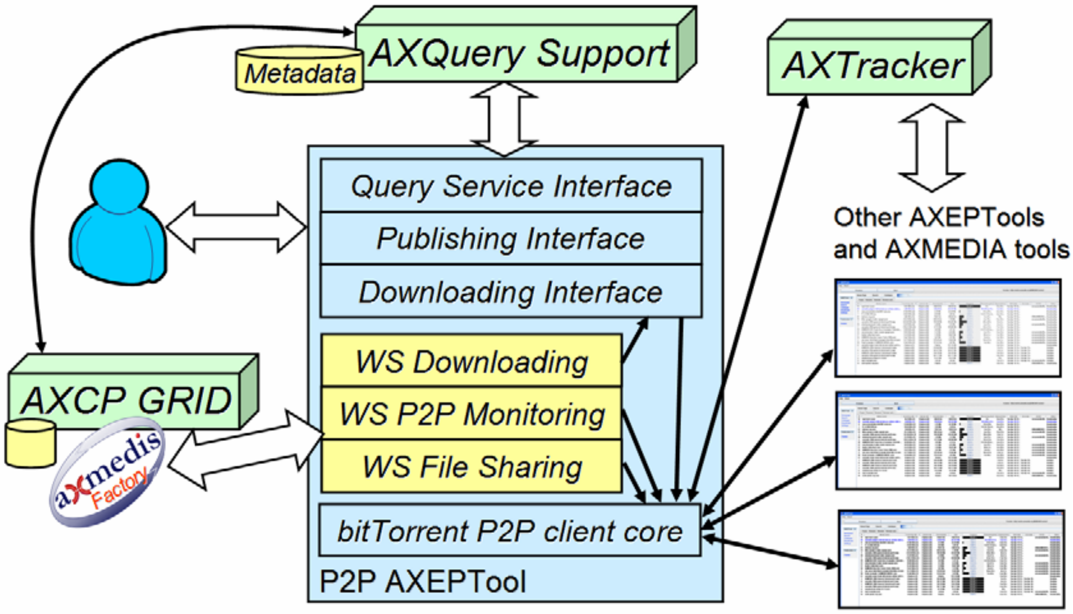
The centralization of this service originally designed for a factory can be a future critical problem in terms of query speed since if the node contains a huge large number of objects the query on the database can be slower than expected in a factory environment.

If this will be the case a solution for distributing the query among different query support will be studied considering both cluster solution for database (if database will be the bottleneck) and solution for distributing indexing and storing of object among different server leaving to the front-end server only the due of distributing and collecting results, while back-office activities are shared among several servers.

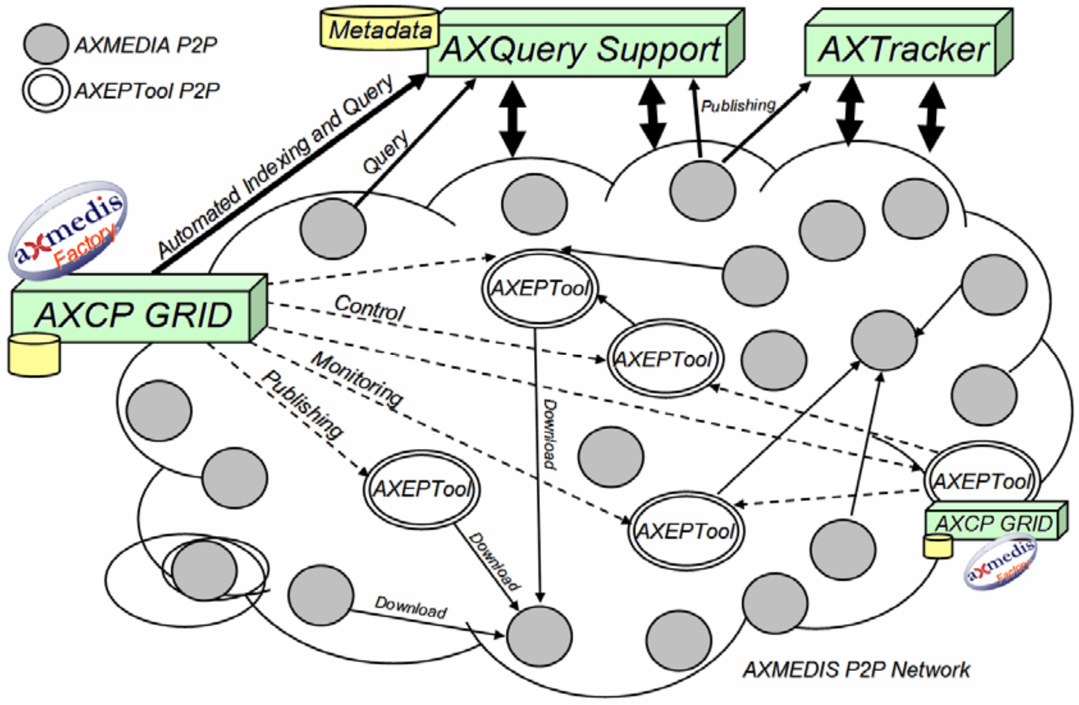
The architecture allows the implementation of both solutions.

6.3.4 Integration in the distribution chain

The AXMEDIS P2P solution in the Tiscali distribution chain is constituted by the elements depicted below:



New digital objects are published in the P2P network by a root node (AXEPTool at B2B), instructed to disseminate its shared contents periodically over the network through the use of the AXMEDIS AXCP, which interacts with the each AXEPTool integrated WebService. Check and monitoring of the whole network at B2B level is also possible and can be easily scheduled, through the use of javascript rules querying the set of P2P nodes to retrieve statistical and performance information. AXTrack can be queried through simplified javascript code to retrieve P2P catalogue's data and various bittorrent events (hosts connected to the tracker, tracker's traffic activity etc). AXQuery Support acts as the media club distribution portal to access all the digital contents metadata and put them in download.



On the basis of the basic functionalities of the AXMEDIS P2P network it is possible to create procedures/scripts for:

- Publishing an object on all the AXEPTools of the P2P Network (for example to accelerate the seeding of a given object on the network)

```
Axoid=getAXOID(MyFile);
publishing(MyAXEPTool, MyFile, TheAXTrack);
LA = listAXEPTool(TheAXTrack);
For each la of LA:  download(la, Axoid);
```

- Notifying the publication of new objects/files in the network among the different AXCP GRIDs/AXEPTools controlled

- Removing/deleting of objects/files from the network, at least from the AXEPTool nodes, AXTracks and AXQuery Support

```
Axoid=getAXOID(MyFile);
      axeptoolList=listAXEPTool(AXTrackURL);
      For each axeptool of axeptoolList:
               delete(axeptool, AXOID);
```

- Monitoring the status of the P2P Network, discovering which are the most active/virtuous AXEPTools, their capabilities, how many downloads have been performed, how many segments have been provided and for whose objects/files, if they are active, etc. This allows to perform specific analysis to assess the reputation of Business actors on the basis of their behavior on the corresponding AXEPTool

- Controlling the content seeded by the AXEPTools, for example constrained them to become an exact replica of each other (for an uniform seeding distribution), or imposing some distribution for the content in the network of the AXEPTools on the basis of the content distribution and statistical analysis

```
LA=listAXEPTool(AXTrackURL);
rootNode=LA[0];
List= listDownloaded(rootNode);
For each la of LA:
        For each list in List:
                 download(la, list);
```

- Activating automated queries for obtaining, downloading and posting these objects into the database of specific content collection on the basis of complex queries. So that, these active queries can be periodically activated to verify if some new content satisfies the criteria and, in the positive case, the automated download can be activated as well

- Activating automated publishing on the P2P Network of accessible collections from the AXCP GRID and crawling facilities etc

The solution has been obtained by exploiting and modifying the BitTorrent protocol and tools; integrating the P2P networking with those of the AXMEDIS Content Processing facilities, to allow setting up a large range of strategies for content distribution, seeding, publishing, promoting, etc. The main benefits of the proposed solution and architecture are scalability, content sharing control, efficient rules scheduling, fast object publishing and dissemination, and in general network controllability without constraining the actors of the network.

7 AXMEDIS clients for distribution on PC, PDA, Tablet PC (DSI, EPFL, SEJER, FUPF, TISCALI from M25)

7.1 State of the art about MPEG-21 clients (DSI, FUPF, EPFL)

As MPEG-21 ([1], [77]) is very recent and some parts of it are still under development, the state of the art in the domain of MPEG-21 terminal players is not very rich. Nevertheless we would like to signal two relevant examples in the following.

7.1.1 SMICL Viewer

The SMICL viewer is one of the first players/viewers developed for MPEG-21, developed at the University of Wollongong, Australia. This viewer supports a subset of the MPEG-21 Digital Item Declaration (DID). It is written in C++ using the wxWindows toolkit for portability. This player is able to read a DID and display it as a tree structure. When a user clicks on one component in the tree, these resources are displayed. The player can render images and text, but not yet for video.
The GUI provides a few menu options:

- Open a DIDL file
- Change configuration
- List IPMP tools
- Clear cache

IPMP tools are also partly implemented, as the viewer supports some IPMP messages.

7.1.2 MPEG-21 Terminal

The end user terminal complying with MPEG-21 specifications and supporting the Integrated Management System (IMS) concept developed in Enthrone, comprises several components related either to the management and processing of Multimedia Content and associated information (Digital Items, Metadata, Protection, and finally Media) or related to the Enthrone specific approach for monitoring and signaling Quality Of Service parameters to the IMS for performing actions to provide the user with the best possible service.

The actual implementation of the Enthrone MPEG-21 Terminal software is based on a set of modules with different levels of integration:

- The *Middleware module* covers the functionality of the Demultiplexer that is in charge of detecting which parts of the data flow from the network shall be delivered to the other components, namely: DID’s, Metadata, Video and Audio data (and possibly other media streams for Object and Scene Description in case of a complete MPEG-4 presentation), and finally IPMP data is routed to the IPMP module.
- The *DID Browser module,* is responsible for presenting the user with a readable version of the information carried by MPEG-21 compliant Digital Items Descriptors. The *Media Player module*, includes the basic functionality of decoding elementary streams for video and audio. In particular, the Media Player used in the Enthrone MPEG-21 Terminal supports MPEG-4 Video (IS 14496-2) and MPEG-4 Audio AAC (IS 14496-3) compliant streams.
- The *IMS module* covers the communication between the Terminal and the other components of the Integrated Management System.

- The *IPMP module* implements the functionality of “Intellectual Property Management and Protection” as defined in MPEG parlance, or “Digital Rights Management” in a more generic context. This module is tightly coupled with the Media Player, since the IPMP system is responsible for controlling the (authorized) usage of multimedia content in the Terminal, with control points that can be placed either at the input or at the output of the media decoders, depending on the type of IPMP tool used.

7.2 State of the art on other multimedia player terminals (EPFL)

Other multimedia tools, mainly based on proprietary formats, are currently spreading on the market. They are based on frameworks that are not as flexible and powerful as MPEG-21, however they constitute a state of the art in the domain, since as noticed above MPEG-21 is still in development stage and current tools are poor. Reviewing these frameworks is important as it provides a direct investigation about what currently means a state of the art player terminal in terms of media support, security, flexibility and other aspects; this is useful to learn about successful features of client terminal players for PC platforms and also to show how much advanced the MPEG-21 based AXMEDIS client will result.

We review here the main multimedia frameworks in their actual state. We do not speculate about their future, or about their possible evolutions. The landscape is likely to evolve quickly, some of the main actors will probably disappear, and new actors will also appear.

