Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE2.2.1

Test Cases and Content Description

Version: 1.8
Date: 9/2/2005
Responsible: FUPF

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: DE2.2.1
Contractual Date of Delivery: Month 4 (December 2004)
Actual Date of Delivery: 31/1/2005
Work-Package contributing to the Deliverable: WP2.2
Task contributing to the Deliverable:
Nature of the Deliverable: Report
Author(s): All partners

Abstract:

This document describes test cases that will be used for validating the functionalities identified by research and development workpackages and, after that, for integration and optimisation activities, including demonstrators. The test cases defined are mainly derived from the Use cases document (DE2.1.1b) and describe the data set that has to be used for each of them.

Keyword List:
Test cases, data set description, content specification

Table of Contents

1 Executive Summary and Report Scope (mandatory, FUPF)

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 optimise 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 of test cases and content description activities inside WP2. The description of test cases is closely related with AXMEDIS user requirements and use cases, defined in deliverables DE2.1.1a and DE2.1.1b, respectively. The description of content is also closely related with test cases, as the content defined in this document will be used as an entry for performing the tests described.

Test cases will cover the different areas present in AXMEDIS project, considering the different AXMEDIS tools inside AXMEDIS framework. To summarise, the collection of test cases should cover the following research areas: multimedia object production, automatic composition, technical metadata for composition, formatting and composition styles, production profiles, Digital Rights Management, fingerprint, indexing for different types of content, simple query production, complex technical queries, peer to peer tools for business to business integrated with workflow, etc.

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 at the same time as user requirements and use cases are newly defined or revised.

Main deliverables in WP2 are:

- DE2.1.1 -- User Requirements and use cases (M3) – this deliverable contains the description of the user requirements and the corresponding use cases in UML, coming from WP2.1 and WP2.2;
- DE2.2.1 – Test cases and content description (M4) – this deliverable contain the description of the test cases for research functionalities and AXMEDIS tool validation, coming from WP2.2;
- DE2.3.1 – User Group Set up and analysis (M4). The analysis will be done on the basis of the curricula and the needs of the AXMEDIS project, to verify that all the aspects and user profiles and roles will be covered by the user group;
- DE2.3.2 – User Group Maintenance (M13).
- DE2.4.1 – Requirements update (M18);

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

WP2.1 -- Early Requirements Analysis -- responsibility DSI -- collection of user requirements by using the expert user groups. The focus will mainly be on: content workflow, content management, content production, content searching, content rights management (licensing, formalising usage rules), content formatting in the various contexts (PC, mobile, i-TV, kiosk, PDA), user profiling, content composition, fingerprint, watermark, indexing, querying, transaction models, push and pull balancing, etc. In addition, a more detailed analysis of the functionalities that could be useful in the above contexts will be done: query on technical aspects, content composition, content formatting, distribution, content exchanging, certification, supervision, etc. The use cases have to be collected by considering the points of view of content designers, multimedia producers, TISCALI, OD2, ANSC, AFI, ILABS, XIM, SEJER. In addition, EUTELSAT, HP, DSI, DIPITA, CPR, CRS4, IRC, UNIVLEEDS, EPFL, COMVERSE, ACIT, etc., will also collect this information from their experts by using specific interview based on guidelines produced by the consortium. A part of this information will be collected by reviewing the results of several past projects. In the analysis of requirements also those of the AXMEDIS partners and potential customers and SMEs in the respect of the WWW pages for getting general AXMEDIS services will be considered.

WP2.2 -- Use cases and test cases description -- responsibility FUPF -- this WP is devoted to the organisation of the requirements in terms of use cases and the corresponding identification and description of test cases. The test cases will be used for validating the functionalities identified by research and development WPs and during the activities of integration and optimisation, and in those of demonstration, which is temporally allocated after the M30. The Content for the test cases will be collected and/or produced in WP8. The description about how the test cases will be selected and about which content will be suitable for that goal is reported in WP8. The use cases will be structured according to the UML model, including: name, ID, description, context assumptions (equipment, paradigm, location), actors (skill, age, instrument, paradigm), steps, variation, non functional aspects, content, interaction protocol, issues, etc. The test cases will be structured according to structure of the AXMEDIS framework and tools that will be developed in these 18 months of work. The model will be UML including: name, ID, description, functionality to be tested, context, partners involved, Validator(s) skill, data set needed, steps, expected results, variations, issues, additional activities to be considered, metrics to be used, etc. In this subWP, the targeted quality of use of the tools that will be developed during the project will be also defined in terms of metrics for usability. To this end users including the general public will be modelled based on the definition of the user requirements. The usability metrics will be focussed on extracting relevant drivers in the real environment of the application. Use cases and test cases for describing the interaction with the AXMEDIS services provided by the AXMEDIS portal will be separately described.

WP2.3 -- Set up and management of a AXMEDIS User Group -- responsibility UNIVLEEDS -- a user group of experts will be set up. The members of the user group will receive updated information about the project evolution and will constitute a source for testing and validating the produced results. The user group has to present experts representing the different users of AXMEDIS tools at business and consumer levels. These are content producers, content integrators, content designers, usability experts, content distributors, content aggregators, publishers, etc.

2 Introduction to AXMEDIS

Market and end-users press 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. AXMEDIS will start from the state of the art, taking the industry of content production and integration beyond, reducing the costs of content production and creating an environment in which content providers, producers, integrators and distributors will have access to a huge amount of content at lower cost by exploiting P2P solutions at B2B level.

The ***main*** project objectives of AXMEDIS project are:

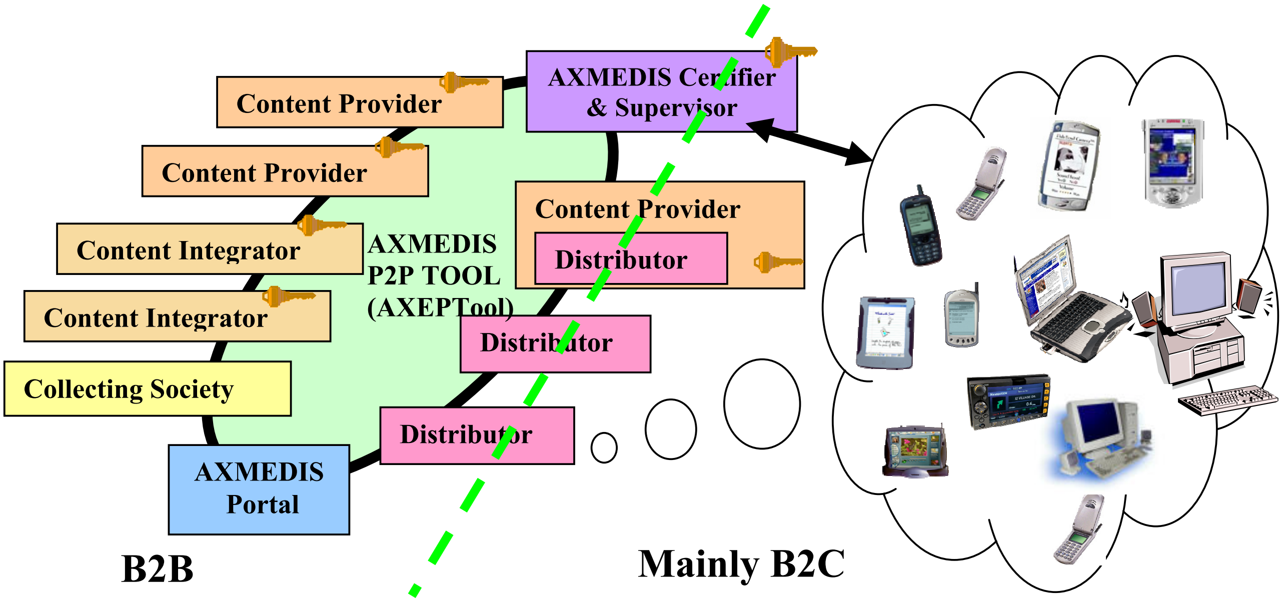
- Reducing the costs of cross media **production** by accelerating the production process with artificial intelligence algorithms for: content composition and formatting, integration and aggregation, and workflow; identification and trial of solutions for accelerating content production and packaging; supporting the standardization process and contributing to MPEG; allowing the reduction of production costs (in the order of 30%) using automatic production of content and production on-demand. The adoption of the AXMEDIS solution and basic technologies in the next 5-8 years by major publishers and distributors and in several of the minor ones;
- Reducing the costs of cross media **distribution** among content producers, aggregators and distributors with a P2P platform at B2B level integrating Content Management Systems and workflows, including digital rights management (DRM) and active protection models supporting different business and transaction models; realization of the AXMEDIS Peer to Peer (P2P) infrastructure and tools for cooperative work on content production;
- Providing algorithms and tools for innovative and flexible **Digital Rights Management**, exploiting MPEG-21 and overcoming its limits, supporting several business and transactions models. Solving difficulties in creating and distributing cross media content generated by using content components coming from different content providers (supporting different Content Management Systems) with the support of a suitable interoperable DRM system. Contributing to these aspects to the MPEG-21 standard. (The project does not fix limitation to the content format and DRM);
- Realization of the AXMEDIS framework including research results, algorithms and tools for content production and distribution to stimulate and support the adoption of the new technologies by SMEs and large companies. The framework will be available for the European Industries and research groups. The new technologies will be open and interoperable to be integrated to present technologies and solutions of the production and distribution chains;
- Developing new knowledge at scientific and technological levels by means of research activities and use these innovative results to reinforce the leadership of Europe in the field of cross media production and distribution, acting in several points of the value chain.

2.1 AXMEDIS main tools

The AXMEDIS digital content and content components (in the following, AXMEDIS content in general) will have a specific format capable of integration inside any kind of cross media format (video, images, animations, document, audio, etc.), adding metadata, identification, classification, categorization, indexing, descriptors, annotation, relationships and play activities and protection aspects. The format will permit the combination of content components, their secure distribution, etc., in the respect of the copyright laws, supporting a large variety of DRM rules and models according to concepts of interoperability among DRMs (mainly, but not only, based on MPEG-21, with both binary and XML low level formats). Within the AXMEDIS content any type of cross media content can be included from simple multimedia files to games, software components, for leisure and entertainment, infotainment, etc.

The General Architecture of AXMEDIS is represented in the next figure, which highlights both:

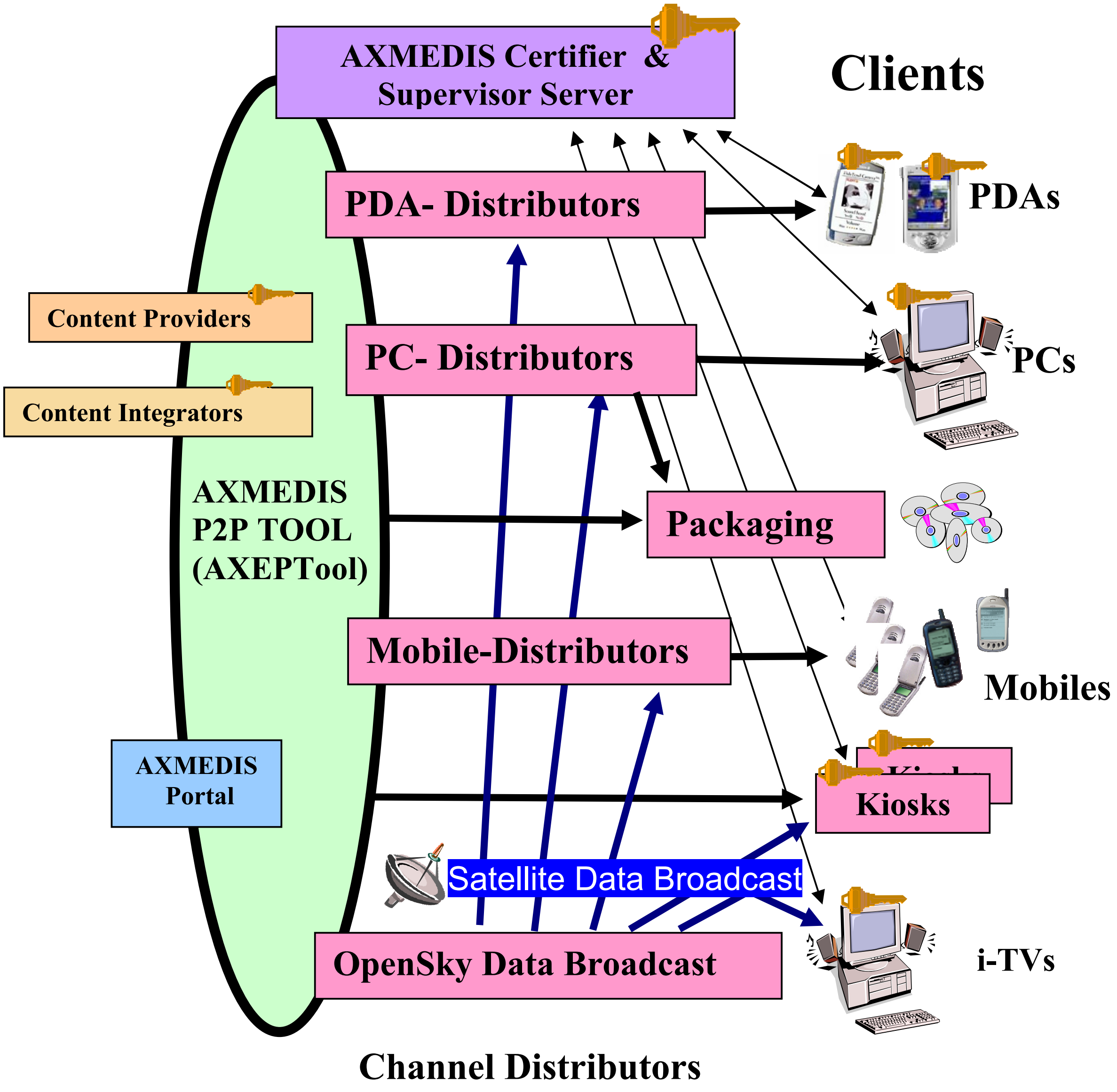
- **production** of AXMEDIS digital content and content components in connection with the AXMEDIS P2P tool (AXEPTool) that follows business mechanisms of B2B and support DRM with a certification authority (AXMEDIS Certifier and Supervisor). This can be connected to the Collection Societies as well as to each Content Provider and to Distributors if needed;
- **distribution** of AXMEDIS digital content towards clients via specific distributors that realize the last level of the distribution chain. This last level can also support a B2B transaction if the distribution is targeted at institutions. Also at this level the sharing via mechanisms of P2P is allowed and stimulated. This will not invalidate the protection model of AXMEDIS DRM.



AXMEDIS General Architecture, mainly B2B side

The standard distribution channel is today a single distribution path for each type of content, and often, multiple proprietary systems of representation for the same content. The definition of distribution channel editorial formats would provide one way, unified and rock-solid content format for multipurpose applications. Alternative solutions support multichannel distribution by using an XML model of content into the Content management systems of the content provider that also include multiple transcoding engines for transforming the XML model of content into the format suitable for the channel. This approach is not flexible enough since the transcoding of content at the source strictly limits the management of Digital Rights. In fact, in models such as CONTESSA the DRM can be applied only to the content in its final version. This creates key problems for the content providers since the content distributors are entitled to receive unprotected content. This is almost unacceptable in most cases.

In AXMEDIS, the channel distributors may maintain their distribution process. They can continue to use the same format for reaching the final users. In AXMEDIS, the content is distributed by using the P2P tool, namely AXEPTool, by using an evolution of the MPEG-21 format, with the AXMEDIS contribution. This content will easily contain and deliver MPEG-4, MPEG formats, PDF, HTML, SVG, images, documents, videos, audio file, etc. (in open standard format for continuation, without the use of proprietary technologies) on demand and for all platforms according to the final format produced by the Distributor. The received content will be formatted by using AXMEDIS tools on the basis of specific editorial formats.



AXMEDIS General Architecture, mainly B2C

The possible Channel Distributors have a large variety of capabilities, they are both of pull and push, and may include off-line and on-line connection from the client to the distributor.

Channel Distributors are interested in:

- Getting AXMEDIS content and components from the Content Providers and using them for distributing content via their channels for redistribution for both B2B and B2C transactions.
- Collecting AXMEDIS contents in a local database for preparing the production content programme that is the agenda/menu proposed to the customers and final users.
- Using AXMEDIS content for creating attractive content for their customers. For this reason, they need to have the possibility of inspecting content in their internal LAN on a client PC.
- Receiving and satisfying requests from their customers for delivering to them the proposed content
- Receiving and satisfying queries performed by their customers that are looking for specific content. This activity is one of the most interesting added value of the AXMEDIS architecture.
- Getting updated information about the possible content that can be recovered from all Content Providers. This activity is performed via a service of the AXMEDIS portal. The updating of the database of the available content is performed in push via satellite data broadcast with specific policies.
- Accessing statistics produced by the AXMEDIS Certifier and Supervisor about the content usage.

Satellite Data Broadcast It is a content distribution mechanism that permits the distribution of the AXMEDIS content in a very efficient manner. This improves the quality of service of the data delivery process (dependent on broadband availability in client location), and Distributors and also PC users can also rely on Satellite Broadcast. This technology, provided by EUTELSAT's Opensky platform, allows large quantities of data to be pushed via satellite directly on the user's PC without congesting local networks. The use of this technology is completely transparent with regard to the AXMEDIS process and only acts as a cost effective and efficient transport mechanism. The same technology also allows the content providers to bring live multimedia streaming content directly to the user's PC either for free to air content (mainly for marketing purposes) or paying on-demand channels. The pushing mechanism can be used to renovate the catalogue of the Distributors periodically at low cost.
This platform appears to be ideally suited for distributing AXMEDIS content and components. It represents an excellent opportunity for content providers for new business and for accelerating the distribution decreasing their costs.

The satellite distribution channel can be used for several activities of content distribution for both B2B and B2C business models:

- The push of content
  - updating the AXMEDIS content and components in the databases of the Distributors and of the Providers;
  - updating the general indexing databases of the Distributors with updated information regarding the available AXMEDIS content and components of the Providers;
  - updating the AXMEDIS content on Kiosks;
  - delivering AXMEDIS content on demand directly to the consumers connected to the satellite i-TV according to their interactive requests;
  - delivering AXMEDIS content to the consumers connected to the satellite i-TV-PC according to their selection performed from the programmed content of the day and week.
- The streaming of AXMEDIS content on MPEG-4 on one or more channels for:
  - Promoting Content Providers' content;
  - Promoting Distributors' services, for example stimulating the acquisition of content in push with a business model based on subscription or pay per view;
  - Creating specific B2B channel with large institutions and consumers.

3 Structure of Test Cases

The test cases will be structured according to structure of the AXMEDIS framework and tools that will be developed in these 18 months of work. The model will be UML including: name, ID, description, functionality to be tested, context, partners involved, Validator(s) skill, data set needed, steps, expected results, variations, issues, additional activities to be considered, metrics to be used, etc.

3.1 Structure of a Test Case

TCId	Unique identifier of the test case
Test case	Name of the test case
Initial conditions	Description of the state of the system before the execution of the test case. This state is the one needed for the correct execution of the test case
Configuration description	Description of configuration conditions, tools involved and connected
Description of functionality to be tested	Functionality to be tested
Partners, people involved	List of people involved in the test, partners, user-groups, other people needed
Validator(s) skill	Skill of the people involved in the test during the validation with end-users
Data set used	Names of or references to the data sets used or their number
Steps	Steps of the test
Expected results	Expected results of the test
Variations	Some changes that can be done for testing some slightly different functionalities
Issues	Other issues, notes, annotations if the Test Case is not clear
Test case Scope/Type	The applicability scope of the test case, such as GUI, backend, etc and the type of the test BlackBox, WhiteBox, UnitTest, and so on

3.2 Content Description (WP4.1: DSI, EPFL, ANSC, ILABS, OD2, TISCALI, XIM, SEJER, UNIVLEEDS, EXITECH, WP4.8: EUTELSAT, OD2, CRS4, SEJER, IRC)

This section describes the content that will appear in the AXMEDIS project. A first tentative definition is provided for some of the content types identified (extracted from RFC 2046: Media types).

- Image: Image data. "Image" requires a display device (such as a graphical display, a graphics printer, or a FAX machine) to view the information.
- Video: Video data. "Video" requires the capability to display moving images, typically including specialised hardware and software. The subtypes of video include: uncompressed (raw video data), compressed (to suit a variety of delivery channels/networks and applications), and image sequences (frames represented by still images). Video content will normally include bundled metadata to describe its properties. Video content may either be managed as discrete self-contained files, or streamed via a transport stream, which has different storage, access and viewing requirements.
- Text: textual information. The subtype "plain" in particular indicates plain text containing no formatting commands or directives of any sort. Plain text is intended to be displayed "as-is". No special software is required to get the full meaning of the text, aside from support for the indicated character set.
- Audio: Audio data. "Audio" requires an audio output device (such as a speaker, headphones or a telephone) to render the contents. The subtypes of audio include: uncompressed (PCM audio data) and compressed (lossless and lossy, to suit a variety of delivery channels/networks and applications). Audio content will normally include bundled metadata to describe its properties. Audio content may either be managed as discrete self-contained files, or streamed via a transport stream, which has different storage, access and listening requirements.
- Bibliographic record: set of data, which could be displayed as a UNIMARC file (plain text as standard exchange format for bibliographic records. It is different from a plain text file because it needs some software to be decoded). Another possible way to display a bibliographic record is an XML file, which is the result of a unimarc conversion. A third possible way is a .mdb file also coming from unimarc.
- PDF file. File, which belongs to Adobe Acrobat application. It embeds images and formatted text. (I'm not sure if this description is correct because a PDF file belongs to Postscript language (which is plain text).
- Multimedia file (.exe Flash, other formats). A game is perhaps a good example. These files will contain scripting code in addition to a mixture of text, images, video, animations and audio to provide a self-contained application, which may also rely on available servers across a cable TV network or the Internet for functionality.
- A postscript file without preview
- A postscript file with preview
- A music sheet file (such as .mus coming from Coda Finale, or files coming from Sibelius, and so on)
- Any other proprietary content format (like autocad (.dwg), Photoshop (.psd), Excel (.xls) Access (.mdb), File maker Pro (.fp5) etc…)
- A compressed file (.zip. .hqx, .sit and so on)
- Disk image file (.img, .iso and so on)
- Metadata:
- AXMEDIS Object: An object generated by the AXMEDIS system. It is derived from the MPEG-21 standard.
- AXMEDIS protected Object: An AXMEDIS object protected by symmetric key encryption.
- Personal data: Data for performing user registration. To be defined.
- Professional data: Data for performing content producer registration. To be defined.
- Data for X.509 certificate:
- Certificate request: PKCS#10
- Asymmetric key pair: A pair of keys to be used with asymmetric encryption / decryption algorithms.
- Symmetric key: A key to be used with symmetric encryption / decryption algorithms for the protection of AXMEDIS objects

For example, the SEJER content could be constituted by

- XML User Interface Language (XUL): A XUL file allows describing a GUI with usual widgets (buttons, trees etc.). Note that XUL content alone has no purpose. It is usually associated with Javascript files and CSS files.
- Javascript: Javascript Files are used to provide interactivity into Web Pages and XUL GUI. Viewing a Javascript file by itself is meaningless. Javascript Files are usually associated to HTML or XUL files. [FUPF] It is not really true, it depends on the context [SEJER] Yes, but when talking about content, Javascript is only a way to provide dynamics to the content.
- Cascading Style Sheets (CSS): CSS files are used to customize XUL GUI or HTML pages aspect. Viewing a CSS file by itself is meaningless. CSS Files are usually associated to HTML or XUL Files
- Resource Description Framework (RDF): RDF is an XML dialect used to describe Graphs. It may serve to describe the data model of an application, or RSS feeds for example. In our case it is only used as data model and has no meaning without a XUL file using it.
- Hypertext Markup Language (HTML/XHTML): HTML pages, possibly dynamic if associated with a JS file
- Macromedia flash: Vector Animations, Videos, Audios using the Macromedia Flash format.
- Scalable Vector Graphics (SVG): SVG is an XML dialect allowing the description of Vector Graphics, possibly with some interactions using Javascript.

Usual Name	**Mime-type**	**Encoding**	**Viewer**	**Editor**	**Description**	**Alternate View**
XUL		Text/UTF-8 Text/ISO-…	Mozilla based Viewer	Text editor	A XUL file allows to describe a GUI with usual widgets (buttons, trees etc.). Note that a XUL content alone has no purpose. It is usually associated with Javascript files and CSS files	An explicitly written GUI can be provided in HTML for non-Mozilla browser, but it will be more poor
Javascript		Text/UTF-8 Text/ISO-…	N/A	Text editor	Javascript Files are used to provide interactivity into Web Pages and XUL GUI. Viewing a Javascript file by itself is meaningless. Javascript Files are usually associated to HTML or XUL files	N/A
CSS		Text/ASCII	N/A	Text editor	CSS files are used to customize XUL GUI or HTML pages aspect. Viewing a CSS file by itself is meaningless. CSS Files are usually associated to HTML or XUL Files	N/A
RDF		Text/UTF-8 Text/ISO-…	N/A	XML Editor, RDF Editor	RDF is an XML dialect used to describe Graphs. It may serve to describe the data model of an application, or RSS feeds for example. In our case it is only used as data model and has no meaning without a XUL file using it.	N/A
HTML/XHTML		Text/UTF-8 Text/ISO-…	Web Browser	Text, HTML editor	HTML pages, possibly dynamic if associated with a JS file	Downgraded HTML viewer on PDAs
Macromedia Flash		Binary	Flash player	Macromedia IDE	Vector Animations, Videos, Audios using the Macromedia Flash format	None
SVG		Text/UTF-8 Text/ISO-…	Adobe SVG player, Mozilla	Text Editor, SVG Editor	SVG is an XML dialect allowing to describe Vector Graphics, possibly with some interactions using Javascript	None

3.3 Content usage inside AXMEDIS

The AXMEDIS Content is:

Prepared/Produced by Content Providers. The content can be created with traditional tools and can be packed, protected and enforced with DRM rules to be distributed and reused as components or complete objects automatically by AXMEDIS tools. This can be done by AXMEDIS tools directly interfacing the AXEPTool with the Content Management System of the content provider, integrator or distributor. Components can be reused for creating other content objects adding more detailed DRM rules, etc. These will take into account the production and distribution phases and the different usages for which the content can be built. Contributions to the improvement of MPEG-21 on these aspects will be performed by the AXMEDIS consortium for the project and to the MPEG forum.