DRM aspects are not covered in detail, as a good review is already available in section 4 above (especially Real HELIX and others).

7.2.1 Flash

Flash is a proprietary system developed by Macromedia, Inc. [79]. It is composed of an authoring tool, and a viewer available for free on the most widely available platforms (Macintosh and Windows based). The viewer can be integrated in an HTML page, so Flash content can be easily integrated in Web content.

Flash is based on vector graphics. In conjunction with scripting (Actionscript), and animation (timeline based as well as scripting based), this has made of Flash the most widely used multimedia framework for the Web. The very low bit rate induced by the use of vector graphics, scripting, and timeline based animation has made of Flash a very convenient format for Web-based animation, storyboards, high quality graphics, and a suitable alternative to HTML for the development of Internet web sites.

Media support

The support of audio can be considered as being poor in Flash. The only compression scheme supported is mp3, together with a proprietary compression scheme known as “Nelly Mosser” for which no information is available to our knowledge. Flash supports also uncompressed schemes, PCM based, such as WAV or AIFF. For structured audio, no format – even MIDI - is supported. Video support in Flash has continued to evolve since its introduction in Flash MX and Flash Player 6. Flash Player 7 greatly improves video quality, supports higher frame rates, and provides additional opportunities for loading dynamic media at runtime.

System support

Flash supports the operating systems of Windows and MacOS. It cannot support the Unix, Linux, BSD system.

Authoring and production

Flash benefits from a proprietary authoring tool developed by Macromedia.

Interactivity and animation

Interactivity and animation can be implemented in Flash by using ActionScript, a proprietary scripting language.

Openness and extensibility

Flash is in principle a closed system on the client's side. No extensions can be developed, no decoders can be added, and no interactivity other than interactivity defined on the authoring side with the ActionScript scripting language can be defined to enhance the standard viewer (this is not to be confused with the extensibility functions available in the new Flash MX 2004, which are available in the Flash authoring application).

Flash is open to XML, and able to exploit XML data in a client-server architecture, via http-based protocol, or via XML socket based, real-time exploitation of data. With this functionality, it is for example possible to imagine a Flash client application exploiting XML data available on line, for example XML-based metadata such as RDF, or Dublin Core, or even MPEG-7 metadata in their XML format.

The Flash file format is itself now open, as well as some parts of the source code, and many developers are developing new Flash based solutions. For example, the NorthCode company [84] has developed SWFStudio (http://www.northcode.com/swfstudio/), a software which makes possible to build stand-alone executables from Flash content. In this configuration, it becomes possible to build plugins to Flash executables. The same society has developed a plugin development kit in order for other developers to build their own extensions to Flash (with the restriction that this works only with Flash stand alone applications – it is always impossible with the Flash standard client).

Security and privacy

Flash implements a browser-like security sandbox scheme in order to ensure the security and privacy of Flash movie and the client machine. The sandbox defines a limited space in which a Flash movie running within the Macromedia Flash Player is allowed to operate. Its primary purpose is to ensure the integrity and security of the client's machine, and as well as security of any Macromedia Flash movies running in the player. Basically, the sandbox idea is the following: A Macromedia Flash movie executes inside a sandbox. Any information inside the sandbox can be communicated only to the domain from which the movie came. Access to information within the sandbox from outside of the sandbox is severely limited.

7.2.2 Windows Media

Being preinstalled with every version of Microsoft Windows sold, Windows Media Player [80] is becoming increasingly widespread on the web.

Media support

Windows Media Player supports most of media formats, but it does not support the RealNetworks content such as the .ra, .rm, .ram media file. It cannot support the QuickTime content like .mov, .qt format. As for the MPEG-4 (.mp4), Windows Media Player gives no support.

System support

Windows Media Player supports the operating systems of Windows and MacOS. It cannot support the UNIX, Linux, BSD system.

Authoring and production

Production of Windows Media content can be done in multiple ways: by the mean of Windows Media Encoder, or by the mean of the toolkits provided by Microsoft for this purpose. This toolkit can be accessed by the mean of the C++ language, the Visual Basic language, or even by the mean of an HTML interface.

Interactivity and animation

No support of interactivity – scripting, controls, etc. – is directly available in Windows Media.

Openness and extensibility

Customization of the Windows Media Player is possible by using the Software development Kit provided to this end by Microsoft. By using the SDK, it is possible to develop a customized end-user interface driving the Windows Media content, in any language supported by the Windows Media SDK (C++, Visual Basic, HTML, .net with C#, etc.).

Security and privacy

Windows media frameworks implement the Windows Media digital rights management (DRM) platform to protect and securely deliver a la carte and subscription content for playback on a computer, portable device, or network device. Windows Media DRM is comprised of multiple components including Windows Media DRM for Portable Devices and Windows Media DRM for Network Devices, as well as an updated Windows Media Rights Manager SDK. These components allow for the seamless flow of content to almost any device, offer the widest range of purchase and rental options for digital media, and ensure the security of premium content as it flows from device to device. Windows Media DRM works in the following five steps: 1.Packaging 2.Distribution 3.Establishing a license server 4.License acquisition 5.Playing the digital media file to secure content providers to protect their content and maintain control over the entire process of the media distribution.

7.2.3 QuickTime

Apple’s products

They are the following:

- QuickTime 7 Pro [81] enables H.264 video creation, audio and video capture, multi-channel audio creation and multiple files export. It is an easy-to-use tool for creating AAC audio files and 3G files for mobile viewing, editing videos and exporting movies.
- QuickTime MPEG-2 Playback Component provides QuickTime users with the ability to import and play back MPEG-2 content, including both multiplexed and non-multiplexed streams. It is suited for content creators with projects such as Professional content production and transcoding video content (from MPEG-2 video to MPEG-4 for example).
- QuickTime Broadcaster is a tool for producing live broadcast events.

Media support

QuickTime 7 Player supports a wide-range of industry-standard audio formats, including AIFF, WAV, MOV, mp3, MP4 (AAC only), CAF and AAC/ADTS. For structured audio, QuickTime supports MIDI. There is no support for audio effects or 3D audio. Multichannel audio is supported by QuickTime 7 up to 24 audio channels, enabling standard surround formats. QuickTime 7 supports H264.

System support

QuickTime supports the operating systems of Windows and MacOS. It cannot support the UNIX, Linux, BSD system.

Authoring and production

Adobe's Premiere or Macromedia Director can generate QuickTime content. There are also a number of production tools available, like FinalCut Pro for instance. Apple does not provide authoring tools, but software like Adobe's Premiere or Macromedia Director are able to produce QuickTime content, generally by the mean of a plug-in. Apple's QuickTime-related products (QuickTime Pro, QuickTime MPEG-2 Playback and QuickTime Broadcaster) are not literally authoring tools, but provide however a few creating, encoding and editing functionalities.

Interactivity, animation

No scripting language is available for defining interactivity, but interactivity can be defined by using the QuickTime Software Development Kit provided by Apple.

Openness, extensibility

Extensions to QuickTime can be defined on the user's side by using the QuickTime Software Development Kit provided by Apple. It provides interfaces in C or Java QuickTime content can be embedded in a web page, but only a restricted set of functions are available from scripting languages such as JavaScript, making QuickTime not very well suitable for development of interactive content on the Web. Timeline-based, raw graphics animation is provided by authoring tools such as Adobe Premiere or Macromedia Director.

Security and privacy: tbd

7.2.4 Real

RealNetworks media are limited to audio, from speech, mono-channel to surround, channel music, and video. There is no native support for interactivity, vector graphics, but the Real Player [82] supports the W3C's standard for synchronized multimedia SMIL, and thus interactivity, animation and support of vector graphics can be integrated this way.

Media support

Real plays every major media format Including, AVI, MP3, RealAudio, RealVideo, WAV Audio, and Windows Media. This feature makes it a very popular media player.

System support

Real not only supports the operating systems of Windows and MacOS, but also support the UNIX, Linux. But it cannot support BSD system.

Authoring and production

The Helix producer enables encoding of streaming media (audio, video), in the native Real formats, with different bit rates. The Helix producer cannot generate SMIL animations – but can generate the included media.

Interactivity, animation

Real Media does not include any native support for interactivity or animation, but these functionalities can be integrated in SMIL animations by integration of Flash or SVG content.

Metadata

There is no support for metadata in Real Media.

Openness and extensibility

Real Media does not include any native support for scripting or extensibility, but the SMIL support for these features must be taken in account.

RealPlayer

- Audio: The Real Player includes support for its audio proprietary format as well as for mp3 format. It is also possible to include MIDI content.
- Vector graphics, animation: Real Player supports the integration of Flash content (only Flash 3 and Flash 4), with some restrictions such as for audio content, which must be integrated using another channel (mp3, or rm). Interaction with Flash content is also supported, enabling in this manner capabilities of interactions with timeline from the user. It is for example possible to develop a simple user interface in Flash, composed of some buttons for playing, stopping, fast reviewing or forwarding an audio track, but this kind of interaction will be limited to interaction with the timeline.
- Real Player supports also integration of SVG.

Security

Real implements the Helix DRM platform for secure media content delivery over PC and non-PC devices, including mobile devices and home equipments. Helix DRM includes a set of products and services enabling business models through secure rights managed distribution of movies, music and other digital content. Helix DRM provides secure media packaging, license generation and content delivery to a trusted media player base across all major platforms to multiple devices. It extends the RealPlayer and Helix Platform open architecture to accommodate the incorporation of a wide range of rights management systems. It integrates into existing infrastructures and back-end systems, supporting a broad set of business models including purchase, rental, video on-demand, and subscription services.

The above media players are all developed for the multiple kinds of content files which cover over the range of document, images, audio, video, multimedia, etc. Here, we also want to enclose two famous audio players iTunes and Winamp with the purpose of learning their good features of Graphical User Interface which could be a good example for the development of Axmedis Player.

7.2.5 iTunes

iTunes [83] is also a digital media player developed by Apple Computer, for playing and organizing digital music and video files. Additionally, the program connects to the iTunes Music Store (sometimes referred to simply as "iTunes" or "iTMS") which allows users to purchase digital music files that can be played by iTunes. The player has gained and maintained a reputation as being easy to use while still providing many features for obtaining, organizing, and playing music. iTunes is also the principal way to manage the music on Apple's popular iPod digital audio player. The program is freely downloadable and is also supplied with Mac OS X as well as Apple's iLife home-application suite.

Media Support

iTunes can currently encode to MP3, AIFF, WAV, MPEG-4 AAC, and Apple Lossless, and can play anything QuickTime can play (even video formats, as long as they have audio). In order to play other formats such as the Ogg Vorbis audio format iTunes requires addition of QuickTime components. However, the extensions for Ogg Vorbis does not work with QuickTime version 7 and Mac OS Tiger installed. In May 2005, video support was introduced to iTunes with the release of iTunes 4.8. But Video support in iTunes is still limited at this point iTunes is so far still incompatible with the most common video formats such as .MPEG and .WMV.

System support

iTunes supports Windows and MacOS. It cannot support the UNIX, Linux and BSD system.

Authoring and production

iTunes was developed from SoundJam MP, a popular commercial MP3 application distributed by the Macintosh software company Casady & Greene.

Openness and extensibility

On the Macintosh, iTunes is tightly integrated with Apple's iWork suite of applications and the rest of the applications in iLife.