Compounded and formatted in an almost automatic way by using specific tools that will be created in AXMEDIS by research and industrial activities. Composition is the simple action of putting together content component on the basis of rules, while formatting is the process to exploit the contained components in some integrated visualisation/(editorial) format for their distribution and usage from the end user. A simple compounded object comprised of several parts (e.g., an audio, a video and a document), can be formatted using one of several methods according to different formatting styles (graphic layout, temporal scheduling of the content, speech generation from text, etc.) producing final content for i-TV, mobile, PC usage, etc. These activities will be based on content features, generic user profile and needs, specific user profile (in the case of composition on demand), formatting style, optimisation parameters, end-user device profile, interactivity level and paradigms, content type and features, metadata, categorization, business information (price, localization, etc.), temporal evolution, DRM rules, delivering time, etc. The production will be performed by using:

- artificial intelligence algorithms: logic engine interpretation and execution such as TILCO temporal logic (see DSI work);
- optimisation algorithms and tools such as those used for solving knapsack problems in the spatial domain, and scheduling optimisation algorithms in temporal domain: DSI has developed considerable skill in optimising processes on both these domains such as performed in SAMOPROS, OPTAMS, WEDELMUSIC, projects and solutions, etc., by using taboo search, genetic algorithms, knapsack and multi knapsack solutions, several scheduling solutions, MILLA formatting language, etc.;
- synchronisation algorithms and tools. DSI, DIPITA, EPFL and UNIVLEEDS have considerable skill on these aspects used in CARROUSO, WEDELMUSIC, projects and solutions, etc.;
- translation algorithms and tools: leading technology on translating text into several languages, and technology in vocal synthesis form text.

This approach will overcome the problems of simple layout optimisation algorithms that do not take into account contextual aspects and time evolution of content. This will reduce the costs of the production process avoiding trivial repetitive operations. Tools for creating formatting styles and profiles will be produced.

Protected by AXMEDIS tools to be distributed and shared in the AXMEDIS P2P Tool for Collaborative Content Production and Control (AXEPTool). The protection is ensured by using MPEG-21 model for DRM with the support of AXMEDIS Certifier and Supervisors Server and supporting interoperability with other DRMs. The protection technologies will be based on encryption and fingerprint technologies. Fingerprint solutions will be used for controlling and supervising any content on the P2P network controlling the usage of protected and non-protected content and content components. The Encryption technology will be used as an active model of protection. DSI, FUPF, and FHGIGD have highly developed skills in this area on several models of passive and active protection.

Distributed and shared at B2B level among Content Providers, designer, integrators and Distributors. Content Providers provide the digital content and/or content components to be used by other Content Providers for further elaboration and processing and/or by Distributors for reaching the end-users. The business-to-business model among providers and distributors will be based on a P2P (peer to peer) tool for content sharing and cooperative work for production. This tool is called the AXMEDIS P2P Tool for Collaborative Content Production and Control (AXEPTool). Providers and Distributors can be connected to the Internet in any manner. In addition, the AXEPTool keeps track of the available content providers and distributors and of the published/available content for the distribution in the P2P network. The content will be in large part visible (catalogued) from the AXMEDIS portal via a satellite data broadcast. The AXEPTool for P2P activities of content production will also provide a specific user interface for technical queries including business aspects (costs, DRM rules, etc.). DSI, FUPF, AFI and ACIT are highly skilled on these topics.

Certified and supervised by the AXMEDIS Certifier and Supervisors Server, which is the certification authority for DRM. It certifies and verifies (i) the integrity and authenticity of the AXMEDIS content when it is produced, distributed and used (providing keys, etc.), (ii) the transactions performed providing authorizations, etc.; (iii) the integrity and the security of the distributors and clients, and of all devices that are involved in the manipulation and/or usage of the protected AXMEDIS content (preventing the usage on the AXEPTool of non authorised content, the authorisation can be simply obtained by any producers via a registered and certified version of an AXEPTool). The AXMEDIS Certifier and Supervisor keeps trace of the activities performed on each part of the content and components and reports these aspects to the authorized Providers, Distributors and Collecting societies in a sanitized form. This will permit the tracking of the revenues due for each distributed object. Specific statistics and related analyses will be also produced. The AXMEDIS Model will support a large set of different transaction models such as: renting, pay per play, subscription, etc., and the content will have different behaviours according to the DRM rules and user profile.

Distributed toward consumers via the Content Distributors. These distribute AXMEDIS content to their clients via their specific distribution channels (without changing radically their distribution model and tools). Several Distributors can be present in the architecture for distributing content covering different or similar channels. They can cover thematic or territorial areas or groups of clients with specific needs, to also coping with different languages and cultural diversity. For instance impaired people, authors, performers, classic music, jazz music, educational content, cartoons, etc. Each Content Distributor may collect and redistribute content provided by all the Content Providers. The Distributors can distribute content according to the authorization/rules associated with the content itself and may add its specific rules constraining them and changing prices. The protection model will support the distribution via P2P network or via traditional B2C transaction models. The distribution is to:

- PC clients:
  - local/internal clients on PCs connected to the Distributor via a LAN (for covering needs of archives, libraries, production process of distributors, schools, etc.);
  - internet clients on PCs connected to Distributor via Internet (traditional or satellite connection);
- Mobile clients (PDAs or Cellular phones):
  - Remotely connected to a Content Distributor via UMTS, GPRS, etc.;
  - Locally connected to a Content Distributor via Wireless LAN (e.g., WiFi, Bluetooth);
- i-TV clients connected via
  - Satellite Data Broadcast (DVBS) with a PC, mainly receiving content in Push on a specific channel;
  - Open Sky distribution on their Set top Box (STB) or other device mainly receiving content in streaming;
  - Terrestrial Data Broadcast (DVBT) with a PC or a STB receiving content in push or streaming respectively;
- In all cases, the content can be also received off-line by the clients via
  - CDs, DVDs produced by some Distributor or friends;
  - Simple files from other consumers with other communication mechanisms, IRDA, LAN, etc.

Listed and Managed in terms of metadata and content evolution, business and technical information by the AXMEDIS Portal service in conjunction to the AXEPTool. This also collects the description and metadata of all the AXMEDIS content and content components that are published on the P2P architecture and network. A satellite data broadcast will be available for distributing information and content rapidly among Providers and Distributors connected with the AXEPTool. This will reduce the time and costs for downloading. It could work as a proxy for the whole information in the B2B network. The AXEPTool in conjunction with AXMEDIS Certifier and Supervisor will be capable of tracing about the evolution of each single digital object, this allows the workflow monitoring of content production, tracking versioning and digital rights exploitation. The AXMEDIS Portal is also a way to provide a large number of other services for the community and supporting the AXMEDIS Framework construction and management.

Searched into the distributed database managed by AXEPTool allowing the specification of technical/professional query including metadata, technical information, business and licensing aspects, content based, DRM rules, etc. In the demonstrators a more specific query engine will be developed to simplify the access to content for PDAs and Mobiles phones on the B2C side. In those cases, the information needed to identify the single object into a database is more related to the content description and metadata rather than to technical details.

Used and consumed by AXMEDIS clients. They can be of several different natures and according to that delivered in different formats plus encrypted parts (HTML, WAP, SCORM, LMO, MPEG-4, etc.). In some cases, they have to be certified/registered by the AXMEDIS Certifier and Supervisor Server, if they intend to manipulate protected content. In other cases, they are already identified by their internal card, for example in the case of cellular phones. The client tools can be implemented as Viewers that can be integrated in classical browsers with a plug in, Java Applet or ActiveX compliant with the AXMEDIS directives or with specific tools. Standards viewers, provided by the consortium via the portal, can be customized at level of user interface, language and skin for specific purposes.

4 AXMEDIS Object editing (DSI, EXITECH, EPFL)

4.1 AXMEDIS Editors, as authoring tools (WP4.1.3, WP5.4.4: DSI)

4.1.1 Creation of a new AXMEDIS object

TCId	TC4.1.1
Test case	Creation of a new AXMEDIS object
Initial conditions	• AXMEDIS Editor is open • The user is someway identified in the system
Configuration description	None
Description of functionality to be tested	The user creates a new AXMEDIS object from scratch, i.e. an empty object
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	None
Steps	1 The user clicks on the "New object" buttons within the AXMEDIS Editor main window 2 The system shows to the user a hierarchical view/editor of the new object. The hierarchical view contains only one node representing the document without any child node, i.e. the node is not expandable (e.g. in Windows system the symbol '+' does not appear near the node) 3 The user opens the metadata view/editor on the unique node
Expected results	The value of the object creator metadata is the id of the user
Variations	• The user clicks on "New…" within the "File" menu of the application
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.2 Load and save AXMEDIS objects

4.1.2.1 Load an AXMEDIS object

TCId	TC4.1.2.1
Test case	Load an AXMEDIS object
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system
Configuration description	The Local AXMEDIS Database is up and reachable
Description of functionality to be tested	AXMEDIS Editor correctly loads an AXMEDIS object
Partners, people involved	Content Provider, Content Integrator
Validator skill	Querying of XML and relational database
Data set used	AXDS-Editor1
Steps	1 The user clicks on the "Open object" buttons within the AXMEDIS Editor main window 2 The AXMEDIS Editor shows a dialog to allow the actor to choose which object he/she wants to open. 3 The user choose, using the dialog box, the file he/she wants to load from the file-system 4 If errors did not occur in opening the object

	4.1 The AXMEDIS Editor shows to the user a hierarchical view/editor, which shows one node representing the whole document. 4.2 The user can navigate through the object to verify its consistency 5 Else 5.1 The AXMEDIS Editor shows a dialog to inform the user about what did not correctly work
Expected results	The object representation coincides with the matter contained in the AXMEDIS object
Variations	Load an AXMEDIS object from the Local AXMEDIS Database or the Crawling Results Integrated Database: • The user clicks on the AXQS pane of the dialog box. The user composes the query to look for the wanted objects and selects the object he/she wants to load, independently by the source the object come from
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.2.2 Save an AXMEDIS object

TCId	TC4.1.2.2
Test case	Save an AXMEDIS object
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • An AXMEDIS Object is opened within the AXMEDIS Editor • The object has been loaded and has not been newly created (see Test Case "Creation of a new AXMEDIS object") • The object has been modified (see Test Cases "Adding AXMEDIS elements to an existing AXMEDIS object", "Removing an element from an AXMEDIS Object", etc…)
Configuration description	None
Description of functionality to be tested	AXMEDIS Editor saves a previously loaded and modified AXMEDIS object on the local file-system
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor1
Steps	1 The user clicks on the "Save object" buttons within the AXMEDIS Editor main window 2 If the object is valid, and the user has the rights to save the object 2.1 The AXOM overwrites the old object with the modified one 2.2 The user closes the saved object and reload it 2.3 The user verifies the consistency of the object (see Test Case "Load an AXMEDIS object") 3 Else 3.1 The AXMEDIS Editor shows a dialog to inform the user about what did not correctly work
Expected results	The reloaded object contains the modification made on the original object
Variations	• If the object has been newly created than you can refer to Test Cases "Save an AXMEDIS object as (new location on file-system)" and "Save an AXMEDIS object as (new location within local AXDB)"
Issues	Note that the object is saved on the location it comes from, whichever is: local file-system or local AXMEDIS Database

Test case Scope/Type	GUI, BlackBox

4.1.2.3 Save an AXMEDIS object as (new location on file-system)

TCId	TC4.1.2.3
Test case	Save an AXMEDIS object as (new location on file-system)
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • An AXMEDIS Object is opened within the AXMEDIS Editor • The object has been newly created or has been loaded from whichever possible source (see Test Case "Load an AXMEDIS object")
Configuration description	Local AXDB is up
Description of functionality to be tested	AXMEDIS Editor saves a previously loaded AXMEDIS object in a new location on the local file-system
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor1
Steps	1 The user clicks on the "Save object as…" buttons within the AXMEDIS Editor main window 2 The AXMEDIS Editor asks the user to input the new location. 3 The user choose the new location on the local file-system 4 If the object is valid, and the user has the rights to save the object in a new location 4.1 The AXOM saves the object in the new location 4.2 The user closes the saved object 4.3 The user loads the original object from the original location and the saved object from the new location 4.4 The user verifies the consistency of the two objects (see Test Case "Load an AXMEDIS object") 5 Else 5.1 The AXMEDIS Editor shows a dialog to inform the user about what did not correctly work
Expected results	The original object (if exists, i.e. the object has been loaded and not newly created) is not changed The saved object contains the modification made on the original object
Variations	None
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.2.4 Save an AXMEDIS object as (new location within local AXDB)

TCId	TC4.1.2.4
Test case	Save an AXMEDIS object as (new location within local AXDB)
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • An AXMEDIS Object is opened within the AXMEDIS Editor • The object has been newly created or has been loaded from whichever possible source except the local AXDB (see Test Case "Load an AXMEDIS object")
Configuration description	None

Description of functionality to be tested	AXMEDIS Editor saves a previously loaded AXMEDIS object in a new location on the local file-system
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor1
Steps	1 The user clicks on the "Save object as…" buttons within the AXMEDIS Editor main window 2 The AXMEDIS Editor asks the user to input the new location. 3 The user choose the new location within the local AXDB 4 If the object is valid, and the user has the rights to save the object in a new location 4.1 The AXOM saves the object in the new location 4.2 The user closes the saved object 4.3 The user loads the original object from the original location and the saved object from the new location 4.4 The user verifies the consistency of the two objects (see Test Case "Load an AXMEDIS object") 5 Else 5.1 The AXMEDIS Editor shows a dialog to inform the user about what did not correctly work
Expected results	The original object (if exists, i.e. the object has been loaded and not newly created) is not changed The saved object contains the modification made on the original object
Variations	None
Issues	An object loaded from the AXDB cannot be saved in a new location in the same AXDB. This kind of operation has no sense.
Test case Scope/Type	GUI, BlackBox

4.1.3 Adding AXMEDIS elements to an existing AXMEDIS object

TCId	TC4.1.3
Test case	Adding AXMEDIS elements to an existing AXMEDIS object
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • A hierarchy view of the object is open
Configuration description	None
Description of functionality to be tested	None
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor2
Steps	1 The user clicks with the right mouse button on an existing element represented by a node in the hierarchal view/editor 2 The hierarchal view shows the proper context menu to the user 3 The user chooses "Add element…" and then chooses the type of element he/she wants to add 4 If necessary, the hierarchal view shows to the user a dialog to fill the element attributes and options

	5 The user confirms the operation 6 If the user has the needed rights: 6.1 The element is added and the hierarchal view shows to the user a new node representing the element in the chosen position 6.2 To verify the modification have been really made the user has to execute Test Case “Save an AXMEDIS object” 7 Else: 7.1 The system shows a dialog box to inform the user about why he/she cannot add the element
Expected results	None
Variations	• The user clicks on “Add element…” within the “Edit” menu of the application instead of using the context menu • The user adds an element as “brother” of an existing element instead as child of a given element. That should be possible by choosing “Insert after…”/“Insert before…” from the “Edit” menu or the context menu (of the reference element)
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.4 Extracting an element from an AXMEDIS Object

TCId	TC4.1.4
Test case	Extracting an element from an AXMEDIS Object
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • A hierarchy view of the object is open • The object contains at least one element
Configuration description	None
Description of functionality to be tested	None
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor3
Steps	1 The user clicks with the right mouse button on an existing element represented by a node in the hierarchal view/editor 2 The hierarchal view shows the proper context menu to the user 3 The user chooses “Extract element…” 4 The hierarchal view shows a dialog to allow the user to choose the location (into the local file-system, into the AXMEDIS Database, etc…) where extracted element should be stored. Moreover the user can choose if he/she want to extract the element in clear or still protected 5 The user confirms the operation 6 If the user has the needed rights: 6.1 A new AXMEDIS object containing the selected elements and all related information (e.g. DRM, etc…) is created in the given location 6.2 To verify the extraction have been really made the user has to execute Test Case “Load an AXMEDIS object” on the location the element have been stored to 7 Else: 7.1 The system shows a dialog box to inform the user about why he/she

	cannot extract the element (e.g. he/she wants to extract it in clear and he/she does not have the requested rights)
Expected results	None
Variations	The user clicks on "Extract element…" within the "Edit" menu of the application instead of using the context menu
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.5 Removing an element from an AXMEDIS Object

TCId	TC4.1.5
Test case	Removing an element from an AXMEDIS Object
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • A hierarchy view of the object is open • The object contains at least one element
Configuration description	None
Description of functionality to be tested	None
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor4
Steps	1 The user clicks with the right mouse button on an existing element represented by a node in the hierarchal view/editor 2 The hierarchal view shows the proper context menu to the user 3 The user chooses "Remove" and then chooses the type of element he/she wants to add 4 The user confirms the operation 5 If the user has the needed rights: 5.1 The element is removed and the hierarchal view shows to the user the modified representation of the object 5.2 To verify the modification have been really made the user has to execute Test Case "Save an AXMEDIS object" 6 Else: 6.1 The system shows a dialog box to inform the user about why he/she cannot remove the element
Expected results	None
Variations	The user clicks on "Remove" within the "Edit" menu of the application instead of using the context menu
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.6 Moving an element within the AXMEDIS Object

TCId	TC4.1.6
Test case	Adding AXMEDIS elements to an existing AXMEDIS object
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • A hierarchy view of the object is open • The object contains at least two elements, one to be moved and another to be used as reference of the move

Configuration description	None
Description of functionality to be tested	None
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor5
Steps	1 The user clicks on an element and drags it 2 When the Actor drops the element, releasing the mouse button, the Hierarchy View controls if the chosen position is an allowed one. 3 If the position is a valid one and the user is allowed to move the element in the new position: 3.1 The element is moved and the hierarchal view shows to the user the element has been removed from the old position and the element has been added in the chosen position 3.2 To verify the modification have been really made the user has to execute Test Case “Save an AXMEDIS object” 4 Else: 4.1 The system shows a dialog box to inform the user about why he/she cannot add the element
Expected results	None
Variations	None
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.7 Adding a resource

TCId	TC4.1.7
Test case	Adding a resource
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • A hierarchy view of the object is open
Configuration description	None
Description of functionality to be tested	None
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor1
Steps	1 The user clicks with the right mouse button on an existing element which can contain a resource element 2 The hierarchal view shows the proper context menu to the user 3 The user chooses “Add element…” and then chooses to add a resource element 4 The hierarchal view shows to the user a dialog which allows him/her to select the resource to put into the AXMEDIS object 5 The user confirms the operation 6 If the user has the needed rights on the AXMEDIS object and on the resource (i.e. the resource is a whatever type of governed digital item): 6.1 The resource element is added and the hierarchal view shows to the user a

	new node representing the element in the chosen position 6.2 To verify the modification have been really made the user has to execute Test Case "Save an AXMEDIS object" 7 Else: 7.1 The system shows a dialog box to inform the user about why he/she cannot add the resource element
Expected results	None
Variations	• The user clicks on "Add element…" within the "Edit" menu of the application instead of using the context menu • The user adds an element as "brother" of an existing element instead as child of a given element. That should be possible by choosing "Insert after…"/"Insert before…" from the "Edit" menu or the context menu (of the reference element)
Issues	None
Test case Scope/Type	GUI, BlackBox

4.1.8 Managing/Modifying a resources

TCId	TC4.1.8
Test case	Managing/Modifying a resources
Initial conditions	• AXMEDIS Editor is open • User is someway identified in the system • A hierarchy view of the object is open • The object contains at least one resource element
Configuration description	None
Description of functionality to be tested	None
Partners, people involved	Content Provider, Content Integrator
Validator skill	None
Data set used	AXDS-Editor6
Steps	1 The user clicks with the right mouse button on an existing resource element 2 The hierarchal view shows the proper context menu to the user 3 If it is available, the user chooses "Open" (i.e. the resource mime type is related to a editor/viewer) 3.1 The editor related to the mime type of the resource is opened 3.2 The user someway modifies the resource using the editor. DRM rules respect is enforced by the editor itself 3.3 After the user closes the editor, the previously extracted is updated with the modified resource 3.4 To verify the modification have been really made the user has to execute Test Case "Save an AXMEDIS object" 4 Else, the user chooses "Open with…" 4.1 The system shows the list of all available editors 4.2 The user chooses the editor he/she wants to associate to the mime type of the resource 4.3 The Test Cases continues from step 3.1
Expected results	None
Variations	• The user double clicks on the resource element
Issues	None

Test case Scope/Type	GUI, BlackBox

4.2 AXMEDIS Internal Viewers (DSI)

4.2.1 Invoking an internal viewer/editor

TCId	TC4.2.1
Test case	Invoking an internal viewer/editor
Initial conditions	• AXMEDIS Editor is open • An object is opened within the AXMEDIS Editor • An hierarchical view of the object is open
Configuration description	AXMEDIS Editor
Description of functionality to be tested	Invoking an internal viewer/editor to view or manipulate an object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	AXDS-IVE1
Steps	1 The actor clicks with the right mouse button on an resource 2 The Editor shows the proper context menu to the actor 3 The actor chooses "View…" 4 The proper viewer/editor is associated with the resource on the basis of MIME type 5 The system sends an opening authorization request to the PMS (via AXOM) 6 If PMS does not provide the authorization 6.1 The system displays an authorization failure message on screen 6.2 The Test Case ends 7 The system performs the verification of the AXMEDIS Editor 8 If the verification is not valid 8.1 The system displays an verification failure message on screen 8.2 The Test Case ends 9 The system activates the proper internal viewer. 10 The internal viewer/editor shows the digital resource
Expected results	The internal viewer/editor shows the digital resource The Editor shows failure messages if the internal viewer/editor is not authorised to display the resource
Variations	None
Issues	None
Test case Scope/Type	GUI, BlackBox

4.2.2 Managing a digital resource by respecting the DRM in an Internal Viewer/Editor

TCId	TC4.2.2
Test case	Managing the digital resource by respecting the DRM in an Internal Viewer/Editor
Initial conditions	The external tool is running and displaying a resource belonging to an AXMEDIS object.
Configuration description	An internal viewer has been invoked by the system
Description of functionality to be tested	Respecting of DRM in the internal Viewer/Editor
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer

Validator skill	Production editing, GUI user
Data set used	AXDS-IVE2
Steps	1 The Actor wants to perform a command on the digital resource 2 The system verifies the DRM of the resource (i.e. if the actor has the right to perform such command) 3 If the user is authorised 3.1 The internal viewer/editor performs the command 4 Else 4.1 The internal viewer/editor notifies a command failure message.
Expected results	The command is performed A dialog displaying an authorisation failure message
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

4.2.3 Closing an Internal viewer/editor

TCId	TC4.2.3
Test case	Closing an Internal viewer/editor
Initial conditions	An Internal viewer/editor is running and displaying a resource belonging to an AXMEDIS object.
Configuration description	An internal viewer has been invoked by the system
Description of functionality to be tested	Closing an Internal viewer/editor and updating of the resource
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	AXDS-IVE2
Steps	1 The actor wants to quit the Internal viewer/editor 2 The user clicks with left mouse button on the close button of the Internal viewer/editor 3 If the digital resource is changed 3.1 The Internal viewer/editor displays a dialog asking for the modification acceptance. 3.2 If the actor does not discard the modification 3.2.1 The resource is updated
Expected results	The Internal viewer/editor is correctly closed The resource has been updated
Variations	• The actor could quit the Internal viewer/editor by selecting “Quit” in the menu bar.
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

4.3 AXMEDIS tools for using / producing AXMEDIS objects in other content tools (WP4.1.3: DSI, WP4.1.4: EPFL)

4.3.1 Invoking external tools with a digital resource belonging to the AXMEDIS object

TCId	TC4.3.1
Test case	Invoking external tools with a digital resource belonging to the AXMEDIS object
Initial conditions	• AXMEDIS Editor is open • An object is opened within the AXMEDIS Editor

	• An hierarchical view of the object is open • The tool that will be invoked is equipped with the AXEMDIS plug-in
Configuration description	AXMEDIS Editor, external tool
Description of functionality to be tested	Invoking an external tool to view or manipulate an object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	AXDS-IVE1
Steps	1 The actor clicks with the right mouse button on an resource 2 The Editor shows the proper context menu to the actor 3 The actor chooses "Open with…" 4 The proper viewer/editor is associated with the resource on the basis of MIME type 5 The system sends an opening authorization request to the PMS (via AXOM) 6 If PMS does not provide the authorization 6.1 The system displays an authorization failure message on screen 6.2 The Test Case ends 7 The system performs the verification of the AXMEDIS Editor 8 If the verification is not valid 8.1 The system displays an verification failure message on screen 8.2 The Test Case ends 9 The system calls the external tool associated with the resource 10 The external tool is configured by the AXMEDIS plug-in according to the DRM rules associated with the digital resource 11 The external tool shows the digital resource
Expected results	The external tool shows the digital resource The Editor shows failure messages if the tool is not authorised to display the resource
Variations	None
Issues	None
Test case Scope/Type	GUI, BlackBox

4.3.2 Managing the digital resource by respecting the DRM in an external tool

TCId	TC4.3.2
Test case	Managing the digital resource by respecting the DRM in an external tool
Initial conditions	The external tool is running and displaying a resource belonging to an AXMEDIS object.
Configuration description	• An external tool has been invoked by the system • The external tool uses the AXMEDIS plug-in • The communication with the AXMEDIS Editor is active via plug-in
Description of functionality to be tested	Respecting of DRM in the external tool
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	AXDS-IVE2
Steps	1 The actor wants to execute a command provided by the external tool 2 The AXMEDIS plug-in verifies the DRM of the resource (i.e. if the actor has the right to perform such command)

	3 If the actor is authorised 3.1 The AXMEDIS plug-in authorises the External tool to perform the command 4 Else 4.1 The AXMEDIS plug-in does not authorise the external tool to execute the command and notifies a command failure message.
Expected results	1. The command is performed 2. A dialog displaying an authorisation failure message
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

4.3.3 Closing External Tool

TCId	TC4.3.3
Test case	Closing External Tool
Initial conditions	• The external tool is running and displaying a resource belonging to an AXMEDIS object.
Configuration description	• An external tool has been invoked by the system • The external tool is using the AXMEDIS plug-in
Description of functionality to be tested	Closing the external tool and updating of the resource
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	AXDS-IVE2
Steps	1 The actor wants to quit the external tool 2 The user clicks with left mouse button on the close button of the external tool menu 3 If the digital resource is changed 3.1 The tool displays a dialog asking for the modification acceptance. 3.2 If the actor does not discard the modification 3.2.1 The resource is updated 4 The tool is closed
Expected results	The tool is correctly closed The resource has been updated
Variations	• The actor could quit the tool by selecting “Quit” in the menu bar.
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5 AXMEDIS Production Tools (DSI, EXITECH, EPFL)

5.1 Compositional Tools (WP4.3.1: DSI, WP5.4.1: DSI)

5.1.1 Compositional Engine (WP4.3.1: DSI, WP5.4.1: DSI)

5.1.1.1 Starting the Execution of a Rule

TCId	TC5.1.1.1
Test case	Starting the Execution of a Rule
Initial conditions	The composition engine is in a ready status and some compositional rules have been activated. The internal scheduler of the composition engine is monitoring the system clock or is listening for an external command to execute on time or on demand one or more schedules and composition rules for actual consumption.
Configuration description	Engine is active and accessing correct system time.
Description of functionality to be tested	Automatic composition of an AXMEDIS object on time or on demand.
Partners, people involved	Content owner, Content Integrator, Content Distributor, AXMEDIS Workflow Manager
Validator skill	People involved with the end-user have the appropriate familiarity with the rule engine and with script languages.
Data set used	AXDS-Composition1
Steps	1 The internal scheduler sends a Composition Rule execution request and the corresponding rule to the Rule Executor. 2 The engine runs the submitted rule
Expected results	The Composition Engine runs the rule.
Variations	• The activation of the engine is performed by the internal scheduler according the time information for rule execution • The activation of the engine is performed by simulating an external command (composition request) coming from the AXMEDIS Workflow Manager
Issues	None
Test case Scope/Type	Backend/WhiteBox

5.1.1.2 Running a rule

TCId	TC5.1.1.2
Test case	Running a rule
Initial conditions	A rule is ready to be run.
Configuration description	AXMEDIS Composition Engine, AXOM and AXMEDIS production tools (Fingerprint, Adaptation, Protection).
Description of functionality to be tested	Communication with the AXOM and interactions with production tools (Fingerprint, Adaptation, Protection).
Partners, people involved	Content owner, Content Integrator, Content Distributor, AXMEDIS Workflow Manager
Validator skill	People involved with the end-user have the appropriate familiarity with the rule engine and with script languages.
Data set used	AXDS-Composition1
Steps	1 The engine runs the rule. 2 If an error occurs during the execution 2.1 End process notification with failure 2.2 The composition is interrupted

	2.3 The test case ends 3 The engine completes the rule execution 4 A new AXMEDIS object has been stored into AXMEDIS Database (AXMEDIS Objects repository) 5 End process notification with success
Expected results	End process notification (failure or success) Production and storing of AXMEDIS object into AXMEDIS Database
Variations	• Using rules on different Selections for objects retrieval from AXMEDIS database • Using rules on different Selections of objects with DRM incompatible with the specified DRM target: the engine notifies the incompatibility • Using rules on different Selections of objects with Potential Available Rights incompatible with rights that rules have to assign with the compound AXMEDIS object: the engine notifies the incompatibility • Using rules on different Selections of objects with different types of contents such as audio, images, video etc… for interaction with the Fingerprint tool (extraction of Content identifiers and descriptors) • Using rules on different Selections of objects with different types of contents such as audio, images, video etc… for the interaction with the Adaptation tool (digital resource adaptation: scaling, changing format, filtering, etc…)
Issues	None
Test case Scope/Type	Backend/WhiteBox

5.1.2 Composition Rules Editor (WP4.3.1: DSI, WP5.4.1: DSI)

5.1.2.1 Create a new compositional rule

TCId	TC5.1.2.1
Test case	Create a new compositional rule
Initial conditions	The AXMEDIS Composition Rules Editor is running
Configuration description	Composition Rules Editor is running
Description of functionality to be tested	Verification that the user can write and store a compositional rule. The user should be able to write rules easily thanks the suitable user interface made available by Composition Rules Editor.
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Data set used	AXDS-Composition2
Steps	1 The User uses GUI to submit query for objects and is returned a list of results 2 The User selects part/all/none of results using the GUI 3 The User writes a new rule or updates an existing rule 4 The User writes or updates the schedule associated with the rule 5 The User stores the created rule into Composition Rules Database
Expected results	Defined rules for composition defaulted as “inactive” and saved in repository
Variations	• The User defines a Selection by writing in the rule the scripting code (Composition Rule Language) for queries to be executed when the rule will be run • The User can define a rule or writing it as scripting code (Composition Rule Language) or in a Visual way.
Issues	None

Test case Scope/Type	GUI/Whitebox

5.1.2.2 Select a compositional rule

TCId	TC5.1.2.2
Test case	Select a compositional rule
Initial conditions	The AXMEDIS Composition Rules Editor is running
Configuration description	Composition Rules Editor
Description of functionality to be tested	Rule loading and visualisation
Partners, people involved	List of people involved in the test, partners, user-groups, other people needed
Validator skill	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Data set used	AXDS-Composition2
Steps	1 The user browses the existing rules in the collection 2 The user selects and loads the chooses rule 3 The rule is displayed on the screen 4 The user can use the "Create a new compositional rule" UC to edit the rule
Expected results	New rules are collected and saved in the collection of rules
Variations	None
Issues	None
Test case Scope/Type	GUI/WhiteBox

5.1.2.3 Activating a compositional rule

TCId	TC5.1.2.3
Test case	Activating a compositional rule
Initial conditions	Set of complete rules defined and set as inactive
Configuration description	The User has completed a composition rule editing and wants to set the rule to be executed (as active)
Description of functionality to be tested	Activation of a compositional rule
Partners, people involved	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Validator skill	Production editing
Data set used	AXDS-Composition3
Steps	1 The User searches into the Repository of Compositional Rules a specific compositional rule 2 If the User doesn't found the rule 2.1 The User can create a new one (see "Create a new compositional rule" UC) 3 The User selects "Activate Rule" function 4 A confirmation on the success of the activation is provided
Expected results	Compositional rules set to active and submitted to the Active Composition Rules repository
Variations	• User can modify/cancel this action before the activation
Issues	None
Test case Scope/Type	GUI/Whitebox

5.1.2.4 Removing an active compositional rule

TCId	TC5.1.2.4
Test case	Removing a compositional rule
Initial conditions	Set of complete rules defined and set as active
Configuration description	The user open the compositional rule editor to remove an active rule
Description of functionality to be tested	Active compositional rule removal
Partners, people involved	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Validator skill	Production editing
Data set used	AXDS-Composition4
Steps	1 The User requests the list of Active Rules in the Active Rules Repository 2 The User selects the active rule to be disabled 3 The User selects "Remove Rule" function 4 The rule is Removed
Expected results	A selected compositional rule is removed from the Active Rules Repository
Variations	None
Issues	None
Test case Scope/Type	GUI/Whitebox

5.1.2.5 Debugging a compositional rule

TCId	TC5.1.2.5
Test case	Debugging a compositional rule
Initial conditions	Set of complete rules defined and set as active
Configuration description	The compositional rule editor is running and a rule is displayed on the screen.
Description of functionality to be tested	Debugging of rules
Partners, people involved	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Validator skill	Production editing
Data set used	AXDS-Composition2
Steps	1 The User load a rule 2 The User chooses the debugging rule mode 3 The Rule Editor enters in the Debugging Mode 4 During the debugging mode the User: 4.1 Check the statements of rule step by step 4.2 Control the values of current variables 5 Exit from the debugging mode
Expected results	The debugging has been successfully executed.
Variations	• The user has written a new rule and wants to debug it
Issues	None
Test case Scope/Type	GUI/Whitebox

5.2 Formatting tools (WP4.3.2: DSI, WP5.4.2: DSI)

5.2.1 Formatting Engine (WP4.3.2: DSI, WP5.4.2: DSI)

5.2.1.1 Starting the Execution of a Rule

TCId	TC5.2.1.1
Test case	Starting the Execution of a Rule
Initial conditions	The formatting engine is in a ready status and some formatting rules have been activated. The internal scheduler of the formatting engine is monitoring the system clock or is listening for an external command to execute on time or on demand one or more schedules and formatting rules for actual consumption.
Configuration description	Engine is active and accessing correct system time.
Description of functionality to be tested	Automatic formatting of an AXMEDIS object on time or on demand.
Partners, people involved	Content owner, Content Integrator, Content Distributor, AXMEDIS Workflow Manager
Validator skill	People involved with the end-user have the appropriate familiarity with the rule engine and with script languages.
Data set used	AXDS-Formatting1
Steps	1 The internal scheduler sends a Formatting Rule execution request and the corresponding rule to the Rule Executor. 2 The engine runs the submitted rule
Expected results	The Formatting Engine runs the rule.
Variations	• The activation of the engine is performed by the internal scheduler according the time information for rule execution • The activation of the engine is performed by simulating an external command (formatting request) coming from the AXMEDIS Workflow Manager
Issues	None
Test case Scope/Type	Backend/WhiteBox

5.2.1.2 Running a rule

TCId	TC5.2.1.2
Test case	Running a rule
Initial conditions	A rule is ready to be run.
Configuration description	AXMEDIS Formatting Engine, AXOM and AXMEDIS production tools (Fingerprint, Adaptation, Protection, External tools).
Description of functionality to be tested	Communication with the AXOM and interactions with production tools (Fingerprint, Adaptation, Protection, External tools).
Partners, people involved	Content owner, Content Integrator, Content Distributor, AXMEDIS Workflow Manager
Validator skill	People involved with the end-user have the appropriate familiarity with the rule engine and with script languages.
Data set used	AXDS-Formatting2
Steps	1 The engine runs the rule. 2 If an error occurs during the execution 2.1 End process notification with failure 2.2 The formatting is interrupted 2.3 The test case ends 3 The engine completes the rule execution 4 A new formatted AXMEDIS object has been stored into AXMEDIS Database

	(AXMEDIS Objects repository) 5 End process notification with success
Expected results	End process notification (failure or success) Production and storing of AXMEDIS object into AXMEDIS Database
Variations	• Using rules on different Selections for objects retrieval from AXMEDIS database • Using rules on different Selections of objects with DRM incompatible with the specified DRM target: the engine notifies the incompatibility • Using rules on different Selections of objects with Potential Available rights incompatible with rights that rules have to assign with the formatted AXMEDIS object: the engine notifies the incompatibility • Using rules on different Selections of objects with different types of contents such as audio, images, video etc… for interaction with the Fingerprint tool (extraction of Content identifiers and descriptors) • Using rules on different Selections of objects with different types of contents such as audio, images, video etc… for the interaction with the Adaptation tool (digital resource adaptation: scaling, changing format, filtering, etc…) • Using rules on different Selections of objects with different types of contents such as audio, images, video etc… for the interaction with the External tools (e.g. Adobe tools, Macromedia tools, etc…)
Issues	None
Test case Scope/Type	Backend/WhiteBox

5.2.2 Formatting Rules Editor (WP4.3.2, WP5.4.2)

5.2.2.1 Create a new formatting rule

TCId	TC5.2.2.1
Test case	Create a new formatting rule
Initial conditions	The AXMEDIS Formatting Rules Editor is running
Configuration description	Formatting Rules Editor is running
Description of functionality to be tested	Verification that the user can write and store a formatting rule. The user should be able to write rules easily thanks the suitable user interface made available by Formatting Rules Editor.
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Data set used	AXDS-Formatting2
Steps	1 The User uses GUI to submit query for objects and is returned a list of results 2 The User selects part/all/none of results using the GUI 3 The User writes a new rule or updates an existing rule 4 The User writes or updates the schedule associated with the rule 5 The User stores the created rule into Formatting Rules Database
Expected results	Defined rules for formatting defaulted as “inactive” and saved in repository
Variations	• The User defines a Selection by writing in the rule the scripting code (Formatting Rule Language) for queries to be executed when the rule will be run • The User can define a rule or writing it as scripting code (Formatting Rule Language) or in a Visual way.
Issues	None

Test case Scope/Type	GUI/WhiteBox

5.2.2.2 Select a formatting rule

TCId	TC5.2.2.2
Test case	Select a formatting rule
Initial conditions	The AXMEDIS formatting Rules Editor is running.
Configuration description	Formatting Rules Editor, set of complete rules defined
Description of functionality to be tested	Rule loading and visualisation
Partners, people involved	List of people involved in the test, partners, user-groups, other people needed
Validator skill	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Data set used	AXDS-Formatting2
Steps	1 The user browses the existing rules in the collection 2 The user selects and loads the chooses rule 3 The rule is displayed on the screen 4 The user can use the “Create a new formatting rule” UC to edit the rule
Expected results	New rules are collected and saved in the collection of rules
Variations	None
Issues	None
Test case Scope/Type	GUI/WhiteBox

5.2.2.3 Activating a formatting rule

TCId	TC5.2.2.3
Test case	Activating a formatting rule
Initial conditions	Set of complete rules defined and set as inactive
Configuration description	The User has completed a formatting rule editing and wants to set the rule to be executed (as active)
Description of functionality to be tested	Activation of a formatting rule
Partners, people involved	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Validator skill	Production editing
Data set used	AXDS-Formatting3
Steps	1 The User searches into the Repository of Formatting Rules a specific formatting rule 2 If the User doesn’t found the rule 2.1 The User can create a new one (see “Create a new formatting rule” UC) 3 The User selects “Activate Rule” function 4 A confirmation on the success of the activation is provided
Expected results	Formatting rules set to active and submitted to the Active Formatting Rules repository
Variations	• User can modify/cancel this action before the activation
Issues	None
Test case Scope/Type	GUI/Whitebox

5.2.2.4 Removing an active formatting rule

TCId	TC5.2.2.4

Test case	Removing a formatting rule
Initial conditions	Set of complete rules defined and set as active
Configuration description	The user open the formatting rule editor to remove an active rule
Description of functionality to be tested	Active formatting rule removal
Partners, people involved	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Validator skill	Production editing
Data set used	AXDS-Formatting4
Steps	1 The User requests the list of Active Rules in the Active Rules Repository 2 The User selects the active rule to be disabled 3 The User selects “Remove Rule” function 4 The rule is Removed
Expected results	A selected compositional rule is removed from the Active Rules Repository
Variations	None
Issues	None
Test case Scope/Type	GUI/Whitebox

5.2.2.5 Debugging a formatting rule

TCId	TC5.2.2.5
Test case	Debugging a formatting rule
Initial conditions	Set of complete rules defined and set as active
Configuration description	The formatting rule editor is running and a rule is displayed on the screen.
Description of functionality to be tested	Debugging of rules
Partners, people involved	People involved with the end-user have the appropriate familiarity with the GUI and with script languages.
Validator skill	Production editing
Data set used	AXDS-Formatting4
Steps	1 The User load a rule 2 The User chooses the debugging rule mode 3 The Rule Editor enters in the Debugging Mode 4 During the debugging mode the User: 4.1 Check the statements of rule step by step 4.2 Control the values of current variables 5 Exit from the debugging mode
Expected results	The debugging has been successfully executed.
Variations	• The user has written a new rule and wants to debug it
Issues	Nome
Test case Scope/Type	GUI/Whitebox

6 AXMEDIS Workflow (WP4.3.3. IRC, WP5.5.5: CRS4, XIM)

6.1 Controlling and supervising local AXMEDIS tools (WP4.3.3. IRC, WP5.5.5: CRS4, XIM)

6.1.1 General Workflow Use Cases

6.1.1.1 Create NPD Workspace

TCId	TC6.1.1.1
Test case	Create NPD
Initial conditions	Always valid: user has been identified by System
Configuration description	WF editor plug in should be available Create NPD process is configured in WF User has the correct rights
Description of functionality to be tested	This use case when run should create a fresh NPD workspace folder with the required configuration files in it etc i.e. a suitable workspace desktop suited to the role of the participant(s) in the value chain segment to which they are contributing towards the NPD as a whole
Partners, people involved	This includes the user client initially as the new NPD owner/initiator However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Click on “Create NPD” button
Expected results	New NPD project(s) space created in the user client & P2P desktops New NPD creation process instance started
Variations	• User has no rights
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.2 Add

TCId	TC6.1.1.2
Test case	Add
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is “add” Adding component available and not protected
Description of functionality to be tested	This use case is responsible for adding components to the NPD. Typically it can be inherited to add projects, people, roles, processes, phases, partners, components, activities, Rights, DRM, etc
Partners, people involved	This includes the user client initially as the NPD owner who should permit the addition of components. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.

Data set used	AXDS-Workflow1
Steps	1 Click on "Add component" button.
Expected results	New component added to active NPD. Started (if any) a sub-process for managing the newly created object
Variations	• User has no rights • Component and AXMEDIS Object incompatibility
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.3 Edit

TCId	TC6.1.1.3
Test case	Edit Note: this is a test case with Workflow tight integration to editors (multiple interface)
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	NPD must exist. User has the correct rights One possible activity in the selected work item is "edit" Editing component available and not protected
Description of functionality to be tested	This use case is responsible for editing various aspects of the NPD. It can be used to edit the current DRM rules or can be used to edit a component based on the selected process and updates versions if required.
Partners, people involved	This includes the user client initially as the NPD owner who should permit the editing of NPD properties. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Click on "Edit" button
Expected results	Proper editor invoked for active NPD.
Variations	• User has no rights
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.4 Remove

TCId	TC6.1.1.4
Test case	Remove
Initial conditions	An non-empty NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the work item is "remove" Removing component available and not protected
Description of	This is a generic use case responsible for removing anything from the NPD. e.g.

functionality to be tested	partners, people, processes, components, etc.
Partners, people involved	This includes the user client initially as the NPD owner who should permit the addition of components. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Select component to remove then click on 'remove' 2 Optional confirmation dialogue
Expected results	Selected component deleted from active NPD.
Variations	• User has no rights
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.5 Group

TCId	TC6.1.1.5
Test case	Group
Initial conditions	Non-empty NPD must be active/open containing more than one component.
Configuration description	WF editor plug in should be available
Description of functionality to be tested	This use case is responsible for bundling components, people, processes, partners, projects, teams, packets, digital assets products, etc into one entity which may be further referred to.
Partners, people involved	This includes the user client initially as the NPD owner who should permit the grouping of components. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow2
Steps	1 Select multiple components (using shift-click or click and drag). 2 Click on "Group".
Expected results	New group created containing the selected objects.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.6 Show

TCId	TC6.1.1.6
Test case	Show
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is "show"
Description of functionality to be	This use case is responsible for showing information related to various components, their copyrights, DRM, History (metadata, timestamp, version),

tested	Template (house styles, business rules), global state of any projects, etc.
Partners, people involved	This includes the user client initially as the NPD owner who should permit the viewing of NPD information. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Click on “Show info”.
Expected results	Properties related to the active NPD displayed.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.7 Discard NPD

TCId	TC6.1.1.7
Test case	Discard NPD
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is “discard”
Description of functionality to be tested	This destroys the NPD workspace, when the decision of No-Go is taken. This removes all the information regarding the NPD.
Partners, people involved	This includes the user client initially as the NPD owner who can delete the NPD. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Click on “Discard NPD”. Confirmation dialogue.
Expected results	Active NPD deleted along with associated components. The process instance initiated with the NPD instance creation is aborted.
Variations	• No rights.
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.8 Search

TCId	TC6.1.1.8
Test case	Search
Initial conditions	An NPD must be active/open
Configuration description	The AXMEDIS Database Manger should be up and available along with Query Support Interface.
Description of functionality to be tested	This is a generic use case that can search for anything. A special case can be inherited to search for eligible components to be worked on.
Partners, people involved	Any WorkGroup members/other partners internal or external

Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Click on the “Search” button and launch a search for either a specific type of object for a particular step within a NPD or for any objects with certain attributes.
Expected results	The search request is communicated through either Query Support or AXOM, fishing for something of interest
Variations	None
Issues	None
Test cases Scope / Type	GUI/BlackBox

6.1.1.9 Track Component

TCId	TC6.1.1.9
Test case	Track component
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is “track component”
Description of functionality to be tested	This tracks down the history of the selected component. The result comprises of all the actions performed on the component along with all the future activities including “wait actions” re “suspended” objects awaiting pending operations which may themselves be contingent on Critical Path Action(s) (CPA) trigger(s).
Partners, people involved	This includes the user client initially as the NPD owner who should permit the tracking of components. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Select a component. 2 Click on “Track component” button.
Expected results	History and planned steps of selected component displayed.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.10 Track CPA

TCId	TC6.1.1.10
Test case	Track CPA
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is “track CPA”
Description of functionality to be	This use case identifies the Critical Path Activities (CPA) and produces all the information regarding those activities e.g. people involved, components being

tested	worked on, processes needing attention, etc.
Partners, people involved	This includes the user client initially as the NPD owner who should permit the addition of components. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow3
Steps	1 Click on "Track CPA" button.
Expected results	Displays the critical path activities for the active NPD.
Variations	None
Issues	None
Test cases Scope / Type	GUI/BlackBox

6.1.1.11 Timestamp Generator

TCId	TC6.1.1.11
Test case	Timestamp generator
Initial conditions	A non-empty NPD must be active/open
Configuration description	WF editor plug in should be available
Description of functionality to be tested	This use case is responsible for generating the timestamp for each of the activities that are performed on an object by an actor or process at anytime, anywhere any place by any partner – in any phase of the production and distribution end-to-end. This can be represented within the metadata and will be used by "Track Component" to locate the evolution status of any object within nested spiral development lifecycles across distributed teams from different units/partners. This will allow global tracking including accommodating re-entrant and re-cursive states of processing of the objects across partner project spaces (projects, phases, processes, persons, partners, places, periods, purpose, progress-to-date, project-work-remaining – 10P STAMP, Badii 2004)
Partners, people involved	This includes the user client initially as the NPD owner who should permit the time-stamping of activities. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	None
Steps	1 Log the beginning and end of any task performed on any object.
Expected results	Upon task closure, updated metadata with the timestamp information both duration from MIS as well as time for modification.
Variations	• This use case can be tested as expected result for each of the other cases.
Issues	AXDS-Workflow1
Test case Scope/Type	GUI / BlackBox

6.1.1.12 Generate Versions

TCId	TC6.1.1.12
Test case	Generate version
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item

	User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is "generate version"
Description of functionality to be tested	This generates hierarchical versions for all the digital and hard copy artefacts for the NPD development
Partners, people involved	This includes the user client initially as the NPD owner who should permit the generation of versions. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Select a component. 2 Click on "Generate version" button.
Expected results	New version of active component added to active NPD.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.13 List Work

TCId	TC6.1.1.13
Test case	List work
Initial conditions	User has been identified by System A non-empty NPD must be active/open
Configuration description	WF editor plug in should be available There are work items to which the user is assigned
Description of functionality to be tested	This use case is responsible for generating a hierarchical list of the sequence of all the work to be done in a particular sectorial workflow scenario, e.g. phases, processes to be invoked on certain objects by certain people with specific globally traceable coordinates as unique and easily retrievable instances (i.e. 10P Stamped Workflow Objects).
Partners, people involved	This includes the user client initially as the NPD owner who should permit listing of work. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow4
Steps	1 Select a person or workflow stage. 2 Click on "List work" button.
Expected results	It is to provide the historical evolution of tasks already performed in a project or all the tasks to be performed within a project. The task can be reference by the object, person or phase of the NPD.
Variations	• The user gets the work list for all of his activities not specific to any particular project or process. Instead a list of all the work to be done by the user is shown.
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.14 Select a work item

TCId	TC6.1.1.14
Test case	Select a work item
Initial conditions	User has been identified by System The user has executed the “personal work list” case or “list work” case.
Configuration description	There are work items to which the user is assigned
Description of functionality to be tested	This use case is responsible for selecting a work item from the work list
Partners, people involved	Any user of the Workflow
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow4
Steps	1 Click on “select work item” button.
Expected results	The user work item activity list and/or description is displayed
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.15 End Current Task

TCId	TC6.1.1.15
Test case	End Current Task
Initial conditions	User has been identified by System The user has selected a Work Item The user has performed the task of the selected work item
Configuration description	
Description of functionality to be tested	Users can invoke this functionality to signal to the workflow system their wish to have an activity terminated. Accordingly the workflow system will proceed to the next step in the workflow process instance (It is important to note that this functionality enables an over-ride control action on the part of the human operator if required)
Partners, people involved	Any user of the Workflow
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow4
Steps	1 Click on “complete work item” button.
Expected results	The work item goes to completed status, the Workflow engine passes to the next activity as planned in the process instance flow, the work item is deleted from the user’s work list
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox, WhiteBox, UnitTest

6.1.1.16 Distribute Work

TCId	TC6.1.1.16
Test case	Distribute work
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is “distribute work”
Description of functionality to be tested	This use case is responsible for distributing the work amongst the people assigned to the NPD. The work can be at component level or at NPD level. Some of the assigned work may be pipelined or suspended in a wait/pending stack, awaiting appropriate triggers for handover
Partners, people involved	This includes the user client initially as the NPD owner who should permit the distribution of work. However it should be possible to add names of WorkGroup members/other partners internal or external This is typical function for a team leader or supervisor. It can also used by a common user for “delegating” specifically a task.
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow2
Steps	1 Either select a specific component first (to distribute at component level), or directly click on “Distribute work” button to distribute at the NPD level.
Expected results	Work is (re)scheduled for the selected component or NPD.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.17 Change State/Phase

TCId	TC6.1.1.17
Test case	Change State/Phase
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is “change phase”
Description of functionality to be tested	This use case is responsible for changing states of objects/actors or phases of a project including triggering and the upload of a new workspace for a new phase in the project. e.g. the object may become available after copyright clearance or a person/partner may become (un)available.
Partners, people involved	This includes the user client initially as the NPD owner who should permit state changes. However it should be possible to add names of WorkGroup members/other partners internal or external. This could also be used for authorisation and rights clearance and management.
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow4
Steps	1 Select a component or actor

	2 Click “change state”.
Expected results	Either the change is with respect to project phases, in which case a phase change occurs including the upload of new profile and workspace environment to cater for the new phase or the attributes, especially e.g rights. State is changed for the selected component or actor.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.18 Notification

TCId	TC6.1.1.18
Test case	Notification
Initial conditions	An non-empty NPD must be active/open
Configuration description	WF editor plug in should be available
Description of functionality to be tested	This use case is responsible for sending out notifications to the responsible users for the start and/or end of the activities/work; e.g. request for information or components, etc.
Partners, people involved	This includes the user client initially as the NPD owner who should permit notifications. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Select one or more actors, select from list of message types 2 Click “notify”.
Expected results	Appropriate notification is sent to responsible actors via appropriate tool (e.g. email).
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.19 Global Viewer

TCId	TC6.1.1.19
Test case	Global viewer
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	WF editor plug in should be available User has the correct rights One possible activity in the selected work item is “global view”
Description of functionality to be tested	This use case is to collect all the information for the current NPD and present a global view for managerial decisions and for Production accounting information feed made accessible any Enterprise MIS platforms such as SAP (along with the 10P Object Stamps)
Partners, people involved	This includes the user client initially as the NPD owner who should permit global views. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.