Security and privacy: tbd

7.2.6 Winamp

Winamp [85] is an audio player made by Nullsoft, part of Time Warner. It is skinnable, multi-format freeware. Originally, MP3 playback was based on the AMP decoding engine by Tomislav Uzelac et al. In later versions this was replaced with Nitrane, a proprietary decoder created by Nullsoft and subject of a lawsuit from Playmedia Systems, Ltd. After an out of court settlement and licensing agreement, Nullsoft switched to an ISO decoder from Fraunhofer Gesellschaft, the developers of the MP3 format

Media Support

Winamp support a wide range of audio format file including MIDI, MOD, MP3, Ogg Vorbis, WAV, WMA and many other audio formats.

System support

Winamp only supports Windows. It cannot support MacOS, UNIX, Linux, and BSD system.

Authoring and production

Winamp was first released by Justin Frankel in 1997.

Security and privacy: tbd

The features related to these above media players will be summarized in the following tables.

	Creator	First public release date	Stable version	Software license	Proprietary format
Flash	Macromedia	December 1996	8.0	Proprietary	Flash
Director	Macromedia	1988	MX 2004	Proprietary	DCR
iTunes	Apple Computer	January, 2001	4.9	Proprietary	Apple Lossless
QuickTime	Apple Computer	December, 1991	7.0.1	Proprietary	QuickTime
RealPlayer	RealNetworks	1995	10	Proprietary	RealAudio, RealVideo
Winamp	Nullsoft	June, 1997	5.094	Proprietary	NSV
Windows Media Player	Microsoft	November, 1992	10	Proprietary	WMA, WMV

Table 4 – General Features and Information

	Windows	Mac OS X	Linux	BSD	Unix
Flash	Yes	Yes	No	No	No
Director	Yes	Yes	No	No	No
iTunes	2000/XP/2003 only	Yes	No	No	No
QuickTime	Yes	Yes	No	No	No
RealPlayer	Yes	Yes	Yes	No	Yes
Winamp	Yes	No	No	No	No
Windows Media Player	Yes	Yes	Planned	No	No

Table 5 – Operating System support (on which the players can run natively without emulation)

	Audio playback	Video playback	Outbound streaming	Skinnable	Media Database
Flash	Yes	Yes	Yes	No	Yes
Director	Yes	Yes	Yes	No	Yes
iTunes	Yes	Yes	Yes	No	Yes
QuickTime	Yes	Yes	No	Partial	No
RealPlayer	Yes	Yes	No	Yes	Yes
Winamp	Yes	Yes	No	Yes	Yes
Windows Media Player	Yes	Yes	No	Yes	Yes

Table 6 – Main Features

	Lossy compression						Lossless compression		
	MP3	**WMA**	**RealAudio**	**Vorbis**	**AAC**	**AC3**	**APE**	**FLAC**	**ALAC**
Flash	Yes	Yes	No	No	No	No	No	No	No
Director	Yes	Yes	Yes	Yes	Yes	?	?	?	?
iTunes	Yes	Partial	No	No	Yes	No	No	No	Yes
QuickTime	Yes	No	No	No	Yes	No	?	No	Yes
RealPlayer	Yes	Yes	Yes	Yes	Yes	?	?	?	?
Winamp	Yes	Yes	No	Yes	Yes	No	No	No	No
Windows Media Player	Yes	Yes	No	No	No	No	No	No	No

Table 7 – Audio Format support

	MPEG-1	MPEG-2	MPEG-4	WMV	RealVideo	Theora	Flash
Flash	Yes	Yes	No	Yes	No	No	Yes
Director	Yes	?	?	No	Yes	?	Yes
iTunes	Yes	No	Yes	No	No	No	Yes
QuickTime	Yes	No	Yes	No	No	No	Yes
RealPlayer	Yes	?	?	No	Yes	?	Yes
Winamp	Yes	No	No	Yes	No	No	No
Windows Media Player	Yes	Yes	No	Yes	No	No	Yes

Table 8 – Video Format support

	AVI	ASF	QuickTime	OGM	Matroska	MP4
Flash	Yes	Yes	No	No	No	No
Director	Yes	?	Yes	?	?	Yes
iTunes	Yes	?	Yes	?	?	Yes
QuickTime	Yes	No	Yes	No	?	Yes
RealPlayer	Yes	No	No	?	?	?
Winamp	Yes	No	No	No	No	Yes
Windows Media Player	Yes	Yes	No	No	No	No

Table 9 – Container Format support

7.3 Prototype description (DSI, EPFL)

Two prototype players have been developed.
The first one exploited the features of Microsoft Foundation Classes in order to rapidly develop a user friendly graphical interface able to open and display objects and the associated AXMEDIS hierarchy.
The second player is based on an open source SMIL player called AMBULANT which has been “embedded” into a WxWidgets application implementing the user interface. The implementation of this player permitted to appreciate the flexibility of the SMIL language for the creation of multimedia scenes composed by several synchronized media objects.
The following sections provide some details and screenshots of the developed applications.

7.3.1 AXMEDIS PC based Player (DSI)

Technical details

Reference to the AXFW location of the demonstrator	https://cvs.axmedis.org/repos/Applications/axactivex/doc/test
List of libraries used	MFC
References to other major components needed	AXMEDIS ActiveX
Problems not solved	
Configuration and execution context	
Programming language	Visual Basic (.NET)

The preliminary prototype of the AXMEDIS PC Player allows to:

- load an axmedis object
- browse the content in the object (only resources)
- hide/show the AXMEDIS hierarchy
- play/stop the content fruition (for audio/video)

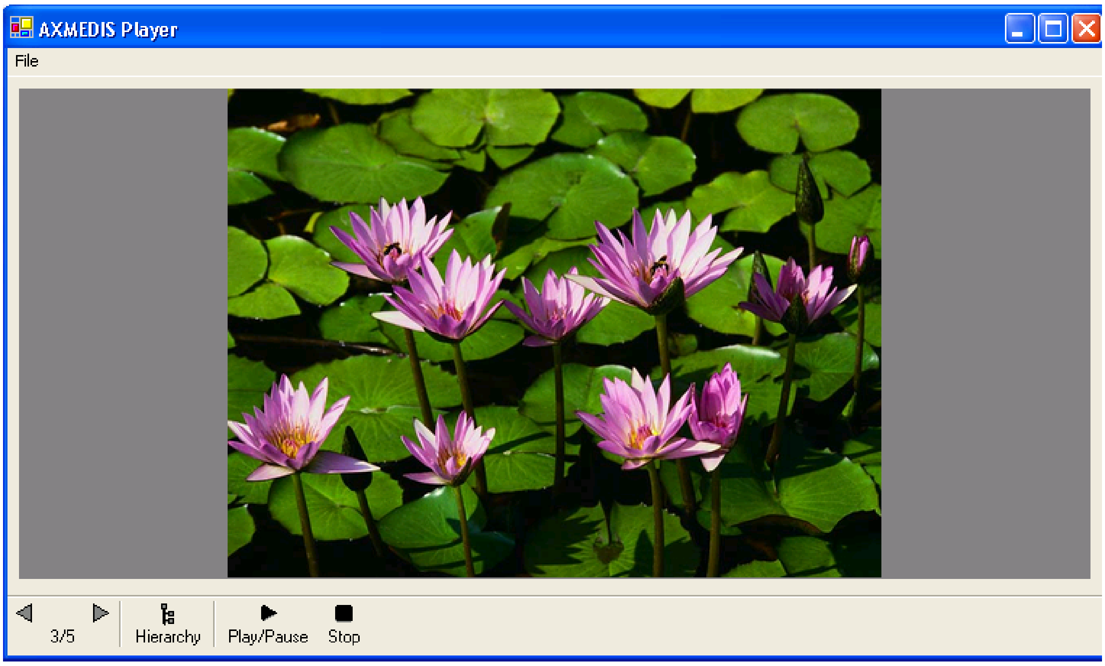


Figure 27 – AXMEDIS PC player rendering an ActiveX image

Note: in the figure above the ActiveX is only the image and not the buttons

When the Hierarchy button is pressed, the AXMEDIS Hierarchy is shown:

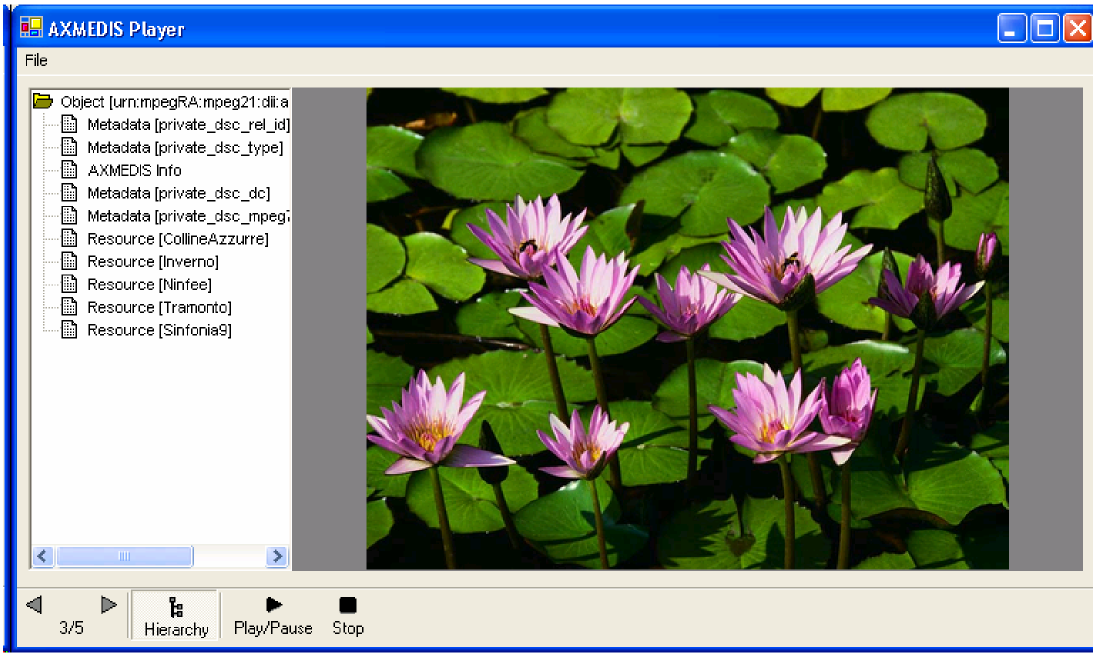

AXMEDIS Player
File
Object [urn:mpegRA:mpeg21:dii:a
Metadata [private_dsc_rel_id]
Metadata [private_dsc_type]
AXMEDIS Info
Metadata [private_dsc_dc]
Metadata [private_dsc_mpeg
Resource [CollineAzzurre]
Resource [Inverno]
Resource [Ninfee]
Resource [Tramonto]
Resource [Sinfonia9]
3/5
Hierarchy
Play/Pause
Stop

Figure 28 – AXMEDIS hierarchy shown with the resource

7.3.2 SMIL Player (EPFL)

Technical details

Reference to the AXFW location of the demonstrator	https://cvs.axmedis.org/repos/Applications/smil-player
List of libraries used	wxWidgets
References to other major components needed	AXOM
Problems not solved	Complete wxWidgets porting, support of main AXMEDIS components necessary for integration, internal media types support
Configuration and execution context	
Programming language	C++

SMIL is an XML language for choreographing multimedia presentations where audio, video, text and graphics are combined in real time. The language, the Synchronized Multimedia Integration Language (SMIL, pronounced, "smile") is written as an XML application and is currently a W3C Recommendation. Simply put, it enables authors to specify what should be presented when, enabling them to control the precise time that a sentence is spoken and make it coincide with the display of a given image appearing on the screen.