Data set used	AXDS-Workflow4
Steps	1 Click “global view”.
Expected results	Global information is displayed/exported for the active NPD.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

6.1.1.20 UI

TCId	TC6.1.1.20
Test case	UI
Initial conditions	None
Configuration description	WF editor plug in should be available
Description of functionality to be tested	This is a generic use case for the user interface, which delivers all the functionalities provided by the user interface of the selected workflow tool. This use case is just a placeholder to for the User Interface requirement and hence there are no steps involved. The specific behaviour of this use-case will depend on the selected workflow tool.
Partners, people involved	This includes the user client initially as the NPD owner who should permit interaction. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	None
Steps	-
Expected results	A browser opens with all the above functionalities of AXWF with appropriate shortcuts.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

6.1.1.21 XUI

TCId	TC6.1.1.21
Test case	XUI
Initial conditions	None
Configuration description	WF editor plug in should be available
Description of functionality to be tested	This is a generic use case for inter-organisational workflow interaction and the interaction between AXWF and other AXMEDIS native tool. This use case is just the placeholder for the interfaces of AXWF and other AXMEDIS tools, which are described in DE3-1-2 framework and tool specification document.
Partners, people involved	This includes the user client initially as the NPD owner who should permit interaction. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	None
Steps	-

Expected results	-
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

6.1.1.22 Check-in

TCId	TC6.1.1.22
Test case	Check-in Note: this is a test case with Workflow loose integration to editors (simple interface)
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System
Configuration description	NPD must exist. User has the correct rights One possible activity in the selected work item is “check-in” Checking-in component available and not protected
Description of functionality to be tested	This use case is responsible for editing manually various aspects of the NPD. It can be used to edit the current DRM rules or can be used to edit a component based on the selected process and updates versions if required.
Partners, people involved	This includes the user client initially as the NPD owner who should permit the editing of NPD properties. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Click on “check-in” button.
Expected results	The object is copied from AXMEDIS DB To an area for exclusive access of the user, ready to be downloaded
Variations	• User has no rights
Issues	None
Test case Scope/Type	Back end / BlackBox

6.1.1.23 Check-out

TCId	TC6.1.1.23
Test case	Check-out Note: this is a test case with Workflow loose integration to editors (simple interface)
Initial conditions	An NPD process instance is active, a task was assigned to the user, the user selected a work item User has been identified by System User has previously checked-out.
Configuration description	NPD must exist. User has the correct rights One possible activity in the selected work item is “check-out”
Description of functionality to be tested	This use case is responsible for copying the object from the user exclusive access area (when he previously uploaded it) to the AXMEDIS DB

Partners, people involved	This includes the user client initially as the NPD owner who should permit the editing of NPD properties. However it should be possible to add names of WorkGroup members/other partners internal or external
Validator(s) skill	Common baseline skills as expected to be possessed by the NPD project members.
Data set used	AXDS-Workflow1
Steps	1 Click on “check-out” button.
Expected results	The file is copied in the AXMEDIS DB
Variations	• User has no rights • It can automatically execute the “task completed”
Issues	None
Test case Scope/Type	Back end / BlackBox

7 AXMEDIS Object Acquisition from CMS (DSI)

7.1 Automatic gathering of Content, Collector Engine (WP4.2.1: DSI with subcontract)

7.1.1 Defining and Collecting content from the CMS

TCId	TC7.1.1
Test case	Collecting content from the CMS.
Initial conditions	External CMS is accessible for Focuseek
Configuration description	Crawler Collector Indexer is up
Description of functionality to be tested	Content is retrieved from the CMS and metadata is stored in the Crawler Results Integrated Database
Partners, people involved	Publishers and Distributors
Validator skill	None
Data set used	AXDS-CMS
Steps	1 Crawler Collector Indexer is started to crawl the CMS giving information to connect to the CMS database and the kinds of information to be collected. 2 Queries are performed on the indexer to search for content that should be present
Expected results	The Crawler Results Integrated Database is populated with the CMS metadata and the data requested is found
Variations	• Content/metadata is updated in the CMS and after a while the updated content should be available • Different plug-ins to access to CMS database have to be tested
Issues	None
Test case Scope/Type	Backend, black box

7.1.2 Creating objects from the Collected Content

TCId	TC7.1.2
Test case	Creating objects from the Collected Content
Initial conditions	Content of the CMS Database has been fully crawled
Configuration description	Crawler Collector Indexer is up
Description of functionality to be tested	Object production from collected content
Partners, people involved	Publishers and Distributors
Validator skill	None
Data set used	AXDS-CMS
Steps	1 The Mapping information to map CMS metadata to AXMEDIS metadata is provided 2 Collector Engine User Interface is used to add an importing rule stating the kind of objects to be imported into the database 3 The rule is activated 4 The AXDB is browsed or queries are performed on AXDB to verify the presence of the object, its structure and the metadata acquired from CMS. AXMEDIS Editor could also be used to check the procuded AXMEDIS

	objects.
Expected results	The AXMEDIS Database is populated with objects produced by Collector Engine
Variations	• Content/metadata is updated in the CMS and after a while the updated AXMEDIS object should be available in the AXMEDIS Database (a different version should be present for the same object) • Different importing rules should be tested simultaneously and overlapping objects (affecting two or more importing rules) should be imported once.
Issues	None
Test case Scope/Type	Backend & GUI, black box

7.1.3 Calculating content descriptors/fingerprint (during crawling)

TCId	TC7.1.3
Test case	Calculating content descriptors/fingerprint (during crawling).
Initial conditions	External CMS is accessible for Focuseek and Focuseek is collecting content from external CMS.
Configuration description	Focuseek, AXMEDIS database, AXMEDIS OID-Generator
Description of functionality to be tested	Content is indexed automatically and fingerprints/content descriptors are calculated. Fingerprints and content descriptors are stored in the AXDB.
Partners, people involved	Publishers and Distributors (WorkGroup members/other partners internal or external)
Validator(s) skill	None
Data set used	AXDS-CMS
Steps	1 Collector Engine User Interface is used to add/modify an importing rule stating the fingerprinting tools to use 2 The rule is activated 3 The AXDB is browsed or queries are performed on AXDB to verify the presence of the object and the values of descriptors calculated by the fingerprinting tools.
Expected results	Fingerprints/content descriptors are calculated. Content is accessible through identifiers or descriptors and is found in queries.
Variations	• Different types of contents (audio, images, and video) • Different sets (e.g. different genres of audio) • Different algorithms (if available) • Content is already stored in the AXDB
Issues	The queries' result depends on the content descriptor type. Queries for similar content are therefore different from queries based on the identifiers.
Test case Scope/Type	Backend (Crawler Collector Indexer, Fingerprinting Technologies, AXDB) Blackbox

8 AXMEDIS Database (EXITECH)

8.1 Managing a Database of AXMEDIS Objects (EXITECH)

8.1.1 Administer Objects in the AXMEDIS DB:

TCId	TC8.1.1a
Test case	Administer Objects in the AXMEDIS DB (remove the last version of the object) [
Initial conditions	None
Configuration description	Before running the test case, in the database a new object with version 1.0 and version 1.1 is inserted. The ID of this object will be referred as OID-versioned in the following
Description of functionality to be tested	When the last revision of an object is removed the previous version still exists.
Partners, people involved	Content Integrator, Content Distributor, and in general all the user that have an AXMEDIS DB in-house
Validator(s) skill	None
Data set used	AXDS-DB1
Steps	1 The Actor through the DB admin interface asks, or a back end module call an API, to requests to remove the last version of the object with ID OID-versioned 2 The administrative DB interface verifies if another version of the object exists. If yes, remove the last version and return back the actual last version of the object 3 The Actor or the backend module tests if the returned version is 1.0
Expected results	The returned version of the object is 1.0
Variations	• The Actor through the DB admin interface asks, or a back end module call an API, to requests to remove again the last version of the object with ID OID-versioned • The system returns an error code to show that the last version of the object is present and therefore the delete object functionality has to be used, to correctly clear all references
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

TCId	TC8.1.1b
Test case	Administer Objects in the AXMEDIS DB (remove an object)
Initial conditions	None
Configuration description	Before running the test case, in the database a new object with version 1.0 and version 1.1 is inserted. The ID of this object will be referred as OID-tobedeleted in the following
Description of functionality to be tested	When an object is deleted, all the versions and reference to the object are removed.
Partners, people involved	Content Integrator, Content Distributor, and in general all the user that have an AXMEDIS DB in-house
Validator(s) skill	None
Data set used	AXDS-DB1
Steps	1 The Actor through the DB admin interface asks, or a back end module call an API, to requests to remove the object with ID OID-tobedeleted 2 The administrative DB interface clear all reference to the object and delete all the versions of the object

	3 The Actor query the system for having the object with ID OID-tobedeleted 4 The system return a NULL reference to show that the object do not exist
Expected results	After the deletion the object exists no more.
Variations	None
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

8.1.2 Administer User/Groups in the AXMEDIS DB

TCId	TC8.1.2a
Test case	Administer User in the AXMEDIS DB (add a user)
Initial conditions	None
Configuration description	None
Description of functionality to be tested	An user with predefined grants is created in the system
Partners, people involved	Content Integrator, Content Distributor, and in general all the user that have an AXMEDIS DB in-house
Validator(s) skill	None
Data set used	None
Steps	1 The Actor through the DB admin interface asks, or a back end module call an API, to requests to create an user with a predefined username, nick, password and grants 2 The administrative DB interface creates such user and returns back the User ID 3 The Actor query the system for having User-ID of the user with the username used in step 1. 4 The UID returned at step 3 is checked against the UID returned at step 2: they must be identical
Expected results	After the creation of the user, the user is present in the system.
Variations	• If the user is already present in the system, then at point 2, a NULL value is returned and the check at step 4 fails.
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

TCId	TC8.1.2b
Test case	Administer Users in the AXMEDIS DB (remove a user)
Initial conditions	An user with a predefined username, user-to-be-deleted in the following, is created in the system
Configuration description	None
Description of functionality to be tested	An user with a predefined username is removed from the system
Partners, people involved	Content Integrator, Content Distributor, and in general all the user that have an AXMEDIS DB in-house
Validator(s) skill	None
Data set used	None apart from that in Initial Conditions
Steps	1 The Actor through the DB admin interface asks, or a back end module call an API, to requests to remove an user with a predefined username, say user-to-be-deleted 2 The administrative DB interface confirms the removing of the user

	3 The Actor query the system for having User-ID of the user with the username used in step 1. 4 The returned value must be NULL
Expected results	After the deletion of a user, the user is no more present in the system
Variations	None
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

TCId	TC8.1.2c
Test case	Administer Users in the AXMEDIS DB (change a user)
Initial conditions	A user with a predefined username, user-to-be-changed in the following, is created in the system with a predefined set of property.
Configuration description	None
Description of functionality to be tested	An user with a predefined username is changed
Partners, people involved	Content Integrator, Content Distributor, and in general all the user that have an AXMEDIS DB in-house
Validator(s) skill	None
Data set used	None apart from that in Initial Conditions
Steps	1 The Actor through the DB admin interface asks, or a back end module call an API, to requests to change an user with a predefined username, say user-to-be-deleted and a new list of user property is submitted 2 The administrative DB interface confirms the changing of the user 3 The Actor query the system for having properties of User-ID of the user with the username used in step 1. 4 The list of property is returned 5 The list is checked against the new list at step 1
Expected results	After the changing of a user, the new properties have been applied to the user
Variations	None
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

TCId	TC8.1.2d
Test case	Administer groups in the AXMEDIS DB (add a group)
Initial conditions	None
Configuration description	None
Description of functionality to be tested	An group is created in the system
Partners, people involved	Content Integrator, Content Distributor, and in general all the user that have an AXMEDIS DB in-house
Validator(s) skill	None
Data set used	None
Steps	1 The Actor through the DB admin interface asks, or a back end module call an API, to requests to create a group with a predefined group name 2 The administrative DB interface creates such group and returns back the group ID 3 The Actor query the system for having the list of groups 4 The new group must exists

Expected results	After the creation of the group, the group is present in the system.
Variations	- If the group is already present in the system, then at point 2, a NULL value is returned and the check at step 4 fails.
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

TCId	TC8.1.2e
Test case	Administer Users in the AXMEDIS DB (remove a group)
Initial conditions	A group with a predefined username, group-to-be-deleted in the following, is deleted from the system
Configuration description	None
Description of functionality to be tested	An user with a predefined username is removed from the system
Partners, people involved	Content Integrator, Content Distributor, and in general all the user that have an AXMEDIS DB in-house
Validator(s) skill	None
Data set used	None apart from that in Initial Conditions
Steps	1 The Actor through the DB admin interface asks, or a back end module call an API, to requests to remove a group, say group-to-be-deleted 2 The administrative DB interface confirms the removing of the group 3 The Actor query the system for having the list of groups 4 The group is not present in the list
Expected results	After the deletion of a group, the group is no more present in the system
Variations	None
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

8.1.3 Accessing a specific version of an AXMEDIS object

TCId	TC8.1.3
Test case	Accessing a specific version of an AXMEDIS object
Initial conditions	None
Configuration description	Before running the test case, in the database a new object with version 1.0 and version 1.1 is inserted.
Description of functionality to be tested	If a version is asked, then the desired version is given back to the actor
Partners, people involved	Generic AXMEDIS user
Validator(s) skill	None
Data set used	AXDS-DB1
Steps	1 The Actor through the User DB Interface asks for version 1.1 of the object in the Configuration 2 The DB will return an object 3 The Actor check if the requested version is correct, by checking the version submitted in the configuration description and that obtained back from DB
Expected results	The test is Ok if the two objects match.
Variations	- If the requested version do not exists (i.e. version 1.2 or the configuration) a NULL reference value is returned
Issues	None
Test case Scope/Type	Backend and BlackBox, possibly also a UnitTest

8.1.4 Removing last version of an AXMEDIS object

This feature is tested at admin level only. See 8.1.1.

8.1.5 Removing an AXMEDIS object

This feature is tested at admin level only. See 8.1.1.

8.1.6 User Management

This feature is tested at admin level only. See 8.1.2.

8.1.7 User Groups Management

This feature is tested at admin level only. See 8.1.2.

8.2 Making queries inside Databases of AXMEDIS objects and inside the objects (EXITECH)

8.2.1 Querying for AXMEDIS objects and inside objects

TCId	TC8.2.1
Test case	Querying for AXMEDIS objects and inside objects
Initial conditions	AXMEDIS system is filled with a predefined set of objects. A query for which the result set is known is created.
Configuration description	AXMEDIS Database, AXMEDIS Query Support, AXEPTool and other tools if needed by the query parameters on external sources
Description of functionality to be tested	Query support returns the correct result-set for a test query
Partners, people involved	Publishers, Distributors, end user
Validator(s) skill	None
Data set used	AXDS-DB2
Steps	1 Actor submits the test query 2 The system returns results 3 The results shown in the query support are checked against the list of expected results
Expected results	The Query Support correctly issue and collect query results
Variations	• A query with an empty result set is created and submitted o The Query Interface has to return an empty result set • A query with only 1 satisfying object is created and submitted o The Query Interface has to return a result set with the known result
Issues	None
Test case Scope/Type	Backend / BlackBox

8.2.2 Bookmark a query

TCId	TC8.2.2
Test case	Bookmark a query
Initial conditions	None
Configuration description	AXMEDIS Query Support user interface
Description of functionality to be tested	The Query Interface is capable of bookmarking a query
Partners, people involved	Publishers, Distributors, end user
Validator(s) skill	None

Data set used	None
Steps	1 Actor submits a query 2 Actor bookmark the query 3 The query is listed in the Actor bookmark
Expected results	The Query Interface correctly bookmarks a query
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

8.2.3 Retrieve a bookmarked query

TCId	TC8.2.3
Test case	Retrieve a bookmarked query
Initial conditions	None
Configuration description	AXMEDIS Query Support user interface
Description of functionality to be tested	The Query Interface is capable of retrieving a previously bookmarked query
Partners, people involved	Publishers, Distributors, end user
Validator(s) skill	None
Data set used	None
Steps	1 Actor submits a query 2 Actor bookmark the issued query 3 Actor select the just issued query among those present in the bookmark 4 Actor verify that the query is the same he/she has bookmarked
Expected results	The Query Interface correctly retrieves a bookmarked query
Variations	None
Issues	None
Test case Scope/Type	BlackBox

8.2.4 Organize bookmarked queries

TCId	TC8.2.4
Test case	Organize bookmarked query
Initial conditions	None
Configuration description	AXMEDIS Query Support user interface
Description of functionality to be tested	The Query Interface is capable of organising bookmarks
Partners, people involved	Publishers, Distributors, end user
Validator(s) skill	None
Data set used	None
Steps	1 The Actor, using the AXMEDIS Query Interface, recalls from his/her profile the bookmarks 2 The Actor creates a folder, renames a folder, deletes a folder, inserts query in a folder and removes queries from folders 3 The Actor confirms the new configuration of the bookmarks 4 The Actor verifies that his/her user profile is organized according to the modification issued and saved.

Expected results	The Query Interface correctly organises bookmarked queries
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

8.2.5 Save an incomplete query

TCId	TC8.2.5
Test case	Save an incomplete query
Initial conditions	None
Configuration description	AXMEDIS Query Support user interface
Description of functionality to be tested	The Query Interface is capable of storing an incomplete query
Partners, people involved	Publishers, Distributors, end user
Validator(s) skill	None
Data set used	None
Steps	1 The Actor, using the AXMEDIS Query Interface, during the composition of a query asks to store the query inside the local query registry 2 The Actor retrieve the query 3 The Actor verifies that the stored query is really the query he/she has stored.
Expected results	The Query Support notifies correctly stores incomplete queries
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

8.2.6 Retrieve an incomplete query

TCId	TC8.2.6
Test case	Retrieve an incomplete query
Initial conditions	None
Configuration description	AXMEDIS Query Support user interface
Description of functionality to be tested	The Query Interface is capable of storing an incomplete query
Partners, people involved	Publishers, Distributors, end user
Validator(s) skill	None
Data set used	None
Steps	1 The Actor, using the AXMEDIS Query Interface, during the composition of a query asks to store the query inside the local query registry 2 The Actor retrieves the query 3 The Actor verifies that the stored query is really the query he/she has stored.
Expected results	The Query Interface notifies that it correctly retrieves incomplete queries
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

9 AXMEDIS AXEPTools for P2P distribution on B2B (CRS4, UNIVLEEDS)

9.1 AXEPTool for P2P on B2B (WP4.4.1: CRS4, WP5.5: CRS4, WP5.5.1: CRS4)

9.1.1 Discovery and connection of peers on B2B P2P network

TCId	TC9.1.1
Test case	Discovery and connection of peers on B2B P2P network
Initial conditions	The AXEPTool is running on the user machine.
Configuration description	The AXEPTool, the database of certificates.
Description of functionality to be tested	The discovery process and the connection process.
Partners, people involved	The AXEPTool user, the supervisor authority.
Validator(s) skill	Expertise with GUI and Internet applications
Data set used	AXDS-Supervisor1, AXDS-P2Pheaders
Steps	1 The user clicks on the ”Connect” button or select an equivalent item in a menu of AXEPTool GUI 2 The successful connection status or an error message, is notified in the status bar (or on a pop-up window) of AXEPTool GUI
Expected results	The local host is allowed to exchange messages in the P2P network
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

9.1.2 Manage Downloads/Uploads (Report P2P downloads/uploads network traffic)

TCId	TC9.1.2
Test case	Manage Downloads/Uploads in the AXEPTool
Initial conditions	The AXEPTool is opened. One or more downloads are running
Configuration description	AXEPTool, one or more other peers are delivering AXMEDIS objects to the network
Description of functionality to be tested	The upload/download are managed by sessions that can be suspended/resumed/terminated by the user.
Partners, people involved	The AXEPTool user.
Validator(s) skill	Expertise with GUI and Internet applications.
Data set used	AXDS-DB2
Steps	1 The user opens the “Download Table” 2 The user selects one row where the value “status” is equal to “running” 3 The user activates a button, or selects an equivalent item in a menu, in order to “suspend”, to “resume” or to “terminate” a download session
Expected results	The selected session is suspended, resumed or terminated
Variations	• All steps are also applicable to an upload session
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2 Publication and loading AXMEDIS Objects of AXEPTool (WP4.4.5: CRS4, WP5.5.3: CRS4, WP4.4.4: CRS4, WP5.5.2: CRS4)

9.2.1 Creation of a publishing rule for the AXEPTool

TCId	TC9.2.1
Test case	Creation of a publishing rule for the AXEPTool
Initial conditions	One or more objects are stored in the AXMEDIS Data Base. The AXEPTool is running on the user machine, the Publication Engine is running and the Publication/Loading Rules/Selections User Interface is open.
Configuration description	AXMEDIS Data Base, AXEPTool, Publication Tool Engine (and Publication/Loading Rules/Selections Interface), AXEPTool OUT AXMEDIS Data Base.
Description of functionality to be tested	The creation of a rule for the publication of AXMEDIS objects into the Output Database of the AXEPTool.
Partners, people involved	The AXEPTool user, content owner
Validator(s) skill	Expertise with GUI, knowledge of the data set used.
Data set used	AXDS-Kiosk3, AXDS-PMS1, AXDS-PMS2, AXDS-DRMSupport7
Steps	1 The user fills the data required to build a new publication rule
Expected results	A new rule is saved in the AXEPTool Active Publication Rules/Selections
Variations	• Creation of a new rule by example: o The user manually selects an AXMEDIS object in the AXMEDIS Data Base, or select them as a result of a query and thus from a Selection. o The user invokes the function “Build rule by example”
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.2 Automatic publication of a selection of objects on the AXEPTool

TCId	TC9.2.2
Test case	Automatic publication of a selection of objects on the AXEPTool
Initial conditions	One or more objects are stored in the AXMEDIS Data Base. The AXEPTool is running on the user machine and the Publication Engine User Interface is opened.
Configuration description	AXMEDIS Data Base, AXEPTool, Publication Tool Engine (and Publication Engine User Interface), AXEPTool OUT AXMEDIS Data Base.
Description of functionality to be tested	The automatic publication of AXMEDIS objects into the Output Database of the AXEPTool.
Partners, people involved	The AXEPTool user
Validator(s) skill	Expertise with GUI and Internet applications.
Data set used	AXDS-Kiosk1, AXDS-AXEPPR
Steps	1 The user opens the “AXMEDIS Data Base Browser” 2 The GUI is presented to the user that selects one or more AXMEDIS objects stored in the AXMEDIS Data Base 3 The user clicks on the “Make this Selection Active” button or select an equivalent item in a menu of the User Interface 4 The user select the rules to apply to the selection
Expected results	A new rule is saved in the AXEPTool Active Publication Rules/Selections
Variations	• Creation of a new rule by example: o The user manually selects an AXMEDIS object in the AXMEDIS

	Data Base, or select them as a result of a query and thus from a Selection. o The user invokes the function “Build rule by example”
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.3 Automatic updating of a modified object on the AXEPTool

TCId	TC9.2.3
Test case	Automatic updating of a modified object on the AXEPTool
Initial conditions	One or more objects are stored in the AXMEDIS Data Base. The AXEPTool and the test environment are running on the user machine, the Publication Engine User Interface is opened, the AXOB belongs to an active Selections.
Configuration description	AXMEDIS Data Base, AXEPTool, Publication Tool Engine (and Publication Engine User Interface), AXEPTool OUT AXMEDIS Data Base, test environment.
Description of functionality to be tested	The automatic updating of a modified AXMEDIS objects into the Output Database of the AXEPTool.
Partners, people involved	The AXEPTool user
Validator(s) skill	Expertise with GUI and Internet applications.
Data set used	AXDS-DB1, AXDS-AXEPPR, AXDS-AXEPAS
Steps	1 The user modify the AXOB. 2 The user clicks on the “Update” button or select an equivalent item in a menu of the User Interface
Expected results	The test environment receives the notification of the change.
Variations	• Another AXEPTool running on a different peer receives the notification of the change.
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.4 Automatic publication of a non protected object on the AXEPTool

TCId	TC9.2.4
Test case	Automatic publication of a non protected object on the AXEPTool
Initial conditions	The unprotected objects is stored in the AXDB. The AXEPTool and the test environment are running on the user machine, the Publication Engine User Interface is opened.
Configuration description	AXMEDIS Data Base, AXEPTool, Publication Tool Engine (and Publication Engine User Interface), Protection Tool Engine, AXEPTool OUT AXMEDIS Data Base.
Description of functionality to be tested	The automatic protection of an AXMEDIS objects into the Output Database of the AXEPTool.
Partners, people involved	The AXEPTool user
Validator(s) skill	Expertise with GUI and Internet applications.
Data set used	AXDS-PMS1, AXDS-PMS2, AXDS-DRMSupport7
Steps	1 The user adds the unprotected AXOB to an active selection.
Expected results	The Publication Tool Engine invokes the Protection Tool Engine. The Protection Tools Engine gets the content from the AXDB and creates an AXOB with the proper protection part. The protected AXOB is stored in the AXEPTool OUT AXDB.

Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.5 Manual publication of AXMEDIS Objects with the AXEPTool

TCId	TC9.2.5
Test case	Manual Publication of AXMEDIS Objects with the AXEPTool
Initial conditions	One or more objects are stored in the AXMEDIS Data Base. The AXEPTool is running on the user machine and the Publication Engine User Interface is opened.
Configuration description	AXMEDIS Data Base, AXEPTool, Publication Tool Engine (and Publication Engine User Interface), AXEPTool OUT AXMEDIS Data Base.
Description of functionality to be tested	The manual publication of AXMEDIS objects into the Output Database of the AXEPTool.
Partners, people involved	The AXEPTool user
Validator(s) skill	Expertise with GUI and Internet applications.
Data set used	AXDS-PMS1, AXDS-PMS2, AXDS-DRMSupport
Steps	1 The user opens the "AXMEDIS Data Base Browser" 2 The GUI is presented to the user that selects one or more AXMEDIS objects stored in the AXMEDIS Data Base 3 The user clicks on the "Publish" button or select an equivalent item in a menu of the User Interface
Expected results	The selected objects are copied in the Output Database of the AXEPTool
Variations	• If one or more selected objects are not valid or complete , the users are requested to check them. • If a selected object is not protected and the user wants it, the Publication Tool Engine invokes the Protection Tool Engine to protect the object.
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.6 Producing a query to search on the AXEPTool

TCId	TC9.2.6
Test case	Producing a query to search on the AXEPTool.
Initial conditions	The AXEPTool is running and connected on the B2B P2P network.
Configuration description	The AXEPTool, specifically the query section of the GUI. Connection on the B2B P2P network.
Description of functionality to be tested	To verify that simply filling-in the query field the user can launch the query. The user should be able to make easily a query thanks the suitable user interface made available by AXEPTool.
Partners, people involved	The AXEPTool user.
Validator(s) skill	People involved with the end-user have the appropriate familiarity with the GUI.
Data set used	AXDS-AXEPQH
Steps	1 The user press the button "New Query 2 A "New Query Dialog" is presented to the user that fills-in the fields 3 The user press the "OK" button
Expected results	A new "Query Result Table" is added in the Queries panel
Variations	• The user launches multiple query sections simply opening several "New Query Dialog" instances.
Issues	None

Test case Scope/Type	GUI / BlackBox

9.2.7 View/Manage query results coming from the AXEPTool

TCId	TC9.2.7
Test case	View/Manage Query Results coming from the AXEPTool
Configuration description	AXEPTool opened and connected to the P2P network. One or more queries have been produced and sent.
Description of functionality to be tested	The user can manage query results for a given query
Partners, people involved	List of people involved in the test, partners, user-groups, other people needed
Validator(s) skill	Accustomed with graphical user interfaces
Data set used	AXDS-AXEPQH
Steps	1 The user selects the panel “Queries” in the AXEPTool UI. 2 A tabbed pane (one tab for each query) is presented to the user that selects one among the queries tab. 3 The user makes a selection of query-hits from the “Query Result Table”. 4 The user right-clicks on the selection. 5 The user invokes a function in the pop-up.
Expected results	Depending on the menu item selected, a function is invoked (for instance, selected results are ‘deleted’).
Variations	None
Issues	None
Test case Scope/Type	GUI/BlackBox

9.2.8 Active query pool management for the AXEPTool

TCId	TC9.2.8
Test case	Active Query Management for the AXEPTool
Configuration description	• AXEPTool opened and connected to the P2P network. • One or more queries have been produced and sent. • Another, and only ONE, AXEPTool called “HOST-2” is connected to the same network and is able to process incoming queries
Description of functionality to be tested	The user can make a query active. An Active Query reposts itself to the network and keeps the AXEPTool up-to-date with respect to a given search.
Partners, people involved	List of people involved in the test, partners, user-groups, other people needed
Validator(s) skill	Accustomed with graphical user interfaces. Expert in networks
Data set used	AXDS-AXEPQH
Steps	1 The user selects the panel “Queries” in the AXEPTool UI. 2 The user selects one among the queries in the tabbed pane. 3 The user clicks on the button labelled “Make this query active”. 4 The user fills, in “Query Activation Dialog”, the interval time equal to 5min, the expire time equal to 15min, and click the button ‘OK’. 5 The user opens the “Network Traffic Panel” GUI in the HOST-2.
Expected results	In the “Network Traffic Panel” GUI in the HOST-2 the user must see three identical incoming queries after 5min, 10min, 15min. No further queries are received in HOST-2.
Variations	None.
Issues	None.

Test case Scope/Type	GUI, P2P Network / Blackbox.

9.2.9 Downloading an AXMEDIS object

TCId	TC9.2.9
Test case	Download an AXMEDIS Object
Initial conditions	One or more objects are available to download in the P2P network
Configuration description	The AXEPTool connected to P2P Network
Description of functionality to be tested	The user wants to start a download of an AXMEDIS Object from P2P network. The test is designed to verify that the download starts and the object is saved into the AXEPTool in AXMEDIS Database. A feedback on the download status must be shown.
Partners, people involved	Developers, Integrators
Validator(s) skill	A basic understanding of the architecture should be useful
Data set used	AXDS-Kiosk2
Steps	1 The user selects the panel “Queries” in the AXEPTool UI 2 The user selects one among the queries in the tabbed pane 3 The user selects object X in the “Query Result Table” 4 The user press the button “Start Download” 5 The user opens the “Download Table” 6 The user waits for the object “X”, that the value “percentage” is equal to “100%”
Expected results	The object “X” is stored in the AXEPTool in AXMEDIS Database.
Variations	None
Issues	None
Test case Scope/Type	GUI, P2P Network / BlackBox

9.2.10 Automatic downloading of a selection of objects available in the P2P network

TCId	TC9.2.10
Test case	Automatic downloading of a selection of objects available in the P2P network
Initial conditions	A suitable Selection of one or more AXOB is available in the AXEPTool network. The AXEPTool connected to P2P Network.
Configuration description	Two AXEPTool running on different peer (or test environment on the same peer), AXEPTool P2P Active Selection Engine.
Description of functionality to be tested	The automatic downloading of AXMEDIS objects into AXEPTool IN AXDB
Partners, people involved	Developers, Testers
Validator(s) skill	Good knowledge of AXEPTool architecture
Data set used	AXDS-DB2, AXDS-AXEPLR
Steps	1 The User activates the Selection by using the Publication/Loading rules/Selections Editor running in a different peer.
Expected results	AXEPTool P2P Active Selection Engine elaborates the active Selections contained in the P2P Active Selections. AXEPTool P2P Active Selection Engine downloads each AXOB of the Selection. The object is stored in the AXEPTool IN AXDB.
Variations	• The test environment simulate a peer, the user activates the Selection through the shell of it.

Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.11 Refining the selection (Active Selections) for the AXEPTool

TCId	TC9.2.11
Test case	Refining the selection (Active Selections) for the AXEPTool
Initial conditions	The AXEPTool is running on the user machine. A Selection of one or more AXOB AXOB have just been loaded and tried by the user AXEPTool network according to the related DRM rules.
Configuration description	AXEPTool, and AXEPTool P2P Active Selection Engine.
Description of functionality to be tested	The refining of a selection (Active Selections) for the AXEPTool
Partners, people involved	Developers, Testers
Validator(s) skill	Good knowledge of AXEPTool architecture
Data set used	AXDS-DB2, AXDS-AXEPLR
Steps	1 By the Publication/Loading rules/Selections The user selects only the AXOB he/she is interested in.
Expected results	The Selection become Active by submitting it to the AXEPTool Active Loading Rules/Selections The Loading Tool Engine of AXEPTool elaborates the Active Selection. Each object of the selection is loaded into the local AXDB.
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.12 Automatic loading new versions of AXMEDIS Objects for the AXEPTool

TCId	TC9.2.12
Test case	Automatic loading new versions of AXMEDIS Objects with the AXEPTool
Initial conditions	The AXEPTool is opened. AXEPTool IN AXMEDIS Data Base and AXMEDIS Data Base are both available. The new version of AXMEDIS object is included in the Active Selection that has already been performed.
Configuration description	Two AXEPTool running on different peer (or test environment on the same peer), AXEPTool P2P Active Selection Engine.
Description of functionality to be tested	The capability of the AXEPTool P2P Active Selection Engine downloads new versions of AXMEDIS objects.
Partners, people involved	Developers, Testers
Validator(s) skill	Good knowledge of AXEPTool architecture
Data set used	AXDS-DB2, AXDS-AXEPLR
Steps	1 The user publish a new version of AXMEDIS object with the Publication Tool Engine User Interface
Expected results	Publication and Monitoring Objects is informed of the new publication. AXEPTool P2P Active Selection Engine is alerted by Publication and Monitoring Objects. AXEPTool P2P Active Selection Engine verifies if the new published objects matches certain features in the Active Selections.

	Selected objects are loaded in the AXINDB or if its eligible as a « loadable » object it is loaded in the AXDB.
Variations	• The Publication and Monitoring Objects is informed of the publication of a new version by the test environment that simulate a peer.
Issues	None
Test case Scope/Type	Backend / BlackBox

9.2.13 Automatic loading new AXMEDIS Objects with the AXEPTool

TCId	TC9.2.13
Test case	Automatic loading new AXMEDIS Objects with the AXEPTool
Initial conditions	The AXEPTool is opened. AXEPTool IN AXMEDIS Data Base and AXMEDIS Data Base are both available. One or more Active Selections have already been performed.
Configuration description	Two AXEPTool running on different peer (or test environment on the same peer), AXEPTool P2P Active Selection Engine.
Description of functionality to be tested	The capability of the AXEPTool P2P Active Selection Engine to move new AXMEDIS objects to the AXMEDIS Database.
Partners, people involved	Developers, Testers
Validator(s) skill	Good knowledge of AXEPTool architecture
Data set used	AXDS-DB2, AXDS-AXEPLR
Steps	1 The user clicks on the "Publish" button or select an equivalent item in a menu of the Publication Tool Engine User Interface
Expected results	Publication and Monitoring Objects is informed of the new publication. AXEPTool P2P Active Selection Engine is alerted by Publication and Monitoring Objects. AXEPTool P2P Active Selection Engine verifies if the new published objects matches certain features in the Active Selections. Selected objects are loaded in the AXMEDIS Data Base.
Variations	• The Publication and Monitoring Objects is informed of the new publication by the test environment that simulate a peer.
Issues	None
Test case Scope/Type	Backend / BlackBox

9.2.14 Manual loading of AXMEDIS Objects with AXEPTool

TCId	TC9.2.14
Test case	Manual loading of AXMEDIS Objects with AXEPTool.
Initial conditions	The AXEPTool is opened. AXEPTool IN AXMEDIS Data Base and AXMEDIS Data Base are both available.
Configuration description	AXEPTool, AXEPTool IN AXMEDIS Data Base and AXMEDIS Data Base.
Description of functionality to be tested	The user can select one ore more AXMEDIS objects from the AXEPTool in AXMEDIS Database. The AXEPTool is able to move objects to the AXMEDIS Database.
Partners, people involved	Content Integrator, Content Distributors.
Validator(s) skill	Basic understanding of AXEPTool.
Data set used	AXDS-DB2
Steps	1 The user opens the "AXEPTool in AXMEDIS Database Browser"

	2 The user selects one or more AXMEDIS objects in the AXEPTool in AXMEDIS Database 3 The user press the "LOAD" button
Expected results	Selected objects are loaded in the AXMEDIS Data Base.
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

9.2.15 Creation of a loading rule for the AXEPTool

TCId	TC9.2.15
Test case	Creation of a loading rule for the AXEPTool
Initial conditions	One or more objects are stored in the AXEPTool IN AXMEDIS Data Base. The AXEPTool is running on the user machine, the Loading Engine is running and the Publication/Loading Rules/Selections User Interface is open.
Configuration description	AXEPTool IN AXMEDIS Data Base, AXEPTool, Loading Tool Engine (and Publication/Loading Rules/Selections Interface), AXEPTool OUT AXMEDIS Data Base.
Description of functionality to be tested	The creation of a rule for the loading of AXMEDIS objects into the AXEPTool IN AXMEDIS Data Base
Partners, people involved	The AXEPTool user, content owner
Validator(s) skill	Expertise with GUI, knowledge of the data set used.
Data set used	AXDS-Kiosk3, AXDS-PMS1, AXDS-PMS2, AXDS-DRMSupport7
Steps	1 The user fills the data required to build a new loading rule
Expected results	A new rule is saved in the Loading Rules repository
Variations	• Creation of a new rule by example: o The user manually selects an AXMEDIS object in the AXEPTool IN AXMEDIS Data Base. o The user invokes the function "Build rule by example"
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

9.2.16 Preview an AXMEDIS object content coming from AXEPTool

TCId	TC9.2.16
Test case	Preview an AXMEDIS object content coming from AXEPTool
Initial conditions	One or more AXMEDIS Objects are in the AXEPTool IN AXMEDIS Data Base.
Configuration description	AXEPTool. Suitable media players/viewers are installed in the user system and AXEPTool is aware of that.
Description of functionality to be tested	The test is designed to test the AXEPTool feature of previewing an object. The AXEPTool must be capable to preview an object using a suitable media player installed in the user system or an error message should be prompted if not possible.
Partners, people involved	Developers, Testers
Validator(s) skill	Expertise with GUI
Data set used	AXDS-Kiosk1
Steps	1 The user selects an Object to preview 2 The user selects the preview button or selects an equivalent item in a menu
Expected results	A suitable media player is opened and preview starts or an error message is shown.

Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

9.2.17 Feedback toward the workflow system

TCId	TC9.1.17
Test case	Feedback toward the workflow system
Initial conditions	The AXEPTool and the workflow systems are opened and connected to the test environment. One or more AXMEDIS Objects are available for publishing or downloading.
Configuration description	The AXEPTool and workflow systems connected to test environment.
Description of functionality to be tested	The test is designed to test the AXEPTool feature to execute a requests of publishing and downloading, from the workflow systems.
Partners, people involved	Developers, Testers
Validator(s) skill	Good knowledge of AXEPTool architecture
Data set used	None
Steps	1 Through the shell of the test environment the user launch the publishing or downloading script.
Expected results	The AXEPTool executes the request. The AXEPTool informs the workflow system about the outcome of the operation.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

10 Programme and Publication Engine Tools (WP5.4.5: UNIVLEEDS, WP4.2.6: FHGIGD)

10.1 Programme and publication rules production

TCId	TC10.1
Test case	The actor, typically a Programme Producer or Manager requests to create/define/edit a programme for certain channel
Initial conditions	1. Active Publication Rules and Data Model support is accessible 2. Query support can provide list of objects available for distribution or formatting and distribution
Configuration description	The programme manager wants to set a list of rules for scheduling and programme production
Description of functionality to be tested	New programme is created correctly.
Partners, people involved	List of people involved in the test, partners, user-groups expert related to programme production
Validator(s) skill	Programme production and partners involved with data queries
Data set used	AXDS-P&P1
Steps	1 Programme producer uses GUI to submit query for objects and is returned a list of results 2 Programme producer to select part/all/none of results using the GUI 3 Programme producer schedules as a new rule or updates an existing rule 4 User specify the distribution channel of this programme 5 Schedule is saved as “rule”
Expected results	Defined programme rules for distribution defaulted as “inactive” and saved in repository
Variations	• Defining different channels for distribution • Repeat for certain programmes e.g. daily, weekly etc.
Issues	None
Test case Scope/Type	GUI/BlackBox

10.2 Programme and publication rules editing

TCId	TC10.2
Test case	Editing and collection of Programme and Production Rules
Initial conditions	Set of rules available for editing and collecting
Configuration description	The programme manager wants to change the rules for the programme production
Description of functionality to be tested	The editing and collection of rules
Partners, people involved	List of people involved in the test, partners, user-groups expert related to programme production
Validator(s) skill	Programme editing and partners involved with data queries
Data set used	AXDS-P&P1, AXDS-P&P2
Steps	1. The user browses the existing rules/programme in the collection 2. The user selects and loads certain rules/programme 3. Use “Programme Production” UC as discussed above to edit the rules
Expected results	New rules are collected and saved in the collection of rules
Variations	• Change distribution channel, schedule times, etc.

Issues	None
Test case Scope/Type	GUI/BlackBox

10.3 Activation of program and publication rules

TCId	TC10.3
Test case	Actor decides to "activate", i.e. to publish, the programme
Initial conditions	Set of complete rules defined and set as inactive (by default)
Configuration description	The programme manager has completed the programme rules editing and wants to set the programme to be published (as active)
Description of functionality to be tested	Loading a programme with a complete set of rules to be set as active/publish; request changes if programme is incomplete; and confirmation on the success of the publication
Partners, people involved	List of people involved in the test, partners, user-groups expert related to programme production
Validator(s) skill	Programme editing
Data set used	AXDS-P&P2
Steps	1 If the programme has not been loaded, the user can select and load the programme, for final checking 2 A GUI to allow the user to activate/publish the programme 3 The component check the status and required information of the programme and ask for more input if the programme is incomplete (e.g. unknown publication date or channel) 4 A confirmation on the success of the publication
Expected results	Programme with rules set to active and submitted to the Active Publication Rules
Variations	• User can modify/cancel this action before the schedule distribution
Issues	None
Test case Scope/Type	GUI/BlackBox

10.4 Launch of program and publication rules from workflow

TCId	TC10.4
Test case	The active engine monitoring the system clock to deliver on time one or more schedules and published programmes for actual consumption
Initial conditions	Active engine monitoring the system clock; Distribution channel profile including bandwidth with an estimated time for the actual delivery and time required for formatting (if on demand is needed) Listen to communication channel from workflow in order to activate on-demand programme and publication with a given programme.
Configuration description	Engine is active and accessing correct system time.
Description of functionality to be tested	Delivery on time of schedules and published programmes
Partners, people involved	Partners involved with distribution, object formatting and database management
Validator(s) skill	1. Database management 2. Object Formatting 3. Distribution
Data set used	AXDS-P&P2
Steps	1 Check for new activated rules 2 If new rules activated check source and target format

	3 Compare profiles of the object and profile of the channel 4 If profiles mismatch, request formatting using the Workflow Manager and receive reply of success or failure 5 Check distribution time from Distribution and Client Profile 6 Set start time for on-time distribution 7 Check time and a list of activated programme for publication 8 Deliver a programme to the distribution server at the correct time
Expected results	Distribute correct object to a correct channel before the object consumption schedule as defined in the activated rule.
Variations	• With and without formatting (on demand) requirement • Different channel with different bandwidth
Issues	Require setting and profiles for different distribution servers and the formatting engine (time requirements)
Test case Scope/Type	Backend, Blackbox

10.5 Trial pre-activation of Programme and Publication rules

TCId	TC10.5
Test case	Actor decides to publish (“quick trial”) or (“full trial”) the programme
Initial conditions	Set of complete rules defined and set as inactive
Configuration description	The programme manager has completed the programme rules editing and wants to set the programme to be trial.
Description of functionality to be tested	Loading a programme with a complete set of rules to be set as quick trial or full trial; request changes if programme is incomplete; and confirmation on the success of the trial publication.
Partners, people involved	List of people involved in the test, partners, user-groups expert related to programme production
Validator(s) skill	Programme editing
Data set used	AXDS-P&P2
Steps	1 If the programme has not been loaded, the user can select and load the programme, for final checking 2 A GUI to allow the user to activate/publish the trial programme as a quick or full trial 3 The component check the status and required information of the programme and ask for more input if the programme is incomplete (e.g. unknown publication date or channel) 4 A confirmation on the success of the publication trial
Expected results	Programme with rules set to quick/full trial and submitted to the Active Publication Rules and flagged as successful or failed when completed
Variations	• User can modify/cancel this action before the schedule distribution
Issues	None
Test case Scope/Type	GUI/Whitebox

10.6 Launch of trial program and publication rules from workflow

TCId	TC10.6
Test case	The active engine monitoring the system clock to deliver on time one or more schedules and published programmes for actual consumption
Initial conditions	Active engine monitoring the system clock; Distribution channel profile including bandwidth with an estimated time for the actual delivery and time required for formatting (if on demand is needed) Listen to communication channel from workflow in order to activate on-demand programme and publication.
Configuration	Engine is active and accessing correct system time.

description	
Description of functionality to be tested	Flagging of successful or failed trial runs of published programmes
Partners, people involved	Partners involved with distribution, object formatting and database management
Validator(s) skill	1. Database management 2. Object Formatting 3. Distribution
Data set used	AXDS-P&P2
Steps	1 Check for new activated rules flagged as a trial 2 If new rules activated check source and target format 3 Compare profiles of the object and profile of the channel 4 If profiles mismatch, request formatting using the Workflow Manager and receive reply of success or failure if full trial or receive reply if formatting is possible 5 Check distribution time from Distribution and Client Profile 6 Set start time for on-time distribution 7 Process immediately the object requested (if full trial) and request reply from distribution server if distribution of object type is possible
Expected results	Distribute correct object to a correct channel before the object consumption schedule as defined in the activated rule.
Variations	• Full trial request formatting and quick trial request reply if formatting is possible • Request possible distribution server replies for different channel with different bandwidth
Issues	None
Test case Scope/Type	Backend, Blackbox

10.7 Client user selects media from “push” list of promoted content (FHGIGD, Comverse, Eutelsat, Tiscali, ILABS)

TCId	TC10.7
Test case	Client user selects media from “push” list of promoted content
Initial conditions	Valid content for query available in database
Configuration description	P&P “pushed” programmes to the distribution server (see test case 10.4)
Description of functionality to be tested	Client user selects media from “push” list of promoted content and receives media in correct format from Distribution Server.
Partners, people involved	End-user, Distributors, and Content Producer (must have initiated P&P)
Validator(s) skill	None (Typical end users does not need any experience)
Data set used	Different programmes addressing a selected scope of the content distributed within AXMEDIS
Steps	1 The distribution server pushes programmes to the clients 2 The end-user selects one or multiple AXMEDIS objects 3 The selection is transferred to the distribution server. 4 The distribution server delivers content to the clients (content is already provided to the distribution server as described in test case 10.4)
Expected results	End user receives selected content.
Variations	• End user cancels request.

Issues	None
Test case Scope/Type	GUI, Backend, Blackbox

10.8 Client user queries content from database (FHGIGD)

TCId	TC10.8
Test case	Client user queries content from database
Initial conditions	Content matching query available in database
Configuration description	AXMEDIS Query Support is accessible via Query Support for Distribution Channels.
Description of functionality to be tested	Client user requests content by querying it and receives media in correct format from Distribution Server
Partners, people involved	Enduser
Validator(s) skill	None (Typical end users does not need any experience)
Data set used	AXDS-P&P3
Steps	1 Client user makes query for AXMEDIS objects. 2 The Query Support for Clients creates a query and forwards it to the Query Support for Distribution Channels. 3 The Query Support For Distribution Channels transforms the client query and propagates the result to the AXMEDIS Query Support. 4 A list of matching AXMEDIS objects is returned by the Query Support For Distribution Channels to the Query Support for Clients. 5 The Query Support for Clients processes the query results. 6 The client GUI presents the query results to the end-user. The following steps (besides 9.2) are similar to the previous test case (TC10.7). 7 The end-user selects one or multiple AXMEDIS objects from the query result list. 8 The selection is transferred to the distribution server. 9 The distribution server delivers content: 9.1 If content is available in the correct format it is directly delivered to the client. 9.2 If content is not available in the correct format a workflow process is initiated by the PnP to produce the content in the specific format (on demand).
Expected results	Content is returned to client user
Variations	• Different types of content: audio, images, video, documents, adaptive (multi-level) interfaces • Adaptive (multi-level) interfaces • Enduser cancels request • Using stored queries (potentially illegal requests for content are possible, e.g. due to national laws/licensing issues)
Issues	None
Test case Scope/Type	P&P, Query Support For Clients, Query Support for Distribution Channels

11 AXMEDIS AXEPTOOLS for Satellite Data Broadcast on B2B (EUTELSAT, CRS4)

11.1 AXMEDIS B2B Client Application (CRS4)

11.1.1 B2B Client Installation

TCId	TC11.1.1
Test case	B2B Client Installation
Initial conditions	The installation procedure is completed and the professional user has executed all steps related to the client installation. The spectrum analyser (used by the installer of the satellite dish) indicates a good quality of signal of the satellite from which the B2B Satellite Data Broadcast will be received. The backend shall be running and sending some basic data through the Satellite B2B Channel.
Configuration description	The satellite adapter, using either the standard tuning application or the AXMEDIS Client integrated tuner, has been configured with the correct parameters to lock the satellite signal coming from the transponder where the B2B Satellite Data Broadcast is transmitted.
Description of functionality to be tested	Well functioning of the AXMEDIS B2B Client after installation: • The DVB Adapter is able to lock the signal coming from the appropriate transponder; • The DVB Adapter is able to switch from a transponder to another; • The AXMEDIS B2B Client is able to run correctly; • The AXMEDIS B2B Client does not create any conflicts with the previously installed applications; • The AXMEDIS B2B Client is able to stop its execution.
Partners, people involved	The AXMEDIS professional user, the AXMEDIS Satellite Data Broadcast B2B Distributor, a professional installer of satellite dishes
Validator(s) skill	Users should be familiar with computers. Users have to be able to interact with an Operating System Interface.
Data set used	AXDS-ITV1
Steps	1 The User launches the B2B AXMEDIS Client (e.g., using either a desktop shortcut or a explicitly provided launch script); 2 The User checks that no apparent conflicts arise after installing the B2B AXMEDIS Client: 2.1 Ethernet card activity; 2.2 Video adapter works well; 2.3 Other application using multicast protocol are not interfering with AXMEDIS data transfer; 3 The User checks that the B2B AXMEDIS Client application is working correctly: 3.1 A special 'guide' file should have a recent date (less than 2 minutes); 3.2 The special 'guide' file is updated regularly; 3.3 The special 'log' files do not indicate errors or warnings. 4 The User stops the B2B Client Application.
Expected results	The AXMEDIS B2B Client (limited to basic functionalities) works fine: • The AXMEDIS B2B Client starts/stops and behaves correctly; • All previously installed application still works fine while B2B AXMEDIS Client is running.
Variations	• The B2B AXMEDIS Client can automatically be launched at system start up.
Issues	In case of problems, the User should contact the Satellite Data Broadcast B2B

	Distributor for troubleshooting.
Test case Scope/Type	GUI, Backend / WhiteBox

11.1.2 B2B Client Customization

TCId	TC11.1.2
Test case	B2B Client Customization
Initial conditions	The installation procedure is completed and the professional user has executed all steps related to the client installation.
Configuration description	The professional user has configured all settings for the B2B service: • Firewall; • Configuration Files; • Updating of internal packages of the OS; • Remote control application.
Description of functionality to be tested	Well functioning of the AXMEDIS B2B Client after customization: • The incoming stream data is able to pass through firewalls (internal and external) and arrives to the B2B receiving station; • The B2B Station is remotely reachable by the previously installed application; • The Remote Control Application allows a total control of the B2B Receiving Station; • The B2B Receiving Station works well after the operating system update;
Partners, people involved	The AXMEDIS professional user, the AXMEDIS Satellite Data Broadcast B2B Distributor
Validator(s) skill	Users have to be able to • Configure a software application • Keep an operating system up to date.
Data set used	AXDS-ITV2
Steps	1 The User verifies that data streams are received on the B2B Station (delete the guide file – see Electronic Programme Guide – and wait it to reappear among the system files) after firewalls configuration; 2 The User verifies that no apparent conflicts arise after installing on the OS all required internal modules; 3 The User verifies that data streams are being received on the B2B Station (delete the guide file and wait to be reappeared among the system files) after modifying configuration files; 4 The User checks reaction time of the B2B Satellite Data Broadcast Provider after the trouble is submitted; 5 The User checks that no newer versions are available for the B2B Client Application component (e.g., drivers, software setup, and additional modules).
Expected results	The AXMEDIS B2B Client is well configured. The B2B AXMEDIS Objects pass through firewalls. The B2B Receiving Station is remotely reachable and controllable. The B2B AXMEDIS Client uses last version of needed components (drivers, modules). The B2B AXMEDIS Distributor is able to assist the professional User in troubleshooting during the configuration phase.
Variations	• The test of the B2B Client configuration could produce a quality label like “tested and approved by…”.
Issues	None
Test case Scope/Type	GUI, Backend / WhiteBox

11.1.3 B2B Client Registration

TCId	TC11.1.3
Test case	B2B Client Registration
Initial conditions	The B2B AXMEDIS Client is well installed and it works fine (able to receive basic data from the B2B AXMEDIS Satellite Data Channel). The professional User has executed the registration procedure. The backend shall be up and running and able to treat all incoming registration request from the professional Users.
Configuration description	The Internet Connection is able to reach the server for registering the B2B AXMEDIS Client Application.
Description of functionality to be tested	The User has finished his registration procedure and now is able to completely access the B2B AXMEDIS Service. The B2B Client Application has received all authorizations in order to receive all type of contents. The B2B Client Application has received some filters in order to receive only the really interesting content.
Partners, people involved	AXMEDIS professional User, B2B Satellite Data Broadcaster.
Validator(s) skill	Professional Users should be familiar with computers.
Data set used	AXDS-ITVlogin, AXDS-ITVpreferences
Steps	1 The User verifies that the Registration finishes with no errors. 2 The User verifies the correct reception of all Authorizations associated with the test login (this verification could need a while to be finished because Authorizations are simultaneously distributed to all Users). 3 The User verifies the correct reception of all Filters associated with the test login.
Expected results	The professional User has successfully finished the B2B Client Registration; he has completely received all related Authorizations; he has completely received all related Filters. The B2B Client Application can completely access the AXMEDIS Selection related to the received filters.
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / WhiteBox

11.2 Enabling a B2B Receiving Station

TCId	TC11.2
Test case	B2B Enabling Receiving Station
Initial conditions	The AXMEDIS Distributor accesses to the AXMEDIS B2B User Interface in order to control the set of B2B Receiving Stations controlled by him. The AXMEDIS Distributor has asked to use the B2B AXMEDIS Satellite Data Broadcast to the B2B Satellite Data Broadcaster. The backend shall be up and running and able to serve all requests coming from the AXMEDIS Distributor.
Configuration description	The AXMEDIS Distributor communicates a set of IP addresses authorized to access to the User Administration Environment. Internet Connection is able to reach the server for managing the B2B AXMEDIS Receiving Stations.
Description of functionality to be tested	The AXMEDIS Distributor is able to display all Users controlled by him. The AXMEDIS Distributor is able to add/modify/delete a controlled B2B Receiving Station.

Partners, people involved	AXMEDIS Distributor, B2B Satellite Data Broadcaster.
Validator(s) skill	AXMEDIS Distributor should be familiar with server environment, he should be able to launch scripts, scroll a log file, and repeat an operation following some step-by-step instructions.
Data set used	AXDS-ITVloginB, AXDS-ITVstations
Steps	1 The Distributor accesses to the User Admin Interface. 2 The Distributor displays all controlled B2B Receiving Stations. 3 The Distributor manages a given B2B Receiving Station.
Expected results	The Distributor has the full control on all B2B Receiving Stations. The Distributor can add/modify/delete a B2B Receiving Station in the B2B AXMEDIS Satellite Data Broadcast System.
Variations	None
Issues	None
Test case Scope/Type	Backend / WhiteBox

11.3 Downloading AXMEDIS Objects from AXEPTool by using Satellite Data Broadcast on B2B

TCId	TC11.3
Test case	Download Object from AXEPTool for the Satellite Data Broadcast
Initial conditions	The AXMEDIS Distributor accesses to the AXEPTool and is able to choose the way to load the Object from the AXEPTool. The backend shall be up and running and able to serve all requests coming from the AXMEDIS Distributor. Some AXMEDIS Objects are available after the submitted query.
Configuration description	An Internet Connection able to deal with the AXEPTool architecture.
Description of functionality to be tested	The AXMEDIS Distributor is able to choose the Download Transfer mode (either P2P or Satellite Data Broadcast) in order to pick up some AXMEDIS Object from the AXEPTool.
Partners, people involved	AXEPTool User, AXEPTool Manager, B2B Satellite Data Broadcaster.
Validator(s) skill	Familiarity with P2P Networks. Knowledge of server communication.
Data set used	AXDS-ITVobjects
Steps	1 The Actor selects one or more objects 2 The Actor chooses the Download Transfer mode (P2P, Satellite Data Broadcast) 3 The Actor starts the download task in AXEPTool 4 The Actor verifies the DRM rules, protections and licensing aspects 5 The Actor checks the download status, showed in a particular view of the AXEPTool (the status is obtained from the Push Server, by calling a specified API).
Expected results	The Actor is able to choose the delivery of a content present in the AXEPTool. The Actor has received the content into his B2B Satellite Data Broadcast Storage section.
Variations	None.
Issues	None.
Test case Scope/Type	Backend / WhiteBox

11.3.1 Pushing an AXMEDIS Object by B2B Carousel

TCId	TC11.3.1

Test case	Push of an AXMEDIS Object by B2B Carousel
Initial conditions	The AXMEDIS Distributor has received some AXMEDIS Objects by the AXEPTool P2P Network. The AXMEDIS Distributor has a set of enabled B2B receiving stations. The backend shall be up and running and able to serve all requests coming from the AXMEDIS Distributor.
Configuration description	An Internet Connection able to deal with the Satellite Data Broadcast Interface.
Description of functionality to be tested	The AXMEDIS Distributor is able to schedule the received AXMEDIS Objects for a simultaneous delivery to the enabled B2B Receiving Stations.
Partners, people involved	AXMEDIS Distributor, B2B Satellite Data Broadcaster.
Validator(s) skill	Familiarity with Professional Managing Interfaces.
Data set used	AXDS-ITVschedule, AXDS-ITVobjects, AXDS-ITVstations
Steps	1 The Distributor packages the downloaded content to be compatible with the Satellite Data Broadcast system 2 The Distributor selects the group of authorized receiving B2B stations to associate with the AXMEDIS Content 3 The Distributor associates the selected Object to a given Programme (the Programme is charged of transmitting the Carousel sequence) 4 The Distributor schedules the Programme for transmission
Expected results	The Distributor is able to package and schedule some previously received AXMEDIS Objects (from the AXEPTool P2P network), in order to delivery them to the enabled B2B Receiving Stations. Each enabled B2B receiving station has received the content into his local cache.
Variations	None.
Issues	None.
Test case Scope/Type	Backend / WhiteBox

11.3.2 Updating AXMEDIS Content by B2B Carousel

TCId	TC11.3.2
Test case	Update an AXMEDIS Content by B2B Carousel
Initial conditions	The AXMEDIS Synchronizer has produced some general updates to send to all AXMEDIS Distributors. The backend shall be up and running and able to serve all requests coming from the AXMEDIS Synchronizer.
Configuration description	An Internet Connection able to deal with the Satellite Data Broadcast Interface.
Description of functionality to be tested	The AXMEDIS Synchronizer is able to schedule the updates of AXMEDIS Content for a simultaneous delivery to the enabled B2B Receiving Stations of the AXMEDIS Distributors.
Partners, people involved	AXMEDIS Synchronizer, AXMEDIS Distributors, B2B Satellite Data Broadcaster.
Validator(s) skill	Familiarity with Professional Managing Interfaces.
Data set used	AXDS-ITVschedule, AXDS-ITVobjects, AXDS-ITVstations
Steps	1 The Distributor produces the periodic update 2 The Distributor uploads the updates packages to the Satellite Data Broadcast storage server (by HTTP, FTP) 3 The Distributor packages the updates content to be compatible with the Satellite Data Broadcast system 4 The Distributor selects the group of authorized receiving B2B stations to

	associate with the AXMEDIS Content 5 The Distributor associates the selected Updates to a given Programme (the Programme is charged of transmitting the Carousel sequence) 6 The Distributor schedules the Programme for transmission
Expected results	The Distributor is able to upload, package and schedule some previously produced AXMEDIS Content Updates in order to delivery them to the enabled B2B Receiving Stations. Each enabled B2B receiving station has received the update into his local cache.
Variations	None.
Issues	None.
Test case Scope/Type	Backend / WhiteBox

11.4 Automatic Content Reception via satellite

TCId	TC11.4
Test case	Automatic Content Reception
Initial conditions	The B2B Client Application has automatically started a download of an AXMEDIS Object addressed to him.
Configuration description	The B2B AXMEDIS Client Application is up and running.
Description of functionality to be tested	The professional user can check the increasing size of the folder containing the incoming Object.
Partners, people involved	Professional User of the B2B Receiving Station.
Validator(s) skill	User is able to interact with an operating system from a shell.
Data set used	AXDS-ITVobjects
Steps	1 The user checks that the folder size of the incoming AXMEDIS Object is increasing.
Expected results	The size of the specified folder is constantly increasing and if the User waits for a while he will find a special flag file indicating the end of the transmission.
Variations	None
Issues	None
Test case Scope/Type	Application Core Test / WhiteBox

11.5 Content Delivery via satellite

TCId	TC11.5
Test case	Content Delivery
Initial conditions	The B2B Client Application has finished receiving (automatically) the AXMEDIS Object. The B2B Client Application has delivered the Object to the appropriate application. The specified Application has executed all actions associated with the Object reception.
Configuration description	The B2B AXMEDIS Client Application is up and running.
Description of functionality to be tested	The professional user can check the execution of all actions associated with the Object.
Partners, people involved	Professional User of the B2B Receiving Station.

Validator(s) skill	User is able to interact with an operating system from a shell.
Data set used	AXDS-ITV3
Steps	1 The user checks in the Original Location that the Object reception is finished. 2 The user checks the actions log file to verify the real execution of the associated actions.
Expected results	The log file contains a final successful log line.
Variations	None
Issues	None
Test case Scope/Type	Application Core Test / WhiteBox

11.6 Content Protection for satellite distribution

TCId	TC11.6
Test case	Content Protection
Initial conditions	The AXMEDIS Object has been protected at the transport level before scheduling. The B2B Client Application has been well registered and it has received all Authorizations during this phase. The AXMEDIS Decryptor is well functioning.
Configuration description	The B2B AXMEDIS Client Application is up and running. The B2B AXMEDIS Client is able to recognize an encrypted AXMEDIS Object.
Description of functionality to be tested	The professional user can verify that the Decryptor is running during the encrypted transmission. At the end of transmission, the professional user is able to open the received Object.
Partners, people involved	Professional User of the B2B Receiving Station.
Validator(s) skill	User is able to interact with an operating system from a shell.
Data set used	AXDS-ITV3
Steps	1 The user checks in the Original Location that the Object reception is finished. 2 The user checks the actions log file to verify the real execution of the Decryptor. 3 The user opens the received Object.
Expected results	The log file shows that the decrypting box is working fine. The user is able to open the Object protected during the transmission at the transport level.
Variations	None
Issues	None
Test case Scope/Type	Application Core Test / WhiteBox

12 AXMEDIS Protection Tools (FUPF, EXITECH, FHGIGD, DSI)

12.1 Super AXCS (DSI)

12.1.1 AXMEDIS Registration of AXCSs (DSI)

TCId	TC12.1.1
Test case	Registration of an AXCSs on SuperAXCS
Initial conditions	None
Configuration description	An AXCS Manager wants to register his AXCS to SuperAXCS. Data should be transferred from AXCS to SuperAXCS.
Description of functionality to be tested	Registration of an AXCS on SuperAXCS
Partners, people involved	AXCS Managers (often Distributors)
Validator(s) skill	None
Data set used	AXDS-AXCS1
Steps	1 Actor submits the test registration 2 The system returns results 3 The results are checked against the list of expected results
Expected results	SuperAXCS registers users correctly
Variations	• An actor tries a registration with missing data o The service has to return an error code • An actor tries a registration with incorrect data o The service has to return an error code • An actor tries a registration already done o The service has to return an error code
Issues	None
Test case Scope/Type	GUI / BlackBox

12.1.2 Tool/device off-line registration (DSI)

TCId	TC12.1.2
Test case	Tool/device off-line registration
Initial conditions	None
Configuration description	An Actor wants to register a new kind of tool in the AXMEDIS network.
Description of functionality to be tested	Registration of a new kind of tool in the AXMEDIS network.
Partners, people involved	AXMEDIS tool producer (i.e. a software house producing a specified tool to use it in the AXMEDIS system), AXMEDIS SuperAXCS Managers.
Validator(s) skill	None
Data set used	AXDS-AXCS2
Steps	1 AXMEDIS tool producer submits the tool/device for test registration 2 The system returns results 3 The results are checked against the list of expected results 4 The SuperAXCS Manager register the received tool/device in the system 5 The results are checked against the list of expected results
Expected results	SuperAXCS registers tools/devices correctly
Variations	• An actor tries a registration with missing data o The service has to return an error code • An actor tries a registration with incorrect data

	o The service has to return an error code • An actor tries a registration already done o The service has to return an error code
Issues	Note that the Actor could be either an AXMEDIS tool producer or the SuperAXCS Manager
Test case Scope/Type	GUI / BlackBox

12.1.2.1 Certification and off-line registration of AXMEDIS tools

TCId	TC12.1.2.1a
Test case	Certification and off-line registration of AXMEDIS tools
Initial conditions	None
Configuration description	None
Description of functionality to be tested	Check that tools that want to become part of the AXMEDIS framework and that accomplish AXMEDIS guidelines are certified and registered
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-AXCS2
Steps	1 Reception of the tool that wants to be certified 2 Off-line checking and test that tool accomplishes AXMEDIS guidelines 3 Create Tool ID, estimate Tool fingerprint and other major parameters
Expected results	The tool is certified, i.e. tool fingerprint is estimated and other major parameter are extracted The tool is registered in the AXCS Registration and Certification Database with all the information collected at the step 1 by using the AXMEDIS SW Tools off-line Registration
Variations	None
Issues	None
Test case Scope/Type	Backend/ BlackBox

TCId	TC12.1.2.1b
Test case	Certification and off-line registration of AXMEDIS tools
Initial conditions	None
Configuration description	None
Description of functionality to be tested	Check that tools that want to become part of the AXMEDIS framework and that do not accomplish AXMEDIS guidelines are not certified nor registered
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-AXCS2
Steps	1 Reception of the tool that wants to be certified 2 Off-line checking and test that tool does not accomplish AXMEDIS guidelines
Expected results	The tool is not certified nor registered
Variations	None
Issues	None
Test case Scope/Type	Backend/ BlackBox

12.1.2.2 Certification and off-line registration of tools that use AXMEDIS framework

TCId	TC12.1.2.2a
Test case	Certification and off-line registration of tools that use AXMEDIS framework
Initial conditions	None
Configuration description	None
Description of functionality to be tested	Check that tools that internally use AXMEDIS framework but are not part of it and want to be inserted into the AXMEDIS framework (or certified as being AXMEDIS compliant) are certified and registered only if AXMEDIS framework components and tools used by the tool under test have not been modified and accomplish AXMEDIS guidelines and the new tool accomplishes AXMEDIS guidelines, if applicable
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-AXCS2
Steps	1 Reception of the tool that wants to be certified 2 Verification that AXMEDIS framework components and tools used by the tool under test have not been modified and accomplish AXMEDIS guidelines 3 Off-line checking and test that the new tool accomplishes AXMEDIS guidelines, if applicable
Expected results	The tool is certified, i.e. tool fingerprint is estimated and other major parameter are extracted The tool is registered in the AXCS Registration and Certification Database with all the information collected at the step 1 by using the AXMEDIS SW Tools off-line Registration
Variations	None
Issues	None
Test case Scope/Type	Backend/ BlackBox

TCId	TC12.1.2.2b
Test case	Certification and off-line registration of tools that use AXMEDIS framework
Initial conditions	None
Configuration description	None
Description of functionality to be tested	Check that tools that internally use AXMEDIS framework but are not part of it and want to be inserted into the AXMEDIS framework (or certified as being AXMEDIS compliant) are not certified nor registered, if AXMEDIS framework components and tools used by the tool under test have been modified and do not accomplish AXMEDIS guidelines and/or the new tool does not accomplish AXMEDIS guidelines, if applicable
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-AXCS2
Steps	• Reception of the tool that wants to be certified • Verification that AXMEDIS framework components and tools used by the tool under test have been modified and do not accomplish AXMEDIS guidelines and/or • Off-line checking and test that the new tool does not accomplish AXMEDIS guidelines, if applicable

Expected results	The tool is not certified nor registered
Variations	None
Issues	None
Test case Scope/Type	Backend/ BlackBox

12.1.3 AXMEDIS Object ID Generator

12.1.3.1 Generation of unique object ID

TCId	TC12.1.3.1
Test case	Generation of unique object ID
Initial conditions	Object ID created, AXMEDIS Editor opened and database connection available.
Configuration description	An actor wants to associate with the AXMEDIS Editor an AXMEDIS Object ID to the newly created object.
Description of functionality to be tested	The AXMEDIS Object ID generated is unique and the information saved is correct.
Partners, people involved	Integrator, Designer
Validator(s) skill	None
Data set used	AXDS-OIDGen1
Steps	1 Object ID and more information are saved into a database. The Object ID will be saved as primary key. If there is not any error message from database, the Object ID will be unique.
Expected results	A new database register with information about the AXMEDIS Object
Variations	None
Issues	None
Test case Scope/Type	Backend / UnitTest

12.1.4 Global Object List WEB Service (DSI)

12.1.4.1 Search of AXMEDIS Objects (via web interface) (DSI)

TCId	TC12.1.4.1
Test case	Search of AXMEDIS Objects (via web interface)
Initial conditions	AXMEDIS system is filled with a predefined set of objects (AXDS-AXCS3), internet browser is running, network connection is available.
Configuration description	An Actor wants to perform a search in the AXMEDIS database to retrieve a set of AXMEDIS Objects satisfying several conditions
Description of functionality to be tested	The AXMEDIS Object is found if is present in database and is not found if is not present in database. It is possible to search object using different self-composing criteria.
Partners, people involved	End User, Distributors
Validator(s) skill	None
Data set used	AXDS-AXCS3 as initial condition and AXDS-AXCS4 to perform searches in database
Steps	1 An actor fill the search form with some data and send search request 2 The system return results 3 The results are checked against the list of expected results
Expected results	The list of results is compliant with the data included in database

Variations	• An actor fills search form with data not included in database o The service has to return an empty results list • An actor fills search form with no data o The service has to return an error code • An actor fills search form with inconsistent data o The service has to return an error code • An actor fills search form with random data o The service has to return a result consistent with data stored in database
Issues	None
Test case Scope/Type	GUI / BlackBox

12.1.4.2 Search of AXMEDIS Objects (inside an AXMEDIS tool)

TCId	TC12.1.4.2
Test case	Search of AXMEDIS Objects (query is composed inside an AXMEDIS tool)
Initial conditions	AXMEDIS system is filled with a predefined set of objects (AXDS-AXCS3), AXMEDIS tool is running, network connection is available.
Configuration description	An Actor wants to perform a search in the AXMEDIS database to retrieve a set of AXMEDIS Objects satisfying several conditions
Description of functionality to be tested	The AXMEDIS Object is found if is present in database and is not found if is not present in database. It is possible to search object using different self-composing criteria.
Partners, people involved	Distributors, SuperAXCS Managers
Validator(s) skill	None
Data set used	AXDS-AXCS3 as initial condition and AXDS-AXCS4 to perform searches in database
Steps	1 An actor fills the search form with some data and sends search request 2 The system returns results 3 The results are checked against the list of expected results
Expected results	The list of results is compliant with the data included in database
Variations	• An actor fill search form with data not included in database o The service has to return an empty results list • An actor fill search form with no data o The service has to return an error code • An actor fill search form with inconsistent data o The service has to return an error code • An actor fill search form with random data o The service has to return a result consistent with data stored in database
Issues	None
Test case Scope/Type	GUI / BlackBox

12.1.5 Super AXCS Collector

12.1.5.1 On-line transfer between AXCS and Super AXCS

TCId	TC12.1.5.1
Test case	On-line transfer between AXCS and Super AXCS
Initial conditions	AXMEDIS system is filled with a predefined set of objects (AXDS-AXCS3), network connection is available

Configuration description	Some information managed by AXCS during an AXMEDIS Object usage has to be transferred to Super AXCS. This transfer involves AXCS Synchronizer and Super AXCS Collector
Description of functionality to be tested	Transfer between AXCS Synchronizer and Super AXCS Collector, AXCS Synchronizer queue capabilities, SuperAXCS Collector capabilities
Partners, people involved	AXCS Managers, SuperAXCS Managers
Validator(s) skill	None
Data set used	AXDS-AXCS3 as initial condition and AXDS-AXCS5 to perform the test
Steps	1 A meaningful set of data about object usage is composed and stored in AXCS database to be transferred to SuperAXCS Collector 2 AXCS Synchronizer transfer functionality is started with the correct parameters to transfer data to SuperAXCS Collector
Expected results	The transfer occurs in the correct way, data is transferred completely with no error and inserted in Super AXCS database, all the needful data is transferred
Variations	• The connection is interrupted o The service has to return an error code o AXCS Synchronizer has to store data to AXCS Synchronizer Queue • Data is corrupted through the transfer o The service has to return an error code
Issues	None
Test case Scope/Type	Backend / WhiteBox

12.1.5.2 Off-line synchronization between AXCS and Super AXCS

TCId	TC12.1.5.2
Test case	Off-line transfer between AXCS and Super AXCS
Initial conditions	AXMEDIS system is filled with a predefined set of objects (AXDS-AXCS3) and data about AXCSs (AXDS-AXCS6), network connection is available
Configuration description	Some information collected by AXCS during an AXMEDIS Object usage has to be transferred to Super AXCS even if the connection between AXCS and SuperAXCS is interrupted. In this case the transfer doesn't occur on-line during the Object usage, but off-line in a second time
Description of functionality to be tested	Transfer between AXCS Synchronizer and Super AXCS Collector, AXCS Synchronizer queue capabilities, SuperAXCS Collector capabilities
Partners, people involved	End Users, AXCS Managers, SuperAXCS Managers
Validator(s) skill	Medium/High AXMEDS system knowledge
Data set used	AXDS-AXCS3 and AXDS-AXCS6 as initial condition and AXDS-AXCS5 to perform the test
Steps	1 A meaningful set of data about object usage is composed and stored in AXCS database to be transferred to SuperAXCS Collector 2 Step 1 repeated for a meaningful number of AXCS 3 SuperAXCS Collector Off-line Synchronization functionality has been started 4 Wait until the test is finished and then control results
Expected results	The transfer occurs in the correct way, data is transferred completely with no error and inserted in Super AXCS database, all the needful data is transferred, all the AXCS Synchronizer queues are empty
Variations	• The connection is interrupted o The service has to return an error code o The pertinent AXCS Synchronizer queue is as the beginning of the

	test - Data is corrupted through the transfer - The service has to return an error code - The pertinent AXCS Synchronizer queue is as the beginning of the test
Issues	None
Test case Scope/Type	Backend / WhiteBox

12.2 AXMEDIS Certifier and Supervisor (WP5.6.1: DSI)

12.2.1 AXMEDIS Registration Service (DSI)

12.2.1.1 End User registration in a distribution channel (DSI)

TCId	TC12.2.1.1
Test case	Registration of an End User in a distribution channel
Initial conditions	AXMEDIS system is filled with a predefined set of user registrations
Configuration description	An actor wants to register in a channel. Data should be transferred from client to Distributor and then to AXCS. The registered End User should receive a confirmation (via email)
Description of functionality to be tested	Registration of an End user in a distribution channel
Partners, people involved	Distributors, End Users
Validator(s) skill	None
Data set used	AXDS-CS7
Steps	1 Actor submits the test registration 2 The system returns results 3 The results are checked against the list of expected results, including insertions in database and email sending/receiving
Expected results	AXCS registers users correctly
Variations	- An actor tries a registration with missing data - The service has to return an error code - An actor tries a registration with incorrect data - The service has to return an error code - An actor tries a registration already done - The service has to return an error code
Issues	None
Test case Scope/Type	GUI / BlackBox

12.2.1.2 End User registration in a different distribution channel (DSI)

TCId	TC12.2.1.2
Test case	Registration of an End User in a different distribution channel
Initial conditions	AXMEDIS system is filled with a predefined set of user registrations (AXDS-AXCS8)
Configuration description	An actor registered in a distribution channel wants to register in another distribution channel. Data should be transferred from client to Distributor and then to AXCS. The registered End User should receive a confirmation (via email)
Description of	Registration of an End user in a distribution channel different from the one the

functionality to be tested	user is already registered in
Partners, people involved	Distributors, End Users
Validator(s) skill	None
Data set used	AXDS-AXCS8 as initial condition, AXDS-AXCS7 to perform the test
Steps	1 Actor submits the test registration 2 The system returns results 3 The results are checked against the list of expected results, including insertions in database and email sending/receiving
Expected results	AXCS registers users correctly
Variations	• An actor tries a registration but he is not registered in any distribution channel o The service has to return an error code • An actor tries a registration with missing data o The service has to return an error code • An actor tries a registration with incorrect data o The service has to return an error code • An actor tries a registration already done o The service has to return an error code
Issues	None
Test case Scope/Type	GUI / BlackBox

12.2.1.3 Registration of a new Teacher/School or Student (DSI)

TCId	TC12.2.1.3
Test case	Registration of a new Teacher/School or Student
Initial conditions	AXMEDIS system is filled with a predefined set of user registrations (AXDS-AXCS8)
Configuration description	A Teacher/School wants to register in the AXMEDIS network itself and his/its Students. Student's registration is linked to the pertinent Teacher/School registration. Data should be transferred from client to Distributor and then to AXCS. The Actor should receive a confirmation (via email)
Description of functionality to be tested	Registration of a new Teacher/School and of the pertinent students.
Partners, people involved	Distributors, End Users (in particular Teachers, School, Students)
Validator(s) skill	None
Data set used	AXDS-AXCS8 as initial condition, AXDS-AXCS7 to perform the test
Steps	1 Actor submits the test registration (itself and pertinent students data) 2 The system returns results 3 The results are checked against the list of expected results, including insertions in database and email sending/receiving
Expected results	AXCS registers users correctly
Variations	• An actor tries a registration with missing data o The service has to return an error code • An actor tries a registration with incorrect data o The service has to return an error code • An actor tries a registration already done o The service has to return an error code
Issues	None
Test case Scope/Type	GUI / BlackBox

12.2.1.4 Registration of an old User of the Channel on AXMEDIS (DSI)

TCId	TC12.2.1.4
Test case	Registration of an Old User of the Channel on AXMEDIS
Initial conditions	AXMEDIS system is filled with a predefined set of user registrations (AXDS-AXCS8)
Configuration description	A Distributor wants to register an End User in the AXMEDIS network. Data should be transferred from Distributor to AXCS. The registered End User should receive a confirmation (via email)
Description of functionality to be tested	The User is registered on the Distributor but not on AXMEDIS. The Distributor wants his old users become AXMEDIS users
Partners, people involved	Distributors, End Users
Validator(s) skill	None
Data set used	AXDS-AXCS8 as initial conditions, AXDS-AXCS7 to perform the test
Steps	1 Actor submits the test registration 2 The system returns results 3 The results are checked against the list of expected results, including insertions in database and email sending/receiving
Expected results	AXCS registers users correctly
Variations	• An actor tries a registration but he is not registered in any distribution channel o The service has to return an error code • An actor tries a registration with missing data o The service has to return an error code • An actor tries a registration with incorrect data o The service has to return an error code • An actor tries a registration already done o The service has to return an error code
Issues	None
Test case Scope/Type	GUI / BlackBox