The SMIL player used in AXMEDIS will be based on the AMBULANT Player. The AMBULANT Open SMIL Player is an open-source, full SMIL 2.0 media player. It is intended for researchers and developers who want a source-code player upon which they can build higher-level systems solutions for authoring and content integration, or within which they can add new or extended support for networking and media transport components. The AMBULANT player may also be used as a complete, multi-platform media player for applications that do not need support for closed, proprietary media formats. The AMBULANT player written in C++, is distributed under a modified GPL license, and it is available for Windows, Linux, and Macintosh.

When the user wants to display the SMIL component, the Player will extract it from the AXMEDIS Object, uncompress it from binary to a media file and store it as a temporary file on the disk. The SMIL file is described in section 8.1.2 as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<smil>

  <head>
    <layout>
      <root-layout background-color="red" height="600" id="rootlayout"
title="axmedis" width="800"/>
      <region height="143" id="RegionName1" left="105" top="124" width="104"/>
      <region height="145" id="RegionName2" left="295" top="125" width="235"/>
      <region height="147" id="RegionName3" left="234" top="287"
width="357"/></layout>
  </head>

  <body>
<par>
    <audio begin="2" end="16" region="RegionName1" src="Reference1"/>
    <video begin="6" end="21" region="RegionName2" src="Reference7"/>
    <video begin="2" end="14" region="RegionName3" src="Reference5"/>
</par>
  </body>

</smil>
```

The src value refers to the resources contained in the AXMEDIS Object, the player extracts relevant media resources from the AXMEDIS Object, stores them as temporary media files on the disk. After this, the Player will replace the <body> part of the above file with these directories of temporary media files and returns them to a new temporary SMIL file with the <body> part as follows:

```
  <body>
    <audio begin="2" end="16" region="RegionName1" src="C:\Temp\Birthday.mp3"/>
    <video begin="6" end="21" region="RegionName2" src="C:\Temp\BPz3-g.mpg"/>
    <video begin="2" end="14" region="RegionName3" src="C:\Temp\Fz3s.mpg"/>
  </body>
```

However this is a temporary solution a way to get resources without saving them to disk have to be studied. It works but it consumes a lot of memory space. Considering the limited space in different clients, such as PDA or mobile phone, this may does not work.

How to play a SMIL component directly from AXMEDIS object?

This section attempts to explain the basic structure of the internal SMIL Player by loosely explaining what happens when you run it and play an AXMEDIS object. The main mechanism and process are based on those of AMBULANT SMIL player. We modify the section of generating data source to get it integrated with AXMEDIS editor. If you want to have a better knowledge of this mechanism, please check AMBULANT website and it will definitely help you to understand our method here.

1. Opening a SMIL component inside an AXMEDIS object

When the user selects Open (or double-clicks) we need to get the data of the SMIL component from AXMEDIS object. It generates the decompressed stream and gives it to the Player. There is an XML parser as a third party library to parse the stream into a DOM tree and create a player to play that DOM tree. In addition, we need to tell the player how it can obtain media data, create windows and more, because the media data are described as “AxIndex” and the content of media data are stored in the AXMEDIS object.

Most player implementations (the Windows player is an exception) have a class with a name like mainloop to handle this. Such a mainloop is created per SMIL component. This mainloop object will first create the various factories and populate them:

- A **window factory** is usually implemented by the main program itself. The player will call this when it needs a window. Usually the first request to create a window will actually return the document window (after resizing to the appropriate size).
- A **global playable factory** is created. This is the object the player will use to create renderers for the various media types. The global factory is filled with the various renderers this ambulant player supports. In effect, this is the step where you get to decide how various media are rendered.
- A **datasource factory** is created and filled with the factory functions that will create datasources for audio, video or other, raw, data such as text. The factory functions that are added to the datasource factory partially determine how data is retrieved over the net, which protocols and formats are supported and such. Partially, because some media items (audio and video, notably) may be rendered by simply passing the URL to some underlying media infrastructure such as DirectX or QuickTime. Note, **since we will directly retrieve the resource through "AxIndex" instead of using original URL, we will modify the code here. i.e., to generate the data source or data stream from "AxIndex".**

Next the factories are put together in a factories struct, and if the architecture supports dynamically loadable plugins we get the plugin engine singleton object and ask it to load the plugins. This will search the plugin directories for dynamic objects with the correct naming convention, load them, and call their initialize routine. The factories object is passed to the initialize routine, so the plugin itself can register any factories it wants.

The next step is to create the DOM tree.
One way to do this is to use to "axom::getAxObjectElement(*resourceIndex)" to read the SMIL component from the AXMEDIS object, and then pass this data to document::create from string. This will return a document object. This object contains the DOM tree itself (implemented by the node object) and some context information. **Note: In AMBULANT SMIL Player, The context information are XML namespace information, original URL for resolving relative URLs used in the document, a mapping from XML IDs to node objects. Since we will directly retrieve the resource through "AxIndex" instead of using original URL and resolved relative URLs, we have to modify the code here.**

The final step is to create a player object. This is done through create smil2 player, passing the document, the factories and one final object, embedder. This object is again implemented by the main program, and implements a small number of auxiliary functions, such as opening an external webbrowser or opening a new SMIL component.

2. Creating the player (same as AMBULANT)
When the smil player object is created it gets the document, factories and embedder arguments. It now needs to create its internal data structures to facilitate playback later on:

- A timer and event processor are created. The timer is the master clock for the presentation, and the event processor is a runqueue object that is used for low-level scheduling. Whenever the high-level scheduler wants some code to be executed it will add an event to the event processor, possibly with a timeout and a priority. The event processor runs in a separate thread, waits for events in it runqueue to become elegible and then runs them. This mechanism is the underlying engine that makes the whole player work, anything that needs to wait doesn't do so inline but uses the event processor to get a callback at a later stage: the scheduler, renderers needing input data, etc.
- A layout manager is created, which will be used to find where media items should be displayed. The smil layout manager reads the <layout> section of the DOM tree and builds a parallel layout tree (which also contains information on some of the body media nodes, the ones that have layout information themselves) of region node objects. Then this tree of region node objects is converted into a tree of surface template objects. To create toplevel windows the new topsurface method of the window factory is used, and

subregions are created using the new subsurface method of their parent surface template. The layout manager also contains mappings to be able to get from a node to the corresponding region node to the surface template, and this will be used during playback to play media items in the correct location.

- A timegraph is created. This is the internal representation of the <body> part of the DOM tree that will be used to play back the document. In addition a scheduler is created, which will interpret the data in the timegraph.

3. Starting playback

When the user selects Play we call the start method of the player object. This will invoke start on the scheduler. This will start playing the root node of the tree. The scheduler will now do all the SMIL 2 magic, whereby events such as the root node being played causes other nodes to become playable, etc. At some point a media item needs to be rendered. The scheduler calls the new playable method from the global playable factory. This will pass the DOM node to the various factories until one signals that it can create a playable for the object. In addition, if the playable has a renderer (which is true for most media objects, but not for things like SMIL animations) we also obtain the surface on which the media item should be renderered, through the layout manager. We then tell the renderer which surface to use. Soon afterwards the start method of the playable is called to start playback. **An average renderer will need to obtain data from AxIndex (note, originally, AMBULANT obtains from URL). It will do this by creating a datasource for the document through the datasource factory object.** Every time the renderer wants more data it calls the start method of the datasource passing a callback routine. Whenver data is available the datasource will schedule a call to the callback routine, through the event processor. When the renderer has enough information to start drawing it will not actually draw immedeately, but it will send a need redraw call to its surface. This will percolate up the surface hierarchy, to the GUI code, and eventually come back down as a redraw call all the way to the renderer (assuming it is not obscured by other media items, etc). At this point the bits finally get drawn on the screen. Whenever anything "interesting" happens in the renderer (the media item stopped playing, the user clicked the mouse, etc) it invokes a corresponding method on its playable notification. This interface is implemented by the scheduler, and these notifications are how the scheduler gets informed that it can start scheduling new things, etc.

4. Steps for real integration

Steps for ambulant player integration:

- Modifying the SMIL file input to direct access
- Modifying text/image/audio/video renders to accept a link to an AxObject
- Modifying memfiles for direct playing text and images
- Creating a DirectShow input with direct play. (In common with DSI)
- Adapting the output to fit the AxEditor frames

Modifying the SMIL file input to direct access:
This is done by defining new methods for DOM Tree/document generation and by using these new methods in a new player constructor with stream interface.

Modifying text/image/audio/video renders to accept a link to an AxObject:
There we have to modify all functions that call loading(render) function to accept an AxObjectManager pointer to access the played data. In the renders functions, we have to solve the "url" to AxIndex to get the datas (done by FindResourceIndex in memfile).

Modifying memfiles for direct playing text and images:
Memfiles are used in text and image playing. By modifying them we allow direct access for image and text. For audio and video, we add all necessary static method for getting istream* from "url".

Creating a DirectShow input with direct play. (In common with DSI):
We have to modify the way Ambulant calls DirectShow to be compliant to istream*. For this we have to define a new DirectShow SourceFilter. And connect this new SourceFilter to the Renders.

Adapting the output to fit the AxEditor frames:
This consists in several little modifications for frames.

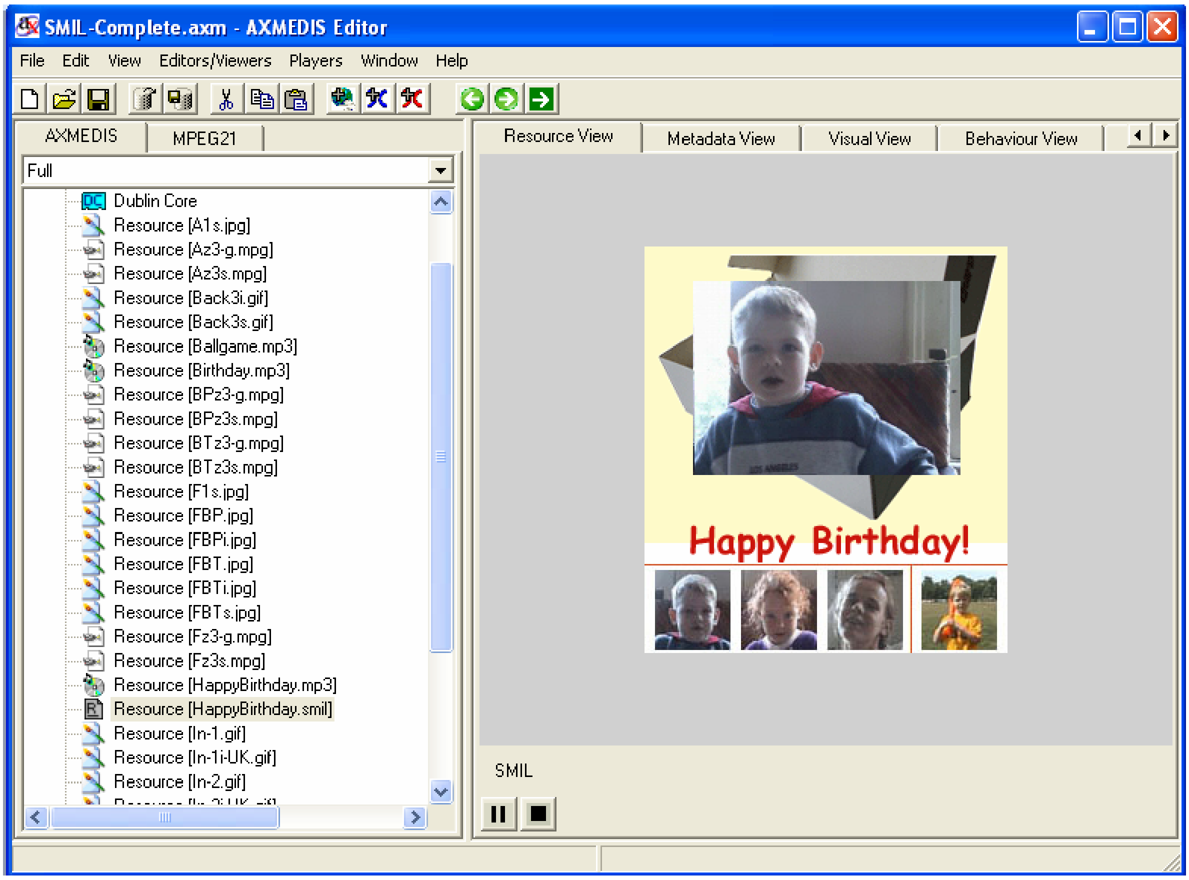


Figure 29 – Sample SMIL presentation rendered by the internal Axmedis SMIL player

7.3.3 Scalability and security issues for PC based SMIL players

No scalability and security is associated with SMIL player.

7.4 AXMEDIS OSMO MPEG-4 Player (EPFL)