12.2.1.5 User password modification

TCId	TC12.2.1.7
Test case	User password modification
Initial conditions	AXMEDIS system is filled with a predefined set of users registrations (AXDS-AXCS8)
Configuration description	AXMEDIS Registration and Certification database
Description of functionality to be tested	User password modification
Partners, people involved	All Users
Validator(s) skill	None
Data set used	AXDS-AXCS8 as initial conditions and AXDS-AXCS9 to perform the test
Steps	1 Actor submits the password modification request 2 The system returns results 3 The results are checked against the list of expected results
Expected results	AXCS changes passwords correctly
Variations	• An actor tries to change a password to a user not registered yet o The service has to return an error code • An actor tries to change a password to a user, but old password is incorrect

	- The service has to return an error code - An actor tries to change a password to a new invalid password - The service has to return an error code
Issues	Password requirements
Test case Scope/Type	GUI / BlackBox

12.2.1.6 Registration of users regarding CMS aspects

TCId	TC12.2.1.8a
Test case	Registration of users already registered in a content partner CMS
Initial conditions	A CMS has its own anagraphic DB of users and profiles: in this DB there is a flag like "allow this user to use AXMEDIS system" (FALSE/TRUE)
Configuration description	A user already registered in a CMS should be registered at the same time into AXMEDIS system
Description of functionality to be tested	Crawling of CMS can include DB of users with the AXMEDIS flag set to true
Partners, people involved	Content providers, people involved in content crawling
Validator(s) skill	None
Data set used	AXDS-AXCS8 as initial condition, AXDS-AXCS7 to perform the test
Steps	1 User connects to the registration tool 2 User enters the required personal data to be registered in AXMEDIS 3 User clicks the "Register" button of the registration tool 4 Registration tool verifies data entered by the user 5 Registration tool inserts data into the users database 6 A certificate request is generated for this user 7 Registration tool sends response to the user, indicating that registration has been done and that he will receive a certificate soon.
Expected results	User already registered in a CMS is also registered in the system
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC12.2.1.8b
Test case	Registration of new users in a CMS of a CP which is also a content distributor
Initial conditions	A CMS has its own procedure for user registration. In this DB there is a flag like "allow this user to use AXMEDIS system" (FALSE/TRUE)
Configuration description	A new user wants to register in a content provider CMS and should be registered at the same time into AXMEDIS system
Description of functionality to be tested	The registration in a CMS is automatically taken also by AXMEDIS system
Partners, people involved	End user
Validator(s) skill	None
Data set used	AXDS-AXCS8 as initial condition, AXDS-AXCS7 to perform the test
Steps	1 User connects to the CMS registration tool 2 User enters the required personal data to be registered in CMS and in

	AXMEDIS 3 User check the appropriate box to be registered also for AXMEDIS 4 User clicks the "Register" button of the CMS registration tool 5 CMS Registration tool verifies data entered by the user 6 CMS Registration tool inserts data into the users database 7 The CMS' user DB is set ready to be read by AXMEDIS DB 8 AXMEDIS takes the registration from CMS 9 AXMEDIS registration tool verifies data entered by the user 10 AXMEDIS registration tool inserts data into the users database 11 A certificate request is generated for this user 12 Registration tool sends response to the user, indicating that registration has been done and that he will receive a certificate soon
Expected results	New user registered in a CMS is also registered in the AXMEDIS system
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

12.2.2 AXMEDIS Certification and Verification

12.2.2.1 Authentication of a Device

TCId	TC12.2.2.1a
Test case	Authentication of a Device
Initial conditions	The authentication is requested by PMS, AXCS, AXOM (or other subjects) or in any other way it is convenient
Configuration description	None
Description of functionality to be tested	Positive authentication of a device
Partners, people involved	AXMEDIS Tool and/or device
Validator(s) skill	High, technical
Data set used	AXDS-CertVer2
Steps	1 The AXCS (the AXMEDIS Certification and Verification) verifies that received information (in particular the status is important) is correct
Expected results	The AXCS (the AXMEDIS Certification and Verification) sends the response to the PMS The mentioned PMS sends the response to the AXMEDIS Tool: in this way the chain is closed
Variations	None
Issues	None
Test case Scope/Type	Backend

TCId	TC12.2.2.1b
Test case	Authentication of a Device
Initial conditions	The authentication is requested by PMS, AXCS, AXOM (or other subjects) or in any other way it is convenient
Configuration description	None
Description of functionality to be tested	Negative authentication of a device

Partners, people involved	AXMEDIS Tool and/or device
Validator(s) skill	High, technical
Data set used	AXDS-CertVer2
Steps	1 The AXCS (the AXMEDIS Certification and Verification) verifies that received information (in particular the status is important) is not correct
Expected results	The AXCS (the AXMEDIS Certification and Verification) sends the response to the PMS The mentioned PMS sends a "deactivation signal" to the AXMEDIS Tool The Device is deactivated by marking it as "blocked"
Variations	None
Issues	None
Test case Scope/Type	Backend

12.2.2.2 Certification of AXMEDIS Tool and User

TCId	TC12.2.2.2a
Test case	Certification of AXMEDIS Tool and User
Initial conditions	An Actor wants to certify a specified tool installed on a terminal (i.e. a PC, a Palmtop, a Phone, a Kiosk and so on) The user and tool are already registered The tool is used for the first time on the terminal by the user
Configuration description	None
Description of functionality to be tested	Certify an AXMEDIS tool that is used for the first time by an AXMEDIS User
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer3
Steps	1 The user opens the tool for its certification 2 AXOM (as a part of the tool) calculates fingerprint or extracts other features to identify the specified tool, the user and the terminal it is installed on 3 AXOM (as a part of the tool) contacts the pertinent PMS sending all the needful information for the registration 4 The mentioned PMS contacts the pertinent AXMEDIS Certification and Verification sending him all the received information 5 Check that the tool and user are registered and correct (check that the user data and status are correct and that the received Tool FP matches the original FP) 6 Check that it is the first use of the tool by the user. 7 The AXMEDIS Certification and Verification generates a TID (tool ID) and inserts it together with all the received information in the AXCS Database 8 The AXMEDIS Certification and Verification sends to PMS the generated TID 9 The PMS sends to AXOM (as a part of the tool) a certification confirmation message, including the TID 10 AXOM registers that the tool is certified and stores also the received TID
Expected results	The tool is certified in the AXMEDIS system The user is certified to have used the tool for the first time A new TID (tool id) is generated and bounded to the tool The requester receives notification about the certification

Variations	None
Issues	None
Test case Scope/Type	Backend

TCId	TC12.2.2.2b
Test case	Certification of AXMEDIS tool and user
Initial conditions	An Actor wants to certify a specified tool installed on a terminal (i.e. a PC, a Palmtop, a Phone, a Kiosk and so on) The user is not registered The tool is used for the first time on the terminal by the user
Configuration description	None
Description of functionality to be tested	Negative certification of the first use of an AXMEDIS tool done by an AXMEDIS User
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer3
Steps	1 The user opens the tool for its certification 2 AXOM (as a part of the tool) calculates fingerprint or extracts other features to identify the specified tool, the user and the terminal it is installed on 3 AXOM (as a part of the tool) contacts the pertinent PMS sending all the needful information for the registration 4 The mentioned PMS contacts the pertinent AXMEDIS Certification and Verification sending him all the received information 5 Check that the user is not registered
Expected results	The PMS is sent a message notifying the unsuccessful certification
Variations	• If the tool is registered (the received Tool FP matches the original FP), the user might be asked to register
Issues	None
Test case Scope/Type	Backend

TCId	TC12.2.2.2c
Test case	Certification of AXMEDIS tool and user
Initial conditions	An Actor wants to certify a specified tool installed on a terminal (i.e. a PC, a Palmtop, a Phone, a Kiosk and so on) The user is registered The tool is used for the first time on the terminal by the user The tool is not registered (the received Tool FP does not match the original FP)
Configuration description	None
Description of functionality to be tested	Negative certification of the first use of an AXMEDIS tool done by an AXMEDIS User
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer3
Steps	1 The user opens the tool for its certification 2 AXOM (as a part of the tool) calculates fingerprint or extracts other features

	to identify the specified tool, the user and the terminal it is installed on 3 AXOM (as a part of the tool) contacts the pertinent PMS sending all the needful information for the registration 4 The mentioned PMS contacts the pertinent AXMEDIS Certification and Verification sending him all the received information 5 Check that the user is registered 6 Check that the received Tool FP does not match the original FP
Expected results	The PMS is sent a message notifying the unsuccessful certification and the tool is quarantined, in order to check if it has to be blocked
Variations	• The tool is blocked
Issues	None
Test case Scope/Type	Backend

12.2.2.3 Verification of AXMEDIS users using AXMEDIS tools on a Device

TCId	TC12.2.2.3a
Test case	Verification of AXMEDIS users using AXMEDIS tools on a Device
Initial conditions	The user, tool and device are already registered in the database (the user has already used the tool on the device) The tool is connected to the AXMEDIS system
Configuration description	None
Description of functionality to be tested	Positive verification of user data, tool data and tool operation history consistency
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 The AXMEDIS Certification and Verification receives the following information: UID, TID, device FP, tool operation history and tool operation history FP 2 The AXMEDIS Certification and Verification (inside AXCS) checks that it is not the first use of the tool by the user on the device. Otherwise see "Certification of AXMEDIS User and Tool" test case 3 The AXMEDIS Certification and Verification retrieves the tool operation history fingerprint that is stored in the AXCS database (using AXCS Database interface) and checks that it matches to the received one
Expected results	The AXMEDIS Certification and Verification returns a notification message targeted to the PMS where it notifies that the user, tool, device and operation history have been verified
Variations	None
Issues	None
Test case Scope/Type	Blackbox

TCId	TC12.2.2.3b
Test case	Verification of AXMEDIS users using AXMEDIS tools on a Device
Initial conditions	The user, tool and device are already registered in the database (the user has already used the tool on the device) The tool is connected to the AXMEDIS system
Configuration description	None
Description of	Positive verification of user data, tool data, device data and tool operation history

functionality to be tested	consistency
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 The AXMEDIS Certification and Verification receives the following information: UID, TID, device FP, tool operation history and tool operation history FP 2 The Certification and Verification (inside AXCS) checks that it is not the first use of the tool by the user on the device. Otherwise see "Certification of AXMEDIS User and Tool" test case 3 The Certification and Verification retrieves the tool operation history fingerprint that is stored in the AXCS database (using AXCS Database interface) 4 The Certification and Verification checks that the retrieved tool operation history FP does not match to the received one 5 The Certification and Verification computes a new fingerprint, derived from the previous one (stored in the database) and the tool operation history (sent by the user). 6 The Certification and Verification compares the new fingerprint to the fingerprint provided by the user, and it matches.
Expected results	The AXMEDIS Certification and Verification returns a notification message targeted to the PMS where it notifies that the user, tool, device and operation history have been verified
Variations	None
Issues	None
Test case Scope/Type	Blackbox

TCId	TC12.2.2.3c
Test case	Verification of AXMEDIS users using AXMEDIS tools on a Device
Initial conditions	The user, tool and device are already registered in the database (the user has already used the tool on the device) The tool is connected to the AXMEDIS system
Configuration description	None
Description of functionality to be tested	Negative verification of tool operation history consistency
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 The AXMEDIS Certification and Verification receives the following information: UID, TID, device FP, tool operation history and tool operation history FP 2 The Certification and Verification (inside AXCS) checks that it is not the first use of the tool by the user on the device. Otherwise see "Certification of AXMEDIS User and Tool" test case 3 The Certification and Verification retrieves the tool operation history fingerprint that is stored in the AXCS database (using AXCS Database interface) 4 The Certification and Verification checks that the retrieved tool operation

	history FP does not match to the received one 5 The Certification and Verification computes a new fingerprint, derived from the previous one (stored in the database) and the tool operation history (sent by the user). 6 The Certification and Verification compares the new tool operation history fingerprint to the tool operation history fingerprint provided by the user and they do not match
Expected results	The AXMEDIS Certification and Verification returns a notification message targeted to the PMS where it notifies that the tool operation history is not consistent.
Variations	None
Issues	None
Test case Scope/Type	Blackbox

12.2.3 AXMEDIS Supervisor

12.2.3.1 User blocking

TCId	TC12.2.3.1
Test case	User blocking
Initial conditions	AXMEDIS system is filled with a predefined set of user registrations
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Blocking of a user
Partners, people involved	Publishers, Distributors, Final Users
Validator(s) skill	High, Technical
Data set used	AXDS-Supervisor1
Steps	1 Actor submits the blocking request 2 The system returns results 3 The results are checked against the list of expected results
Expected results	AXCS blocks users correctly
Variations	• An actor tries to block a user not registered yet o The service has to return an error code • An actor tries to block a user already blocked o The service has to return an error code
Issues	None
Test case Scope/Type	BlackBox

12.2.3.2 User unblocking

TCId	TC12.2.3.2
Test case	User unblocking
Initial conditions	AXMEDIS system is filled with a predefined set of user registrations
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Unblocking of a user

Partners, people involved	Publishers, Distributors, Final Users
Validator(s) skill	High, Technical
Data set used	AXDS-Supervisor1
Steps	1 Actor submits the unblocking request 2 The system returns results 3 The results are checked against the list of expected results
Expected results	AXCS unblocks users correctly
Variations	• An actor tries to unblock a user not registered yet o The service has to return an error code • An actor tries to unblock a user already unblocked o The service has to return an error code
Issues	None
Test case Scope/Type	Backend / BlackBox

12.2.3.3 Tool blocking

TCId	TC12.2.3.3
Test case	Tool blocking
Initial conditions	AXMEDIS system is filled with a predefined set of tool registrations
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Blocking of a tool
Partners, people involved	Publishers, Distributors, Final Users
Validator(s) skill	High, Technical
Data set used	AXDS-Supervisor2
Steps	1 Actor submits the blocking request 2 The system returns results 3 The results are checked against the list of expected results
Expected results	AXCS blocks tools correctly
Variations	• An actor tries to block a tool not registered yet o The service has to return an error code • An actor tries to block a tool already blocked o The service has to return an error code
Issues	None
Test case Scope/Type	Backend / BlackBox

12.2.3.4 Tool unblocking

TCId	TC12.2.3.4
Test case	Tool unblocking
Initial conditions	AXMEDIS system is filled with a predefined set of tool registrations
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Unblocking of a tool
Partners, people involved	Publishers, Distributors, Final Users

Validator(s) skill	High, Technical
Data set used	AXDS-Supervisor2
Steps	1 Actor submits the unblocking request 2 The system returns results 3 The results are checked against the list of expected results
Expected results	AXCS unblocks tools correctly
Variations	• An actor tries to unblock a tool not registered yet o The service has to return an error code • An actor tries to unblock a tool already unblocked o The service has to return an error code
Issues	None
Test case Scope/Type	Backend / BlackBox

12.2.3.5 AXMEDIS Protection information delivery

TCId	TC12.2.3.5
Test case	AXMEDIS Protection information delivery
Initial conditions	AXMEDIS User has a protected AXMEDIS object and need the corresponding key to unprotect and consume it. The AXMEDIS User has already asked for authorisation and received a positive response from PMS
Configuration description	None
Description of functionality to be tested	An AXMEDIS User receives information for unprotecting an AXMEDIS object it has
Partners, people involved	Final user, any user that can consume AXMEDIS protected object
Validator(s) skill	Advanced user
Data set used	AXDS-PMS1
Steps	1 AXMEDIS User has received a positive response from PMS and he can consume the protected AXMEDIS object 2 AXCS protects the protection information 3 The AXCS sends the protected protection information to the AXMEDIS User 4 AXMEDIS User unprotects the protection information and makes use of the AXMEDIS protected object
Expected results	The client has the protection information for content fruition
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

12.2.3.6 Association of protection information to an AXMEDIS Object

TCId	TC12.2.3.6a
Test case	Association of protection information to an AXMEDIS object
Initial conditions	AXMEDIS User, Protection tool engine wants to protect an AXMEDIS object The AXMEDIS object was not protected, and no protection information was stored for it
Configuration description	None
Description of functionality to be tested	An AXMEDIS User or the Protection tool engine wants to protect an AXMEDIS object
Partners, people	AXMEDIS User, AXMEDIS User operating Protection tool engine

involved	
Validator(s) skill	Advanced user
Data set used	AXDS-PMS1
Steps	1 Check that the AXMEDIS Object does not have associated protection information 2 If so, AXMEDIS Supervisor generates the corresponding protection key(s) 3 Apply the protection mechanism to the AXMEDIS object 4 Store the AXMEDIS Object protection information in the AXMEDIS Objects database
Expected results	The AXMEDIS Object is protected and the corresponding protection information is stored in the objects database
Variations	• The AXMEDIS Object had associated protection information. See next use case.
Issues	None
Test case Scope/Type	BlackBox

TCId	TC12.2.3.6b
Test case	Association of protection information to an AXMEDIS object (already protected)
Initial conditions	AXMEDIS User, Protection tool engine wants to protect an AXMEDIS object The AXMEDIS object was already protected
Configuration description	None
Description of functionality to be tested	An AXMEDIS User or the Protection tool engine wants to protect an AXMEDIS object already protected
Partners, people involved	AXMEDIS User, AXMEDIS User operating Protection tool engine
Validator(s) skill	Advanced user
Data set used	AXDS-PMS1
Steps	1 Retrieve protection information for the AXMEDIS object (see Query Protection information of an AXMEDIS Object) 2 Unprotect the AXMEDIS Object 3 AXMEDIS Supervisor generates the corresponding protection key(s) 4 Apply the new protection mechanism to the AXMEDIS object 5 Store the AXMEDIS Object protection information in the AXMEDIS Objects database. The former protection information has to be deleted or moved to a historical record
Expected results	The AXMEDIS Object is protected with a new protection key/algorithm and the corresponding protection information is stored in the objects database
Variations	• The AXMEDIS Object was not protected, then refer to the previous test case
Issues	It has to be marked in a field of the AXMEDIS database that the object has been re-protected in order to send new versions of the protected AXMEDIS object / protection keys to the users accessing it
Test case Scope/Type	BlackBox

12.2.3.7 Requesting of protection information of an AXMEDIS Object

TCId	TC12.2.3.7a
Test case	Requesting of protection information of an AXMEDIS Object
Initial conditions	Protection information exists

Configuration description	None
Description of functionality to be tested	The protection information of a given AXMEDIS object is retrieved from the AXMEDIS Database
Partners, people involved	
Validator(s) skill	High, Technical
Data set used	AXDS-PMS2
Steps	1 Protection information associated to an AXMEDIS Object is queried 2 PMS checks that operation is permitted 3 Check that protection information exists 4 Protection information is extracted from the AXMEDIS Database
Expected results	Protection information is returned
Variations	None
Issues	None
Test case Scope/Type	BlackBox

TCId	TC12.2.3.7b
Test case	Requesting of protection information of an AXMEDIS Object
Initial conditions	Protection information exists
Configuration description	None
Description of functionality to be tested	The protection information of a given AXMEDIS object is retrieved from the AXMEDIS Database
Partners, people involved	
Validator(s) skill	High, Technical
Data set used	AXDS-PMS2
Steps	1 Protection information associated to an AXMEDIS Object is queried 2 PMS checks that operation is permitted 3 Check that protection information exists 4 Protection information does not exist, an error is returned
Expected results	Protection information is not returned as it does not exist
Variations	None
Issues	None
Test case Scope/Type	BlackBox

12.2.4 AXMEDIS Reporting Service (EXITECH)

12.2.4.1 Object usage reporting

TCId	TC12.2.4.1
Test case	Object usage reporting
Initial conditions	AXMEDIS system is filled with a predefined set of action logs
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Object usage reporting
Partners, people	Distributors, Providers

involved	
Validator(s) skill	None
Data set used	None
Steps	1 Actor submits the object usage reporting request 2 The system returns results 3 The results are checked against the list of expected results
Expected results	AXCS correctly elaborates reports
Variations	• Empty report • Report with only one item • Report with a very huge number of items
Issues	None
Test case Scope/Type	BlackBox

12.2.5 Accounting Manager and Reporting Tool (EXITECH)

12.2.5.1 List of all operations performed on an object

TCId	TC12.2.5.1
Test case	List of all operations performed on an object
Initial conditions	AXMEDIS system is filled with a predefined set of action logs
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Operations performed reporting
Partners, people involved	Distributors, Content Providers, Collecting society
Validator(s) skill	None
Data set used	AXDS-ACCREP1
Steps	1 Actor submits a request for having all operations performed on an object 2 The system returns results 3 The results are checked against the list of expected results
Expected results	The full list of operation is reported
Variations	• Empty report • Report with only one item • Report with a very huge number of items • Report with only one type of performed operations • Report with all the possible types of operations performed
Issues	None
Test case Scope/Type	BlackBox

12.2.5.2 List of all operations performed by a user

TCId	TC12.2.5.2
Test case	List of all operations performed by a user
Initial conditions	AXMEDIS system is filled with a predefined set of action logs
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Operations performed reporting
Partners, people	Distributors

involved	
Validator(s) skill	None
Data set used	AXDS-ACCREP1
Steps	1 Actor submits a request for having all operations performed by an user 2 The system returns results 3 The results are checked against the list of expected results
Expected results	The full list of operation is reported
Variations	• Empty report • Report with only one item • Report with a very huge number of items • Report with only one type of performed operations • Report with all the possible types of operations performed
Issues	None
Test case Scope/Type	BlackBox

12.2.5.3 Usage statistics for an object

TCId	TC12.2.5.3
Test case	Usage statistics about
Initial conditions	AXMEDIS system is filled with a predefined set of action logs
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Usage statistics for an object
Partners, people involved	Distributors, Content Providers, Collecting society
Validator(s) skill	None
Data set used	AXDS-ACCREP1
Steps	1 Actor submits the query for obtaining statistic data for an object 2 The system returns results 3 The results are checked against the list of expected results
Expected results	The result set of statistic has
Variations	• Usage statistic about a distributor • Usage statistic about a content provider
Issues	None
Test case Scope/Type	BlackBox

12.2.5.4 Usage statistics about a distributor

See Variation 1 of test case 12.2.5.3.

12.2.5.5 Usage statistics about a provider

See Variation 2 of test case 12.2.5.3.

12.2.5.6 List objects for which an administrative account can be requested

TCId	TC12.2.5.6
Test case	List objects for which an administrative account can be requested
Initial conditions	AXMEDIS system is filled with a predefined set of action logs
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be	The list of objects that are in charge (from an administrative point of view) to the Actor is returned back

tested	
Partners, people involved	Distributors, Content Providers, Collecting society
Validator(s) skill	None
Data set used	AXDS-ACCREP1
Steps	1 Actor submits the query for obtaining the list of object for which AXMEDIS can provide to the Actor administrative information 2 The system returns a result set 3 The results are checked against the list of object that are really in charge to the Actor
Expected results	The list produced by the system and the list in the hand of the actor must match
Variations	None
Issues	None
Test case Scope/Type	BlackBox

12.2.5.7 List distributors

TCId	TC12.2.5.7
Test case	List AXMEDIS clients for a distributor/channel
Initial conditions	AXMEDIS system is filled with a predefined number of users that are identified as distributors
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	The list of distributors present in the AXMEDIS network is returned back to the Actor
Partners, people involved	Content creators, Distributors, End users, Content Providers.
Validator(s) skill	None
Data set used	None
Steps	1 Actor submits the query for obtaining the list of AXMEDIS distributors 2 The system returns a result set 3 The results are checked against the list of distributors of Initial Conditions
Expected results	The list produced by the system and the list in the Initial condition have to match
Variations	None
Issues	None
Test case Scope/Type	BlackBox

12.2.5.8 Report transactions over AXMEDIS objects

TCId	TC12.2.5.8
Test case	Generate reports about the transactions of a given content provider or aggregator.
Initial conditions	None.
Configuration description	AXMEDIS Reporting tool.
Description of functionality to be tested	Communicating with the AXMEDIS Certifier and Supervisor to get specific information related to the transactions performed on the objects of a given content provider or aggregator
Partners, people involved	Content creator, distributor
Validator(s) skill	None.
Data set used	None.
Steps	1 A user requests to consult the history of the transactions of a content provider.

	2 Obtain from the Certifier and Supervisor AXMEDIS database the specific information related to the transactions performed on the objects of a given content provider or aggregator. 3 List clients of the provider, with the history of their transactions, etc.
Expected results	Get information related to the transactions of a given content provider or aggregator
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

12.2.6 AXCS Synchroniser

TCId	TC12.2.6
Test case	AXCS Synchroniser
Initial conditions	AXCS on one channel is filled with predefined logs
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	AXCS Synchroniser is capable of giving log to SuperAXCS via Super AXCS Collector
Partners, people involved	Collecting society or other actors that interact with SuperAXCS, SuperAXCS, AXCS
Validator(s) skill	None
Data set used	None
Steps	1 SuperAXCS send to AXCS a request for log 2 AXCS Synchroniser send the logs collected in the AXCS 3 Logs returned are checked against those filled in the Initial Conditions
Expected results	AXCS correctly returns Action Logs
Variations	None
Issues	None
Test case Scope/Type	BlackBox

12.3 Protection Tool Engine (WP4.5: FUPF, EXITECH, WP5.6.5: FHGIGD)

Protection tool engine test cases regarding DRM support are defined in section 12.5.2, DRM support.

12.3.1 Protection Rules Editor (WP4.3.1: DSI, WP5.4.1: DSI)

12.3.1.1 Create a new protection rule

TCId	TC12.3.1.1
Test case	Create a new protection rule
Initial conditions	The AXMEDIS Protection Rules Editor is running
Configuration description	Protection Rules Editor is running
Description of functionality to be tested	• User can create and store protection rules • User can load, debug, and activitate protection rules • User can deactivate and delete protection rules
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved in the protection of objects (familiar with the GUI and with script languages).