Some players for mobile will be also studied with reduced capabilities: e.g. OSMO MP4 Player has been successfully integrated with AXOM without using temporary files. The MPEG-4 internal player constitutes a slightly different case of Media Player for AXMEDIS. In fact MPEG-4 itself not only support media content in terms of different media files or streams, but it satisfies a much more relevant number of requirements providing tools to multiplex and synchronize all the elementary media streams even in the wider context of a rich multimedia scene (including user navigation, user interaction, inherent behaviour of the scene and presentation of natural and synthetic sounds and media). All this is included in a compliant MPEG-4 Player, so that any kind of control description or rule is normally coded inside the mp4 file or systems specific stream. The overall architecture of the Player in accordance to the MPEG-4 specification is reported in the following picture (control flow in dotted lines):

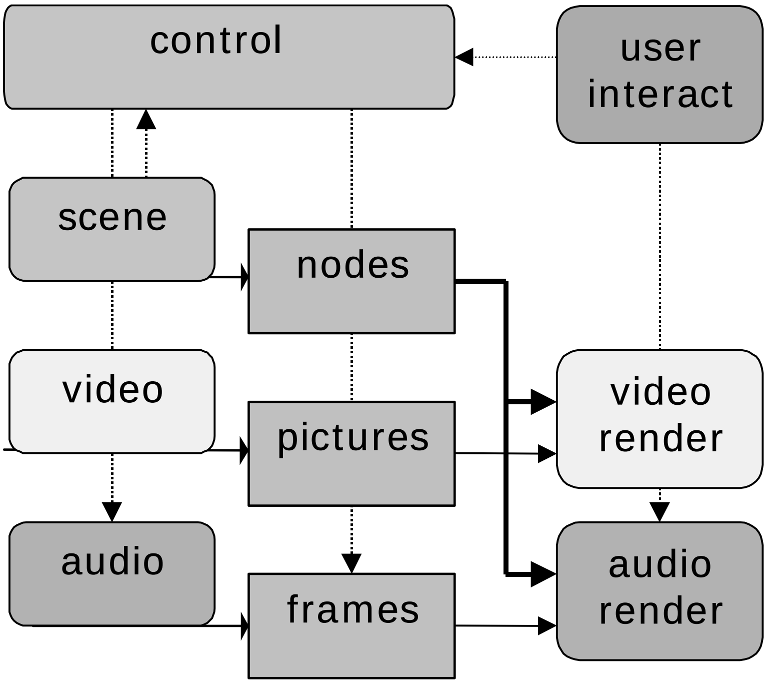


Figure 30 – Architecture of the Player

Management of specific protection rules is also possible in relevant points of the above diagram according to the MPEG IPMPX specification.

For all these reasons including the MPEG-4 Player into the internal AXMEDIS resources may result rather straightforward as only a very reduces number of commands are transmitted from the current *Player* user interface to the underlying architecture (executive control).

The overall player interface can be based on the abstract class **axMediaPlayer** (see previous sections above), through the specialized class **axMPEG4Player**. The functionality that is implemented by this class is rather reduced in terms of operations, given the complex architecture of the player itself and associated content described above.
Currently the MPEG-4 Player can allow two working modes:

- Network Channel (DMIF): in this modality the only possible command is **open** of a network address After this is done by validation of the rights through the AXOM, all the streaming content is received and rendered including audiovisual objects and scene/interaction. Connection is closed when a new one is open or the Player is closed.
- File (MP4): in this modality and under the AXOM control **load** of a file is possible and content is available as for the network modality. In any case this mode may allow the implementation of simple axMedia functions like **start/stop/pause** since file is available and no indeterminate buffering is necessary. All more than this may be really complex as it will interact with the decoding process of all built-in MPEG-4 decoders. More complex behavior for multiple media in AXMEDIS can be implemented in single objects linked through SMIL in the main AXMEDIS Player (and Editor).

Once open or load are allowed, user activity can be monitored by built-in tracing capabilities and possibly reported: it is in any case *MPEG-4 activity* in terms of operation on the MPEG-4 content by built-in sensors and controls.

The OSMO player is based on the GPAC library. GPAC is written in C for portability reasons (embedded platforms and DSPs), attempting to keep the memory footprint as low as possible. GPAC provide a possible work mode on the MPEG-4 file, however, AXMEDIS editor only provide a C++ istream class interface to access the MPEG-4 resource in AXMEDIS object. A simple solution for this problem is to extract the

MPEG-4 resource and save them into a temporary MPEG-4 file. But this solution is not optimal because it requires an addition disk space for storing the temporary file.

A better solution is that the player directly accesses the MPEG-4 resource from AXMEDIS object without saving a temporary file. To implement this solution, GPAC interface has to be changed. A new interface has been developed and it consists of several interface functions, which make it possible for GPAC C code to operate on a C++ istream class instead of a file. Then some interface source code in GPAC has to be modified, and all the source code relative to file operation has to be replaced by the new interface, which is used for player to access the MPEG4 resource from AXMEDIS object using C++ istream class.

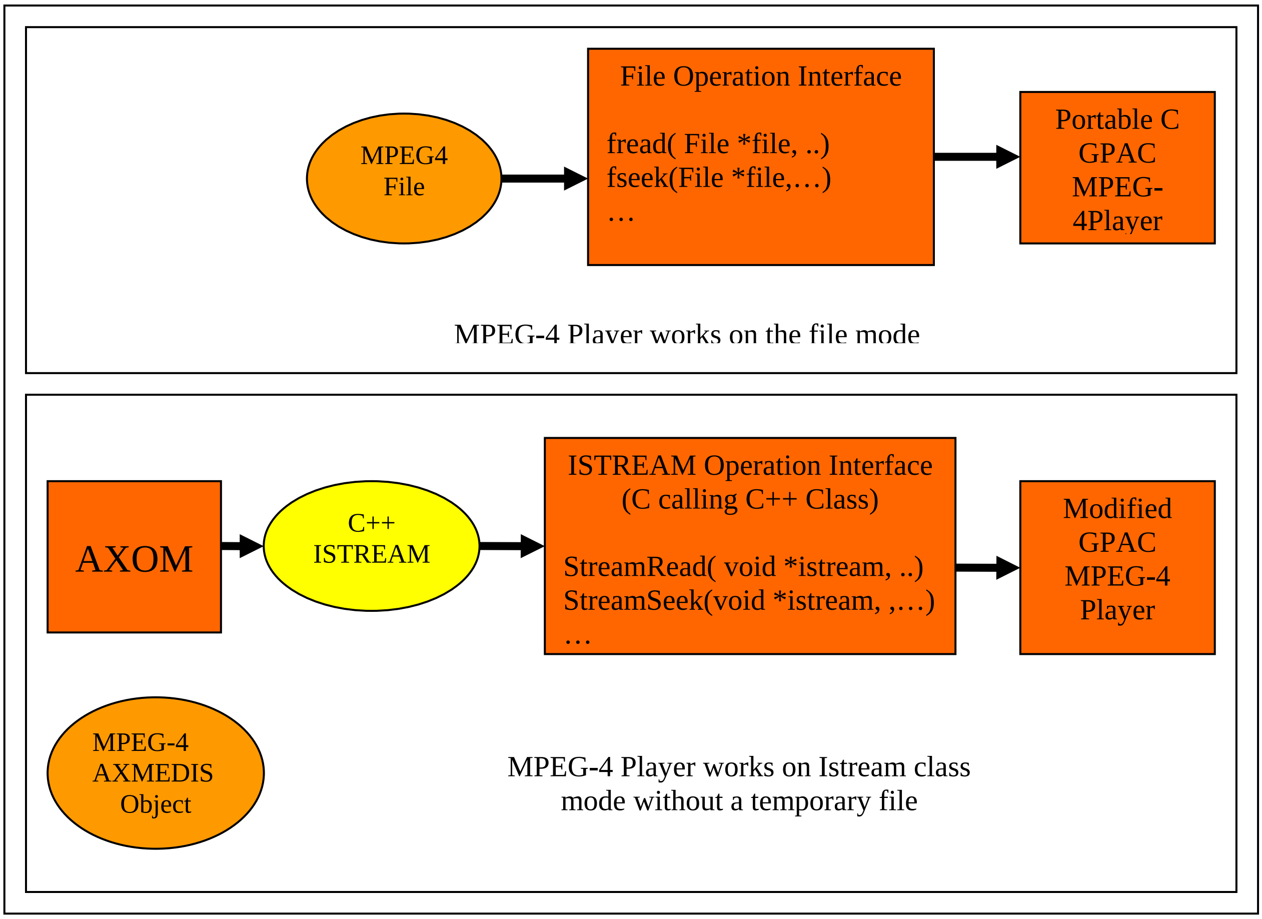

MPEG4 File
File Operation Interface
fread(File *file, ..)
fseek(File *file,…)
…
Portable C GPAC MPEG-4Player
MPEG-4 Player works on the file mode
AXOM
C++ ISTREAM
ISTREAM Operation Interface (C calling C++ Class)
StreamRead(void *istream, ..)
StreamSeek(void *istream, ,…)
…
Modified GPAC MPEG-4 Player
MPEG-4 AXMEDIS Object
MPEG-4 Player works on Istream class mode without a temporary file

Figure 31 – MPEG-4 Player accesses MPEG4-resource from an AXMEDIS object using an ISTREAM operation interface without a temporary file

7.4.1 Integration of the OSMO MPEG-4 Player

The integration has been finished with AXMEDIS Editor. The technical detail can be seen in the above chapter and more details can be found in the last version of the deliverable DE3-1-2-3-4 “Specification of AXMEDIS Editors and Viewers”.

7.4.2 Scalability and security issues related to the OSMO MPEG-4 Player

To protect the audiovisual resources is used IPMP-X for MPEG-4 player.

7.5 AXMEDIS PDA Player (EPFL - TISCALI , DSI, FUPF, SEJER from 25)

AXMEDIS consortium is working on realising a PDA player for MPEG-21 it is going to include:

- AXOM for the loading parts (DSI)
- Protection processor with limited capabilities (DSI)
- Fingerprint hardware and software (DSI)
- PMS client with secure connection capabilities and secure cache of sensible information (only in a second stage), (FUPF)
- User interface (TISCALI)
- Integrated OSMO MPEG-4 player (TISCALI)
- Integrated SMIL player (TISCALI)
- Possible hierarchy view (DSI), only in a second phase
- Possible document viewer, only and if it is possible in a second phase, including HTML, PS, and PDF (SEJER or DSI)

A first prototype of the PDA was accessible for the following capabilities:

- AXOM for the loading parts (DSI)
- Protection processor with limited capabilities (DSI)
- Fingerprint hardware and software (DSI)
- User interface (TISCALI)
- Integrated OSMO MPEG-4 player (TISCALI)

At present, the PDA player application is complete.

7.5.1 Technical details

Reference to the AXFW location of the demonstrator	https://cvs.axmedis.org/repos/extlibs/axPocketPCPlayer/bin/AXPDAPlayerInstaller.CAB
List of libraries used	OpenSSL, Xcerces
References to other major components needed	AXOM
Problems not solved	
Configuration and execution context	
Programming language	C, C++

The AXMEDIS PDA player is an application shell, mixing different “playing” technologies.
Currently are supported 3 capabilities:

- SMIL content playing/execution
- MPEG4 content playing/execution
- HTML playing/execution

The UI is completely written in C/C++ and compiled with Visual Studio 2005.
Current target for the application are PDAs with Windows Mobile 2005 operating system.

7.5.2 The user interface

The AXPDAPlayer presents a “File Menu” and a “View Menu”.The View Menu is just a set of commands that can modify the behaviour of the player. The menu can change in relation of the playing style (MPEG4, SMIL or HTML).

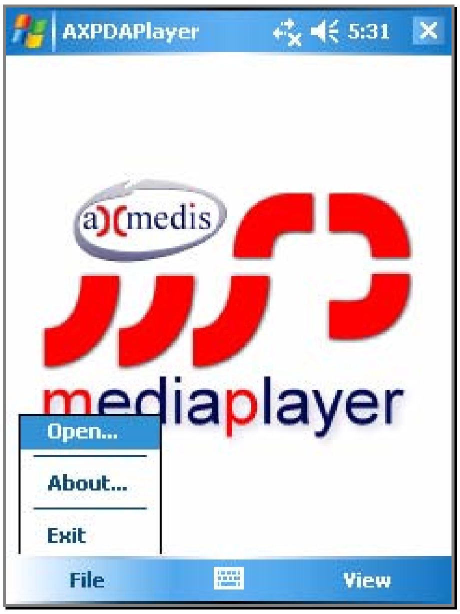


Figure 32 - The File Menu

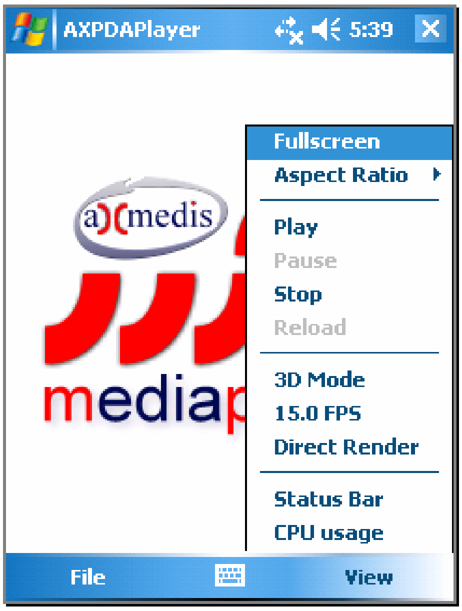
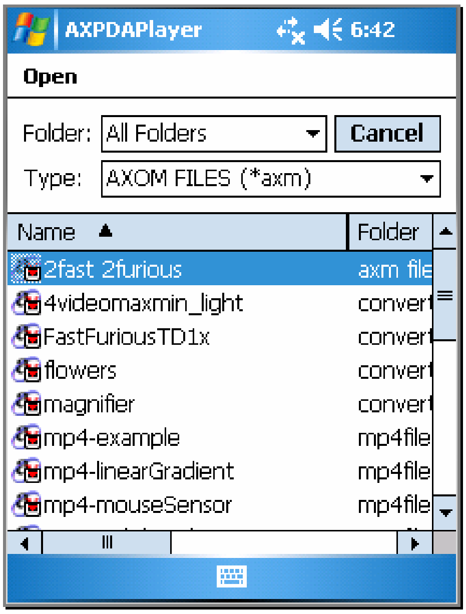


Figure 33 - The View Menu

The File Menu has as main command "Open…". Using the "Open…" command will bring up the PDA File Chooser Dialog.



Folder: All Folders

Type: AXOM FILES (*axm)

Name	Folder
2fast 2furious	axm file
4videomaxmin_light	convert
FastFuriousTD1x	convert
flowers	convert
magnifier	convert
mp4-example	mp4file
mp4-linearGradient	mp4file
mp4-mouseSensor	mp4file

Figure 34 - File Chooser Dialog

When a file has been chosen, all the resources in the AXOM will be scanned. The scanner looks for resources with these mime/types:

- "application/smil"
- "text/html"
- "video/mpeg4"
- "video/mp4"
- "application/mp4"

When one of these types is found the application starts the related “player”.

7.5.3 The SMIL Player

The SMIL player is an implementation with some tweaking of the AMBULANT Open SMIL player. The work done is very close to the efforts done for the AXOM PC Player. A big push has been done to port and recompile under Windows Mobile 2005.
The functionalities are equivalent to the one implemented in AXMEDIS PC player (section 7.3.2).
When the inner resource scanner find a mime/type “application/smil” the related istream to the axom resource is assigned to the SMIL player. Naturally tweaking were done to let the SMIL player accept AXOM streams.
New constructor methods were added to all the players and renderers in the LIBAMBULANT framework. The main reason was to keep the LIBAMBULANT aware of the `AxObjectManager object.`
The other big changes were the streming methods contained int win32_memfile.cpp. We added methods able to stream directly from Resource embedded in AXOM objects.
Everything is memory based, no files are saved locally, The SMI LPlayer play and acts just as it is playing a standard SMIL file (as described in section 7.3.2).

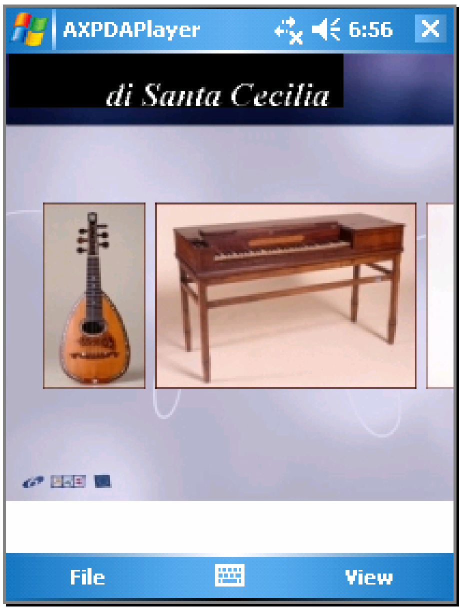


Figure 35 - AXPDA Player in SMIL mode

7.5.4 The MPEG-4 Player

As with the SMIL player, the MPEG-4 player is based on Open source player: the OSMO MP4 player. The integration of the OSMO MP-4 player code has been done in the same way as the PC player porting (check section 7.4 and Figure 30 for reference).

The main engine for timing and rendering is the GPAC library. The main working mode for the GPAC is file based, so we had to do some tweaking to let the GPAC accept iStreams coming from AXMEDIS embedded resources.
The other problem was to interface C++ streams with a library (GPAC) able only to use “FILE*” style streaming. We changed the “bitstream.c” file and we created a shim using set of function, practically simulating a file pointer using the c++ streams (axom_stream_mgr.cpp source) (Figure 31).
As it is now the MPEG-4 player in the PDA is opening AXOM resource only using memory and not saving temporary files.

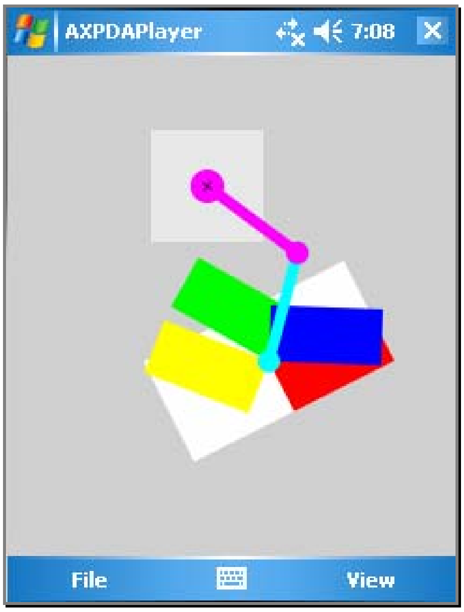
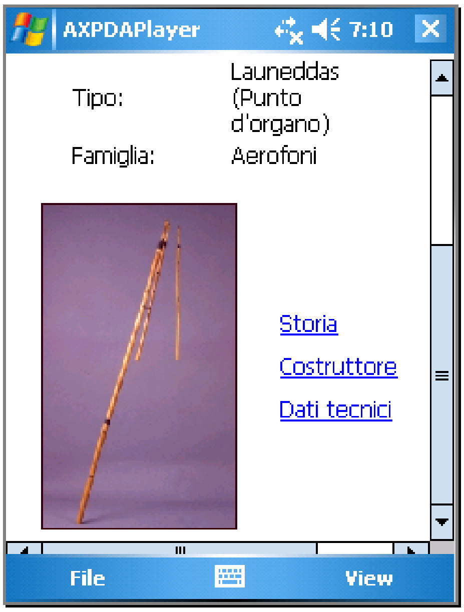


Figure 36 - AXPDA Player in MPEG4 mode

7.5.5 The HTML Player

The present AXMEDIS PDA player is endowed of an integrated player for HTML. Also in this case the resources referred by the HTML may located into the MPEG-21/AXMEDIS objects or can be exernal links. For this purpose the protocol for processing the links has been modified.



Tipo: Launeddas (Punto d'organo)

Famiglia: Aerofoni

Figure 37 - AXPDA Player in HTML mode

7.5.6 Integration of the PDA player and security issues

In the AXMEDIS scenery there are 2 ways the PDA player becomes involved: the player certification and content authorizations.

- *PDA player certification*

The PDA player as the PC player or any other player in the future must be authorized to play protected AXMEDIS objects. A certificate is issued for a specific device. When the player starts, it tries to authenticate itself to the PMS server. The authentication is done automatically by the AXOM part of the player. The AXOM framework knows where is located in the device the preferred current certificate and use it for the authentication. Using this certificate, the AXOM framework open a channel with the PMS server using a component called PMS Client.

- *PDA content authorization*
  When a protected content, let’s say a protected video MPEG-4 embedded in the AXMEDIS object, is opened by the player, the AXOM framework will do a sequence of operation:

  - *AXPDAPlayer verification:* to check if the player was tampered, an hash is computed and sent to the PMS server. The PMS server will confirm if the player is the one authorized by the player certificate. The AXPDAPlayer has no secure cache to store keys and memorize rights when is offline (as we suppose the tool is always online).
  - *Content authorization:* the AXOM framework will query the PMS for the AXOID rights. If the user owns the license, the PMS will return the “protection info” necessary to decrypt the AXOID. At this point the player can show the contents (in this case the video).

7.6 Cryptography tools and algorithms for security processing

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.
The following subsections describe the external library that may be used in the AXMEDIS framework to implement the needed cryptography functionalities.

The cryptlib functionalities could be used as plug-ins through the AXCP interface. The interface of AXCP plug-ins can map the formal description of the function and allows entering textually all parameters of the function (key, Mode, Algorithm).

7.6.1 Algorithms

This section describes the characteristics of each algorithm used in cryptlib and any known restrictions on their use.

- **AES** AES is a 128-bit block cipher with a 128-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_AES.
- **Blowfish** Blowfish is a 64-bit block cipher with a 448-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_BLOWFISH.

- **CAST-128** CAST-128 is a 64-bit block cipher with a 128-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_CAST.
- **DES** DES is a 64-bit block cipher with a 56-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_DES. Note that this algorithm is no longer considered secure and should not be used. It is present in cryptlib only for compatibility with legacy applications. Although cryptlib uses 64-bit DES keys, only 56 bits of the key are actually used.
- **Triple DES** Triple DES is a 64-bit block cipher with a 112/168-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_3DES. Although cryptlib uses 128, or 192-bit DES keys (depending on whether two- or three-key triple DES is being used), only 112 or 168 bits of the key are actually used.
- **Diffie-Hellman** Diffie-Hellman is a key exchange algorithm with a key size of up to 4096 bits and has the cryptlib algorithm identifier CRYPT_ALGO_DH. Diffie-Hellman was formerly covered by a patent in the US, this has now expired.

7.6.2 Mode

A symmetric key algorithm encrypts plaintext in fixed-size n-bit blocks (often n = 64/128/256). For messages exceeding n bits, the simplest approach is to partition the message into n-bit blocks and encrypt each separately. This electronic-codebook (ECB) mode has disadvantages in most applications, motivating other methods of employing block ciphers (*modes of operation*) on larger messages.

The four most common modes are ECB, CBC, CFB, and OFB. These are summarized and discussed below.

- **ECB mode** The *electronic codebook* (ECB) mode of operation
- **CBC mode** The *cipher-block chaining* (CBC) mode of operation involves use of an n-bit initialization vector, denoted IV
- **CFB mode** the *cipher feedback (CFB)* where some applications require transmission without delay.
- **OFB mode** The *output feedback* (OFB) mode of operation may be used for applications in which all error propagation must be avoided. It is similar to CFB, and allows encryption of various block sizes (characters), but differs in that the output of the encryption block function E (rather than the ciphertext) serves as the feedback.

See figure below.

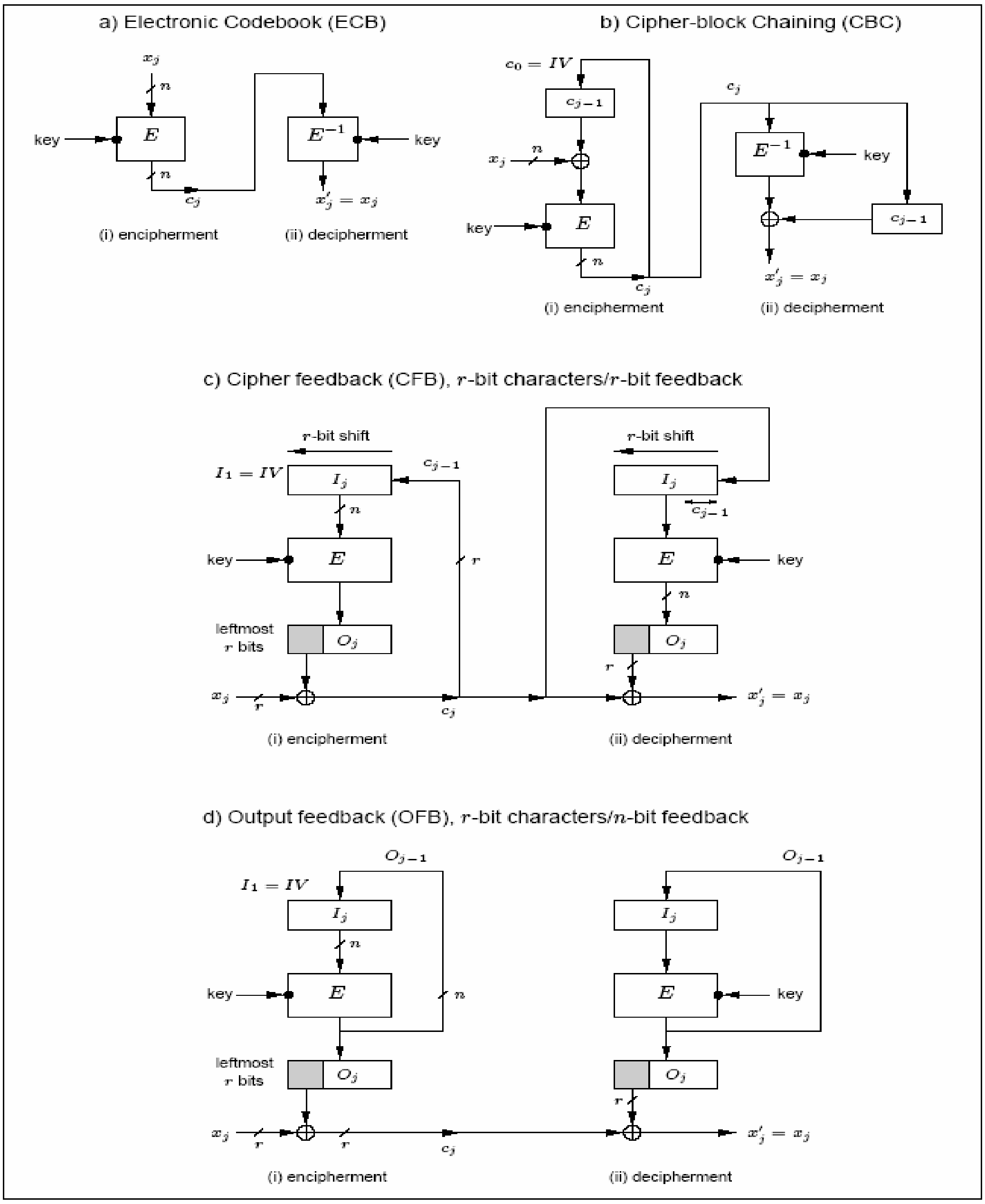


Figure 38 – Modes

8 Conclusion

(This section has to report the conclusion of the work done regarding this deliverable. **From the last Verification Meeting Agenda and Minute: *"Close a report with a clear summary/conclusions section on what have been achieved, what remains to be done and what conclusions may be drawn from the conducted work."*)**

(DSI, please, check/update or add information about the work done for this deliverable - for the sections on "Supporting AXMEDIS DRM in Windows Media" , "P2P support for B2C application and distribution" and "AXMEDIS PC based Player"-)

(EPFL, please, check/update or add information about the work done for this deliverable - sections on "AXMEDIS OSMO MPEG-4 player" and "AXMEDIS SMIL Player"-).

This deliverable is devoted to the analysis of all the aspects of content distribution via Internet: CMS features, DRM and interoperability issues, distribution of the content in the B2B and B2C environment and content consumption on different devices. The deliverable performs the research activity preliminary for the development step of all the necessary software tools for content distribution on Internet.

Starting from the state of the art of the CMS solutions available on the market (section 3), this deliverable studies specific problems related to content distribution of multimedia assets via Internet. The aim is to point out which are the main features suitable for solving problems associated with cross media asset management and distribution and for simplifying their integration with all the components of the AXMEDIS technology. The analysis also includes an introduction to XAURA, an open, java based CMS used by Tiscali and suitable for rapid prototyping content applications for the web. The research activity performed in this deliverable has been preparatory for the integration of the AXMEDIS framework with the XAURA platform. The final result has been the development of an experimental channel for the distribution via Internet of content coming from the AXMEDIS framework.

In section 4, the deliverable performs a deep analysis of the state of the art of DRM technology for content distribution also taking into account interoperability issues among different systems. After that, the document focuses on security issues related to the AXMEDIS technology, highlighting weak points of some existing technologies regarding the respect of user's privacy (according to some European laws) and explaining how the AXMEDIS solution faces this kind of problems.
The section also describes users and tools registration, certification and verification for content usage in every possible scenario and how the DRM models for Internet distribution have been developed. In order to tackle the integration of AXMEDIS DRM and other commercial tools that also use their own DRM, the cases of Microsoft Windows Media Rights Manager and Windows Media Player technologies have been taken into account. Furthermore, the section describes a set of licenses accomplishing the specific requirements made by TISCALI for content distribution via Internet.

Section 5 verifies the possibility of using DRM model and format into a Windows Media delivering of content.

In section 6 a study has been performed in order to adapt the AXMEDIS P2P solution for B2B distribution (AXEPTool) to the content delivery in the B2C environment (AXMEDIA). The B2B solution (based on BitTorrent) is presented, describing the main use case in order to clarify functionalities to be exposed by the AXEPtool.

Section 7 is devoted to the analysis of the state of the art of existing MPEG-21 clients and other multimedia player terminals and to the description of the developed AXMEDIS clients for distribution on PC, mobile and PDA.

Regarding the AXMEDIS player for PC, two players have been developed and described also including some screenshots: the first one uses the Microsoft Foundation Classes in order to open and display an AXMEDIS object and the associated hierarchy and the second one is based on an open source SMIL player (called AMBULANT).

The description of the player for the mobile shows the successful integration of the existing OSMO MP4 Player with AXOM, without using temporary files.

The AXMEDIS PDA player is an application supporting SMIL, MPEG4 and HTML playing/execution. The player functionalities and GUI have been described also explaining the security issues related to its integration with the AXMEDIS framework.

Finally, the section includes information about the library used to implement cryptographic functionalities into the AXMEDIS framework.

9 Bibliography

[1] Leonardo Chiariglione, MPEG Group, http:// www.chiariglione.org/mpeg , standard MPEG-21 , http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm, 2004.
[2] ISO/IEC, ISO/IEC 2nd Edition TR 21000-2 – Vision, Technologies and Strategy.
[3] ISO/IEC, ISO/IEC 2nd Edition FCD 21000-2 – Digital Item Declaration.
[4] ISO/IEC, ISO/IEC FDIS 21000-3 – Digital Item Identification.
[5] ISO/IEC, ISO/IEC CD 21000-4 – Intellectual Property Management and Protection.
[6] ISO/IEC, ISO/IEC IS 21000-5 – Rights Expression Language.
[7] ISO/IEC, ISO/IEC IS 21000-6 – Rights Data Dictionary.
[8] ISO/IEC, ISO/IEC IS 21000-7 – Digital Item Adaptation.
[9] ISO/IEC, ISO/IEC FDIS 21000-8 – MPEG-21 Reference Software.
[10] ISO/IEC, ISO/IEC IS 21000-9 – File Format.
[11] ISO/IEC, ISO/IEC FCD 21000-10 – Digital Item Processing.
[12] ISO/IEC, ISO/IEC TR 21000-11– Evaluation Methods for Persistent Association Technologies.
[13] ISO/IEC, ISO/IEC TR 21000-12 – Test Bed for MPEG-21 Resource Delivery.
[14] ISO/IEC, ISO/IEC CD 21000-14 – Conformance Testing.
[15] ISO/IEC, ISO/IEC WD 21000-15 – Event Reporting.
[16] ISO/IEC, ISO/IEC FDIS 21000-16 – Binary Format.
[17] ISO/IEC, ISO/IEC WD 21000-17 – Fragment Identification of MPEG Resources.
[18] Bob Boiko, “Content Management Bible”, John Wiley & Sons, Inc. - New York, NY, USA 2001
[19] Bill Trippe, Mark Walter, “Content Management and Web Publishing”, chapter 10 of “The Columbia Guide to Digital Publishing”, Columbia University Press, New York, NY, USA 2003, http://www.digitalpublishingguide.com
[20] Addey, Ellis, Suh, Thiemecke, “Content Management Systems (Tools of the Trade)”, Apress L.P., Berkeley, CA, USA 2003
[21] Step Two Designs papers, http://www.steptwo.com.au/papers/index.php
[22] TechSoup.org articles from “Web Building” section, http://www.techsoup.org/howto/articles.cfm?topicid=13&topic=Web%20Building
[23] “XML Signature Syntax and Processing”, W3C Recommendation 12 February 2002, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/
[24] Microsoft DRM, http://www.microsoft.com/windows/windowsmedia/drm/default.aspx
[25] CMS Matrix website, http://www.cmsmatrix.org/
[26] CMS Review website, http://www.cmsreview.com/
[27] Hartman Communicatie BV, Content Management Overview web pages, http://tools.hartman-communicatie.nl/overzicht.html
[28] Internet Streaming Media Alliance (ISMA), http://www.isma.tv/
[29] Digital Media Project (DMP), http://www.dmpf.org/
[30] Interoperable Digital Rights Management Platform, D440, 15 April 2005.
[31] IETF RFC 1737, K. Sollins and L. Masinter, Functional Requirements for Uniform Resource Names, December 1994.
[32] Alan McCord, "Overview of Digital Asset Management Systems", EDUCASE Evolving Technologies Committee, University of Michigan, September 2002
[33] Mark Choate, "XML and Cross Media Content Management", Business Intelligence Advisory Service Executive Update, August 2004
[34] Magan Arthur, "Intro to Digital Asset Management: Just what is a DAM?", April 2005, article from CMS Watch web site, http://www.cmswatch.com/
[35] Mark Steinwinter, "Content Management Systems: Bursting the DAM", Business Intelligence Advisory Service Executive Update, September 2004
[36] Chris Lynn, "Digital Asset Management: The Product Landscape", August 2002, article from CMS Watch web site, http://www.cmswatch.com/
[37] "Managing Enterprise Content: A Unified Content Strategy", White Paper from Advisor Web page, http://zones.advisor.com/doc/12268

[38] Ann Rockley, "A Metadata Primer", February 2003, article from CMS Watch web site, http://www.cmswatch.com/
[39] Creative Commons, http://www.creativecommons.org/
[40] Real Networks, http://www.realnetworks.com/
[41] Helix DRM, http://www.realnetworks.com/products/drm/index.html
[42] Real Networks Media Commerce Suite, http://www.realnetworks.com/products/commerce/description.html
[43] Trymedia ActiveMark DRM, http://www.trymedia.com/services/security_drm.shtml
[44] Bill Rosenblatt and Gail Dykstra, "Integrating Content Management with Digital Rights Management", May 2003, from DRM Resource Center Web pages, http://www.drmcentre.org.in/
[45] James Hilton, "Digital Asset Management Systems", 2003, from Educause Web pages http://www.educause.edu/
[46] MySource Matrix Web page, http://matrix.squiz.net
[47] Mambo Web page, http://www.mamboserver.com
[48] Apache Lenya Web page, http://lenya.apache.org
[49] Plone Web page, http://www.plone.org
[50] OpenCMS Web page, http://www.opencms.org
[51] Typo3 Web page, http://www.typo3.com
[52] Drupal Web page, http://www.drupal.org
[53] eZ Publish Web page, http://ez.no/ez_publish
[54] Vignette V7 Web page, http://www.vignette.com/contentmanagement/0,2097,1-1-1928,00.html
[55] Stellent Universal Content Management Web page, http://www.stellent.com/stellent3/idcplg?IdcService=SS_GET_PAGE&nodeId=64&ssSourceNodeId=7
[56] FatWire Content Server Web page, http://www.fatwire.com/products/contentserver.html
[57] Documentum Web page, http://www.documentum.com/products/contentmanagement_products.html
[58] Livelink Web page, http://www.livelink.com
[59] Mediasurface ECM Web page, http://www.mediasurface.com/product
[60] QP7.Enterprise Web page, http://www.quantumart.com/qp7enterprise
[61] Tridion R5 Web page, http://www.tridion.com/Products/R5/Overview.asp
[62] Interwoven Web page, http://www.interwoven.com
[63] Communiqué Web page, http://www.day.com/site/en/index/products/content- centric_applications/overview.html
[64] CoreMedia CMS 2005, http://www.coremedia.com/coremedia.aspx/products/cms-2005-news/language=en/id=67852/cms-2005.html
[65] ECMsuite Web page, http://www.componence.com
[66] CMS400.NET Web page, http://www.ektron.com/cms400.aspx
[67] FileNet P8 Web page, http://www.filenet.com/English/FileNet_P8/index.asp
[68] Ingeniux CMS Web page, http://www.ingeniux.com/x628.xml
[69] RedDot XCMS Web page, http://www.reddot.com/products_enterprise_content_management.htm
[70] Rhythmyx Web page, http://www.percussion.com/products/content-management/rhythmyx/
[71] Terminalfour Site Manager Web page, http://www.terminalfour.com/products/sitemanager/
[72] Xerox DocuShare Web page, http://docushare.xerox.com
[73] IBM DB2 Web page, http://www-306.ibm.com/software/data/cm/cmgr/mp/
[74] Oracle Portal Web page, http://www.oracle.com/appserver/portal_home.html
[75] Microsoft CMS 2002 Web page, http://www.microsoft.com/cmserver/
[76] Ariadne Web page, http://www.ariadne-cms.org/
[77] Richard Lonneborg, “Extending an MPEG-21 viewer to manage access rights”, PhD thesis, University of Wollongong.
[78] IST-ENTHRONE Project, start on 2004, http://www.enthrone.org/
[79] Macromedia Flash Web page, http://www.macromedia.com/software/flash
[80] Microsoft Windows Media Player Web page, http://www.microsoft.com/windows/windowsmedia/
[81] Apple Quick Time Web page, http://www.apple.com/quicktime/
[82] Real Player Web page, http://www.realnetworks.com/info/real10_platform/
[83] iTunes Apple Web page, http://www.apple.com/itunes/
[84] Northcode Inc. Web homepage, http://www.northcode.com/

[85] WinAmp Web product page, http://www.winamp.com/
[86] Apple Fairplay, http://www.apple.com/itunes/
[87] Coral Consortium, http://www.coral-interop.org
[88] DReaM, http://www.openmediacommons.org
[89] Marlin, http://www.marlin-community.com/
[90] OpenSDRM, http://sourceforge.net/projects/opensdrm/
[91] Silvia Llorente, Jaime Delgado, Rubén Barrio, Xavier Maroñas, Translation between XML-based rights expressions using UML and relational models, AXMEDIS 2006 Conference

10 Glossary

A

API, **A**pplication **P**rogramming **I**nterface
A set of definitions of the ways one piece of computer software communicates with another.

ASP, **A**ctive **S**erver **P**ages
A web-scripting language by Microsoft.

B

B2B, **B**usiness to **B**usiness
Refers to one business communicating with or selling to another.

B2C, **B**usiness to **C**onsumer
Refers to a business communicating with or selling to an individual rather than a company.

C

CM, **C**ontent **M**anagement
Designs the set of processes and technologies supporting the evolutionary life cycle of digital information.

CMS, **C**ontent **M**anagement **S**ystem
A software tool designed to help content managers create, manage, and publish their content.

CRM, **C**ustomer **R**elationship **M**anagement
An information industry term for methodologies, software, and Internet capabilities that help an enterprise manage customer relationships in an organized way.

CSS, **C**ascading **S**tyle **S**heets
A style sheet format for HTML documents endorsed by the World Wide Web Consortium.

D

DAM, **D**igital **A**sset **M**anagement

DRM, **D**igital **R**ights **M**anagement
A system for protecting the copyrights of digital content.

E

ECM, **E**nterprise **C**ontent **M**anagement
The set of technologies, tools, and methods used to capture, manage, store, preserve, and deliver content across an enterprise.

ERP, **E**nterprise **R**esource **P**lanning
An integrated information system that serves all departments within an enterprise.

F

FTP, **F**ile **T**ransfer **P**rotocol
A protocol used to transfer files over a TCP/IP network.

G

GNU, **G**nu's **N**ot **U**NIX
A project sponsored by the Free Software Foundation that develops and maintains a complete software environment.

GPL, **G**NU General **P**ublic **L**icense
The license that accompanies the GNU software from the Free Software Foundation.

GUI, **G**raphical **U**ser **I**nterface
A computer terminal interface based on graphics instead of text.

H

HTTP, **H**yper**T**ext **T**ransport **P**rotocol
The communications protocol used to connect to servers on the World Wide Web.

HTTPS, **H**yper**T**ext **T**ransport **P**rotocol **S**ecure
Version of the HTTP protocol that enables secured transmission of Web pages.

HTML, **H**yper**T**ext **M**arkup **L**anguage
T*he document format language used on the World Wide Web.*

I

ICE, **I**nformation and **C**ontent **E**xchange
A data sharing specification that allows one Web site to obtain data from another Web site.

IP, **I**nternet **P**rotocol
The network layer protocol in the TCP/IP communications protocol suite.

J

J2EE, **J**ava **2** **E**nterprise **E**dition
A version of Java for developing and deploying enterprise applications.

L

LDAP, **L**ightweight **D**irectory **A**ccess **P**rotocol
A protocol used to access a directory listing.

N

.NET
Microsoft's framework for Web services and component software.

O

ODBC, Open **D**ata**B**ase **C**onnectivity
A database programming interface from Microsoft that provides a common language for applications to access databases on a network.

R

RDF, **R**esource **D**escription **F**ramework
A standard for describing resources on the Web endorsed by the World Wide Web Consortium.

RSS, **R**eally **S**imple **S**yndication
A syndication format (developed by Netscape in 1999) for aggregating updates to blogs and the latest news from Web sites.

S

SSL, **S**ecure **S**ockets **L**ayer
A protocol developed by Netscape for transmitting private documents on the Internet.

SCORM, Shareable **C**ontent **O**bject **R**eference **M**odel
A standard for web-based E-learning.

SOAP, **S**imple **O**bject **A**ccess **P**rotocol
A message-based protocol based on XML for accessing services on the Web.

U

UDDI, **U**niversal **D**escription **D**iscovery and **I**ntegration
An industry initiative for a universal business registry of Web services.

UML, **U**nified **M**odeling **L**anguage
An object-oriented analysis and design language.

URI, Uniform **R**esource **I**dentifier
The addressing technology for identifying resources on the Internet.

URL, **U**niform **R**esource **L**ocator
A type of uniform resource identifier (URI) that uses an HTTP connection.

W

WCM, **W**eb **C**ontent **M**anagement

WSDL, **W**eb **S**ervices **D**escription **L**anguage
An XML-based language (Developed by Microsoft and IBM) for defining Web services.

WYSIWYG, **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et
A graphical interface to a process which shows how the end-result will look as it is being produced.

X

XHTML, e**X**tensible **H**yper**T**ext **M**arkup **L**anguage
A reformulation of HTML markup language with the same expressive possibilities but a stricter syntax.

XML, e**X**tensible **M**arkup **L**anguage
A metalanguage used to design markups languages.

XSL, e**X**tensible **S**tylesheet **L**anguage
A standard from the World Wide Web Consortium for describing a style sheet for XML documents.

XSLT, e**X**tensible **S**tylesheet **L**anguage **T**ransformation
Processing extensions to the XSL stylesheet language used to convert XML documents.