Data set used	AXDS-PTE1
Steps	1 The User uses GUI to submit query for objects and is returned a list of results 2 The User selects part/all/none of results using the GUI 3 The User writes a new rule or updates an existing rule 4 The User writes or updates the schedule associated with the rule 5 The User stores the created rule into Protection Rules Database (local repository)
Expected results	Defined rules for composition defaulted as "inactive" and saved in repository
Variations	• The User defines a Selection by writing in the rule the scripting code (Protection Rule Language derived/similar to Composition Rule Language) for queries to be executed when the rule will be run • The User can define a rule or writing it as scripting code
Issues	None
Test case Scope/Type	GUI/Whitebox

12.3.1.2 Editing a protection rule

TCId	TC12.3.1.2
Test case	Editing a protection rule
Initial conditions	The AXMEDIS Protection Rules Editor is running
Configuration description	Protection Rules Editor
Description of functionality to be tested	Loading an existing protection rule
Partners, people involved	List of people involved in the test, partners, user-groups, other people needed
Validator skill	People involved in the protection of objects (familiar with the GUI and with script languages).
Data set used	AXDS-PTE2
Steps	1 The user browses the existing rules in the repository 2 The user selects a rule and this rule is loaded 3 The user edits the loaded rule
Expected results	New rules are created and saved in the repository
Variations	None
Issues	None
Test case Scope/Type	GUI/WhiteBox

12.3.1.3 Activating a protection rule

TCId	TC12.3.1.3
Test case	Activating a protection rule
Initial conditions	Set of complete rules defined and set as inactive
Configuration description	The User has completed a protection rule editing and wants to set the rule to be executed (as active)
Description of functionality to be tested	Activation of a protection rule
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved in the protection of objects (familiar with the GUI and with script languages).
Data set used	AXDS-PTE2
Steps	1 The User browses the Repository of Protection Rules

	2 The User selects a specific protection rule 3 The User activates the protection rule 4 A confirmation on the status of the activation is provided
Expected results	Protection rule(s) activated and submitted to the Active Protection Rules repository
Variations	• User can modify/cancel this action before the activation
Issues	None
Test case Scope/Type	GUI/Whitebox

12.3.1.4 Removing an active protection rule

TCId	TC12.3.1.4
Test case	Removing a protection rule
Initial conditions	Set of complete rules defined and set as active
Configuration description	The user opens the protection rule editor to remove an active rule
Description of functionality to be tested	Active protection rule removal
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved in the protection of objects (familiar with the GUI and with script languages).
Data set used	AXDS-PTE2
Steps	1 The User browses Active Rules in the Active Protection Rules Repository 2 The User selects the active rule to be disabled 3 The User deactivates the selected rule 4 The rule is removed from the Active Protection Rules Repository
Expected results	A selected protection rule is removed from the Active Protection Rules Repository
Variations	None
Issues	None
Test case Scope/Type	GUI/Whitebox

12.3.1.5 Debugging a protection rule

TCId	TC12.3.1.5
Test case	Debugging a protection rule
Initial conditions	Set of complete rules defined and set as active
Configuration description	The protection rule editor is running and a rule is displayed on the screen.
Description of functionality to be tested	Debugging of rules
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved in the protection of objects (familiar with the GUI and with script languages).
Data set used	AXDS-PTE2
Steps	1 The User loads a rule 2 The User chooses the debugging rule mode 3 The Rule Editor enters in the Debugging Mode 4 During the debugging mode the User: 4.1 Check the statements of rule step by step 4.2 Control the values of current variables

	5 Exit from the debugging mode
Expected results	The debugging has been successfully executed.
Variations	• The user has written a new rule and wants to debug it
Issues	Nome
Test case Scope/Type	GUI/Whitebox

12.3.2 Printing protection rules

TCId	TC12.3.2
Test case	Printing protection rules
Initial conditions	Set of rules defined/stored.
Configuration description	The protection rule editor is running and a protection rule is loaded.
Description of functionality to be tested	Protection rule is correctly printed under user request
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator(s) skill	People involved in the protection of objects (familiar with the GUI and with script languages).
Data set used	AXDS-PTE2
Steps	1 The user requests to print the protection rules. 2 Protection rules are correctly printed.
Expected results	Protection rules is stored
Variations	None
Issues	None
Test case Scope/Type	Blackbox/GUI

12.3.3 Protecting an AXMEDIS Object

TCId	TC12.3.3
Test case	Protecting an AXMEDIS object
Initial conditions	The protection tools engine (PTE) is ready for processing a protection rule. The PTE monitors the active protection rule and waits for an external event (e.g. initiated by the work flow or by the system clock).
Configuration description	Engine is active and accessing correct system time.
Description of functionality to be tested	Automatic protection of an AXMEDIS object (on time or on request).
Partners, people involved	Content Owner, Content Integrator, Content Distributor, AXMEDIS Workflow Manager
Validator(s) skill	User of the AXMEDIS framework (no end-users) who are familiar with the protection features, the protection tools (Protection Rule Editor, rule engine, and script language).
Data set used	AXDS-PTE2
Steps	1 External event triggers the protection of an object (system clock, workflow, user at the Protection tool Editor) 2 Rule is loaded and parameters are verified 3 Content is selected via the Query Support 4 Content is accessed via the AXOM

	5 The PAR are estimated on the basis of the licenses of the included resources and then the PAR is included in the AXInfo contained into the protected and non protected parts of the object 6 PMS Domain factory creates/adapts the license from the rules or the user input 7 Verification of PAR or License against given rights 8 PMS Domain factory creates required keys (e.g. for encryption or hash functions) 9 Creation of the protection information 10 Protection of the object (resulting in a new object or a new version of the object). Encryption support (see use case Encryption) is used via the AXOM 11 If the protection is successful and the protection information has been generated to protect this object, the protection information has to be stored (see use case Storage of security information) 12 Sending the license and the Protection information to the PMS. The PMS forwards the Protection Information to the AXCS 13 A new AXMEDIS object or a new version has been stored into AXMEDIS Database (AXMEDIS Objects repository) 14 End process notification with success
Expected results	The AXMEDIS object is protected and can be unprotected (only) by a legal user.
Variations	• An error occurs during the execution o Rule execution is interrupted o Process notification is failed • Content related variations o Different multi-media objects (e.g. number/types of internal objects) • Processing related variations o Using different selections o Using different protection rules (e.g. encryption or fingerprinting) o Using "incompatible" protection targets (e.g. DRM licensing or PAR) • Trigger related variations o initiated by time event (system clock) o initiated by work flow o initiated by user using the Protection Rule Editor
Issues	None
Test case Scope/Type	WhiteBox/BlackBox/Backend

12.3.4 Protection Information Editor and Viewer (FHGIGD)

12.3.4.1 Viewing Protection Information

TCId	TC12.3.4.1
Test case	Viewing protection information
Initial conditions	The AXMEDIS Editor is running. An AXMEDIS object is loaded.
Configuration description	AXCS is accessible.
Description of functionality to be tested	User can view protection information about the current object
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved in the protection of objects (familiar with the GUI).
Data set used	AXDS-PIE1
Steps	1 The user invokes an internal viewer/editor (see test case 4.2.1: invoking an

	internal viewer/editor) 2 The protection information is requested (see test case 12.2.3.7: Requesting of protection information of an AXMEDIS Object) 3 The protection information is shown to the user.
Expected results	Protection information is available in the viewer.
Variations	The object is not protected: no protection information is available.
Issues	None
Test case Scope/Type	GUI/Blackbox

12.3.4.2 Editing Protection information

TCId	TC12.3.4.2
Test case	Editing a protection information
Initial conditions	The AXMEDIS Editor is running. An AXMEDIS object is loaded.
Configuration description	Protection Tool Engine is running. PMS and AXCS are accessible.
Description of functionality to be tested	Modification of the protection of an AXMEDIS object.
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	People involved with the end-user have the appropriate familiarity with the GUI
Data set used	AXDS-PIE1
Steps	1 The user invokes an internal viewer/editor (see test case 4.2.1: invoking an internal viewer/editor) 2 The protection information is requested (see test case 12.2.3.7: Requesting of protection information of an AXMEDIS Object) 3 The protection information is shown to the user. 4 The user modifies the protection information/status. 5 The protection information is updated and AXMEDIS object is protected (see test case 12.2.3.6a/b: Association of protection information to an AXMEDIS Object) 6 The protection information is updated and shown to the user.
Expected results	Protection of AXMEDIS object is modified. Protection information is updated.
Variations	AXMEDIS object is initially unprotected.
Issues	None
Test case Scope/Type	GUI/BlackBox

12.4 Administrative Information Integrator (WP9.1: EXITECH)

12.4.1 Distributor/Collecting Society asks for administrative information

TCId	TC12.4.1
Test case	Distributor asks for administrative information
Initial conditions	AXMEDIS system is filled with a predefined set of action logs
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Usage statistics (Account-Logs) for an object
Partners, people involved	Distributors
Validator(s) skill	High for interacting with CMS to recover information
Data set used	AXDS-DB3

Steps	1 Distributor submits asks for administrative information to be put in the CMS 2 The system returns results in an XML format 3 The XML format is mapped to the CMS 4 The CMS is queried to verify if all the Action Logs in the Initial conditions are mapped back to the CMS
Expected results	The result set of statistic has
Variations	• Actor is the collecting society
Issues	None
Test case Scope/Type	BlackBox

12.5 Protection Manager Support / Server General

12.5.1 Protection Manager Support / Server

12.5.1.1 Authorisation and key management of a protected and governed AXMEDIS object in a connected environment

TCId	TC12.5.1.1a
Test case	Authorisation and key management of a protected and governed AXMEDIS object in a connected environment
Initial conditions	Verification is done by test case "Verification of AXMEDIS users using AXMEDIS tools on a Device" User is registered and has the appropriate licenses that give him permissions to consume the AXMEDIS object. Authorization support and AXCS are running.
Configuration description	None
Description of functionality to be tested	An end-user wants to consume a protected and governed AXMEDIS object for first time.
Partners, people involved	End-user
Validator(s) skill	None
Data set used	AXDS-PMS1
Steps	1 PMS requests the authorisation to the authorisation support. It sends an authorisation request that includes the user identification, the right, the resource, optionally the license(s) or its(their) identifier(s) and the status information. The authorisation server obtains the licenses associated to the user from the database of DRM licenses, if necessary, and performs the authorisation. 2 As the end-user has the appropriate license, the authorisation is positive. 3 PMS checks with AXCS if end-user has got the keys for decrypting object. 4 As it is the first time that the end-user tries to consume the object he does not have the keys. Then, the PMS obtains the secret information (decryption keys) needed to unprotect the object from the AXCS. This information is delivered to the user over a secure channel.
Expected results	The end-user consume the protected AXMEDIS object
Variations	None
Issues	None

TCId	TC12.5.1.1b
Test case	Denied authorisation of usage of a protected and governed AXMEDIS object in a connected environment

Initial conditions	
Configuration description	Verification is done by test case “Verification of AXMEDIS users using AXMEDIS tools on a Device” User is registered and does not have the appropriate licenses that give him permissions to consume the AXMEDIS object. Authorization server and AXCS are running.
Description of functionality to be tested	None
Partners, people involved	End-user
Validator(s) skill	None
Data set used	AXDS-PMS1
Steps	1 PMS request the authorisation to the authorisation server. It sends an authorisation request that includes the user identification, the right, the resource, optionally the license(s) or its(their) identifier(s) and the status information. 2 As the end-user has not been previously purchased the appropriate license, the authorisation server does not found the license(s) associated to this authorization request, and the result of the authorisation is negative. 3 The authorization server returns the reasons why the end-user has not been authorised and the user cannot consume the AXMEDIS object
Expected results	The end-user cannot consume the AXMEDIS object and he is informed of the reasons why he cannot consume it.
Variations	None
Issues	None

TCId	TC12.5.1.1c
Test case	Authorisation and key management of a protected and governed AXMEDIS object in a connected environment
Initial conditions	Verification is done by test case “Verification of AXMEDIS users using AXMEDIS tools on a Device” User is registered and has the appropriate licenses that give him permissions to consume the AXMEDIS object. Authorization server and AXCS are running.
Configuration description	None
Description of functionality to be tested	An end-user wants to consume a protected and governed AXMEDIS object and he has previously consumed this object.
Partners, people involved	End-user
Validator(s) skill	None
Data set used	AXDS-PMS1
Steps	1 PMS request the authorisation to the authorisation server. It sends an authorisation request that includes the user identification, the right, the resource, optionally the license(s) or its(their) identifier(s) and the status information. The authorisation server obtains the licenses associated to the user from the database of DRM licenses, if necessary, and performs the authorisation. 2 As the end-user has the appropriate license, the authorisation is positive. 3 PMS checks with AXCS if end-user has got the keys for decrypting object. 4 As the end-user has previously consumed this AXMEDIS object, he has the

	secret information (decryption keys) needed to unprotect the object.
Expected results	The end-user consume the protected AXMEDIS object
Variations	None
Issues	None

12.5.1.2 Authorisation and key management of a protected and governed AXMEDIS object in an unconnected environment

TCId	TC12.5.1.2a
Test case	Authorisation and key management of a protected and governed AXMEDIS object in an unconnected environment
Initial conditions	User is registered and has the appropriate licenses that give him permissions to consume the AXMEDIS object. Content Consumption status manager, License Interpreter, key manager and secure cache manager are running.
Configuration description	None
Description of functionality to be tested	An end-user wants to consume a protected and governed AXMEDIS object in an unconnected environment and he has the appropriate licenses to consume it.
Partners, people involved	End-user
Validator(s) skill	None
Data set used	AXDS-PMS1
Steps	1 An end-user tries to consume a protected and governed AXMEDIS object. 2 PMS obtains the status information from its Content Consumption status manager 3 PMS request the authorisation to the License Interpreter. It sends an authorisation request that includes the user identification, the right, the resource, the license(s) and the status information. The license interpreter performs the authorisation. 4 As the end-user has the appropriate license, the authorisation is positive. 5 PMS obtains the secret information (decryption keys) needed to unprotect the object from the Key Manager
Expected results	The end-user consume the protected AXMEDIS object
Variations	None
Issues	None

TCId	TC12.5.1.2b
Test case	Denied authorisation of usage of a protected and governed AXMEDIS object in an unconnected environment
Initial conditions	User is registered and does not have the appropriate licenses that give him permissions to consume the AXMEDIS object. Content Consumption status manager, License Interpreter, key manager and secure cache manager are running.
Configuration description	None
Description of functionality to be tested	An end-user wants to consume a protected and governed AXMEDIS object in an unconnected environment and he does not have the appropriate licenses to consume it.
Partners, people involved	End-user

Validator(s) skill	None
Data set used	AXDS-PMS1
Steps	1 An end-user tries to consume a protected and governed AXMEDIS object. 2 PMS obtains the status information from its Content Consumption status manager 3 PMS request the authorisation to the License Interpreter. It sends an authorisation request that includes the user identification, the right, the resource, the license(s) and the status information. The license interpreter performs the authorisation. 4 As the end-user does not have the appropriate license, the authorisation is negative 5 The License Interpreter returns the reasons why the end-user has not been authorised
Expected results	The end-user cannot consume the AXMEDIS object and he is informed of the reasons why he cannot consume it.
Variations	None
Issues	None

12.5.1.3 Protection of an AXMEDIS object

TCId	TC12.5.1.3
Test case	Protection of an AXMEDIS object
Initial conditions	AXMEDIS editor is opened with an AXMEDIS object loaded The user has logged in and his identity has been validated. PMS is running.
Configuration description	None
Description of functionality to be tested	A user wants to protect an AXMEDIS object
Partners, people involved	Content creator (in general, any user of AXMEDIS editor)
Validator(s) skill	High, technical
Data set used	AXDS-DRMSupport5
Steps	1 A user wants to protect an AXMEDIS object. 2 AXMEDIS editor makes use of Encryption support. Symmetric key for AXMEDIS object is encrypted with the public component of the creator's asymmetric key 3 If the encryption is correct, and the encryption information has been generated to protect this object, it has to be stored (see use case Storage of security information) 4 PMS sends the protection information (Symmetric key for encrypting / decrypting AXMEDIS objectAlgorithm used) to the AXCS, which stores it in the database.
Expected results	AXMEDIS object has been protected and the protection information has been stored.
Variations	None
Issues	None

12.5.1.4 Protection and association of licenses of/to an AXMEDIS object

TCId	TC12.5.1.4
Test case	Protection and association of licenses of/to an AXMEDIS object

Initial conditions	AXMEDIS editor is opened with an AXMEDIS object loaded The user has logged in and his identity has been validated. PMS is running.
Configuration description	None
Description of functionality to be tested	A user wants to protect an AXMEDIS object and include within it the appropriate license(s)
Partners, people involved	Content creator (in general, any user of AXMEDIS editor)
Validator(s) skill	High, technical
Data set used	AXDS-DRMSupport6
Steps	1 A user wants to protect an AXMEDIS object. 2 PMS checks if the user can protect the object. 3 A new key is generated for protecting the object by AXMEDIS supervisor. 4 AXOM makes use of Encryption support. 5 The encryption is correct PMS sends the protection information (Protection key for encrypting / decrypting AXMEDIS object, Algorithm used) to the AXCS, which stores it in the database. 6 License Generator generates the appropriate license(s) that the user wants to associate to this AXMEDIS object.
Expected results	AXMEDIS object has been protected and the protection information has been stored. The rights expressions have been generated.
Variations	None
Issues	None

12.5.1.5 Renewal of IPMP information after detection of a succeed attack (connected)

TCId	TC12.5.1.5
Test case	Renewal of IPMP information after detection of a succeed attack
Initial conditions	A succeed attack over the protection of an AXMEDIS object has been detected by AXCS AXMEDIS AXOM and PMS are running.
Configuration description	None
Description of functionality to be tested	A succeeded attack has been detected by the AXCS, then the IPMP information has been renewed and the AXMEDIS object re-protected.
Partners, people involved	
Validator(s) skill	High, technical
Data set used	AXDS-DRMSupport6
Steps	1 New key for protecting the object is generated by AXCS 2 The AXMEDIS object is re-protected with the new key and new algorithm 3 The AXMEDIS object is stored in the AXMEDIS object database. 4 PMS sends the protection information (Protection key for encrypting / decrypting AXMEDIS object, Algorithm used) to the AXCS, which stores it in the database 5 It is also indicated that the protection method has changed in order to inform the users accessing the protected AXMEDIS object
Expected results	AXMEDIS object has been properly stored together with its new protection information
Variations	None

Issues	None

12.5.2 DRM Support (WP4.5.1: FUPF)

12.5.2.1 License creation for new content

TCId	TC12.5.2.1a
Test case	License creation
Initial conditions	User must be registered
Configuration description	None
Description of functionality to be tested	An actor wants to create a license associated to some content
Partners, people involved	An actors that needs a license
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport1
Steps	1 User connects to the license generator 2 User enters the required data to create the license 3 User clicks the "Submit" button of the license creation tool 4 License generator creates the license based on the received information 5 License verificator (better verifier) validates the generated license and the result is positive 6 License manager inserts the license into the license database 7 The license generator returns to the actor the license ID, the license or both.
Expected results	A license is created and stored into the license database
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license storage) / BlackBox

TCId	TC12.5.2.1b
Test case	License creation
Initial conditions	User must be registered
Configuration description	None
Description of functionality to be tested	An actor wants to create a license associated to some content
Partners, people involved	An actor that needs a license
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport1
Steps	1 User connects to the license generator 2 User enters the required data to create the license 3 User clicks the "Submit" button of the license creation tool 4 License generator creates the license based on the received information 5 License verificator validates the generated license and the result is negative 6 The license generator returns to the actor a message explaining the reasons why the license couldn't be created
Expected results	An informative message explaining the reasons why the license couldn't be created

Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license storage) / BlackBox

12.5.2.2 License creation for cross-media content

TCId	TC12.5.2.2a
Test case	License creation for cross-media content
Initial conditions	User must be registered. At least are needed two licenses for create the result content.
Configuration description	None
Description of functionality to be tested	An actor requests a license to consume, create or distribute cross-media content.
Partners, people involved	An actors that needs a license
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 User connects to the License Generator 2 User enters the required data to create the cross-media license 3 User clicks the "Submit" button of the license creation tool 4 License Generator obtains from the License Manager all the licenses associated to the original AXMEDIS objects 5 License Generator derives a new license from the obtained licenses 6 License Verifier validates the new license 7 License Verifier verifies that the derived conditions are consistent 8 License manager inserts the license into the license database 9 The license generator returns to the actor the license ID, the license or both.
Expected results	A license is created and stored into the license database
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license storage) / BlackBox

TCId	TC12.5.2.2b
Test case	License creation for cross-media content
Initial conditions	User must be registered. At least are needed two licenses for create the result content.
Configuration description	None
Description of functionality to be tested	An actor requests a license to consume, create or distribute cross-media content.
Partners, people involved	An actors that needs a license
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 User connects to the License Generator 2 User enters the required data to create the cross-media license 3 User clicks the "Submit" button of the license creation tool 4 License Generator obtains from the License Manager all the licenses

	associated to the original AXMEDIS objects 5 License Generator derives a new license from the obtained licenses 6 License Verifier validates the new license 7 License Verifier verifies that the derived conditions are NOT consistent 8 The license generator returns to the actor a message explaining the reasons why the license couldn't be created.
Expected results	An informative message explaining the reasons why the license couldn't be created
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license storage) / BlackBox

12.5.2.3 License migration

TCId	TC12.5.2.3a
Test case	License migration
Initial conditions	User must be registered. At least one license is needed, and it contains the information about the user devices it can be transferred to (if no user devices specified, it is valid for all the devices of the user).
Configuration description	None
Description of functionality to be tested	A user wants to migrate a license from one device to another.
Partners, people involved	An actors that needs to migrate a license
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport4
Steps	1 The user has the license stored locally. 2 The user want to transfer the license to another device listed in the license. 3 The license is transferred unmodified to the destination device.
Expected results	The licenses is migrated to the destination device
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license storage) / BlackBox

TCId	TC12.5.2.3b
Test case	License migration
Initial conditions	User must be registered. At least one license is needed, and it contains the information about the user devices it can be transferred to (if no user devices specified, it is valid for all the devices of the user).
Configuration description	None
Description of functionality to be tested	A user wants to migrate a license from one device to another.
Partners, people involved	An actors that needs to migrate a license
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport4
Steps	1 The user wants to migrate a license that is stored in the the license DB. 2 The user want to transfer the license to another device listed in the license. 3 The license is not transferred to the destination device, because only is needed

	that the user identifies himself from the destination device..
Expected results	The licenses is not migrated
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license storage) / BlackBox

12.5.2.4 User authorisation

TCId	TC12.5.2.4a
Test case	Authorisation based on licenses
Initial conditions	User or tool must be registered
Configuration description	None
Description of functionality to be tested	An actor wants to perform an action over a resource. The authorisation process must check whether the action is granted by the license terms.
Partners, people involved	An actor that wants to perform an action over a resource
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport2
Steps	1 User sends to the authorisation server the required data to perform the authorisation, which includes the user identification, the right, the resource and the license(s) identifier(s). 2 The authorisation server obtains the licenses specified in the request from the DRM licenses database. 3 The authorisation server creates the authorisation request and story. 4 The authorisation server performs the authorisation and it is positive. 5 The authorisation server notifies that the user is authorised.
Expected results	The positive authorisation of the user request
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC12.5.2.4b
Test case	Authorisation based on licenses
Initial conditions	User or tool must be registered
Configuration description	None
Description of functionality to be tested	An actor wants to perform an action over a resource. The authorisation process must check whether the action is granted by the license terms.
Partners, people involved	An actor that wants to perform an action over a resource
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport2
Steps	1 User sends to the authorisation server the required data to perform the authorisation, which includes the user identification, the right, the resource and the license(s) identifier(s). 2 The authorisation server obtains the licenses specified in the request from the DRM licenses database. 3 The authorisation server creates the authorisation request and story. 4 The authorisation server performs the authorisation and it is negative. 5 The authorisation server notifies that the user is not authorised.

Expected results	The negative authorisation of the user request
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC12.5.2.4c
Test case	Authorisation based on licenses
Initial conditions	User or tool must be registered
Configuration description	None
Description of functionality to be tested	An actor wants to perform an action over a resource. The authorisation process must check whether the action is granted by the license terms.
Partners, people involved	An actor that wants to perform an action over a resource
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport2
Steps	1 User sends to the authorisation server the required data to perform the authorisation, which includes the user identification, the right, the resource and no license(s). 2 The authorisation server performs a search in the DRM licenses database and finds one or more licenses that match the entry data (user, right and resource). 3 The authorisation server creates the authorisation request and story. 4 The authorisation server performs the authorisation and it is positive. 5 The authorisation server notifies that the user is authorised.
Expected results	The positive authorisation of the user request
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC12.5.2.4d
Test case	Authorisation based on licenses
Initial conditions	User or tool must be registered
Configuration description	None
Description of functionality to be tested	An actor wants to perform an action over a resource. The authorisation process must check whether the action is granted by the license terms.
Partners, people involved	An actor that wants to perform an action over a resource
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport2
Steps	1 User sends to the authorisation server the required data to perform the authorisation, which includes the user identification, the right, the resource and no license(s). 2 The authorisation server performs a search in the DRM licenses database and does not find any licenses that match the entry data (user, right and resource). 3 The authorisation server notifies that the user is not authorised.
Expected results	The negative authorisation of the user request
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

12.5.2.5 Rights Expression Translator

TCId	TC12.5.2.5
Test case	Rights Expression Translator
Initial conditions	A license to be validated and the License Validator Tool available.
Configuration description	The system wants to translate a valid licenses (for instance, a mobile profile) from a REL into another with the Rights Expression Translator tool
Description of functionality to be tested	The destination license is valid
Partners, people involved	Integrator, Designer
Validator(s) skill	None
Data set used	AXDS-RET1
Steps	1 Select a source license 2 Execute the Rights Expression Translator 3 Check the destination license with the license validator.
Expected results	A new license translated to the corresponding REL
Variations	• If the source or destination license is not valid, the translation will not be possible and the system will show a message
Issues	None
Test case Scope/Type	Backend / BlackBox

12.6 Encryption/Decryption Support (FUPF)

12.6.1.1 Encryption

TCId	TC12.6.1.1a
Test case	Encryption of AXMEDIS object
Initial conditions	AXMEDIS editor is opened with an AXMEDIS object loaded
Configuration description	None
Description of functionality to be tested	Encryption of an AXMEDIS object using a symmetric key
Partners, people involved	Content creator (in general, any user of AXMEDIS editor)
Validator(s) skill	High, Technical
Data set used	AXDS-PMS1
Steps	1 An actor calls the “Save object” button on AXMEDIS Editor 2 It is checked if actor has permission to save the object 3 If actor has permission, the key for encrypting the object is recovered from its storage (license, etc) 4 The object is encrypted and can be saved
Expected results	AXMEDIS object is encrypted with the symmetric key
Variations	The key for encrypting the object does not exist and has to be created
Issues	None
Test case Scope/Type	BlackBox

TCId	TC12.6.1.1b
Test case	Encryption of AXMEDIS object

Initial conditions	AXMEDIS editor is opened with an AXMEDIS object loaded
Configuration description	None
Description of functionality to be tested	An actor tries to encrypt an AXMEDIS object using a symmetric key, but he has no permission
Partners, people involved	Content creator (in general, any user of AXMEDIS editor)
Validator(s) skill	High, Technical
Data set used	AXDS-PMS1
Steps	1 An actor calls the "Save object" button on AXMEDIS Editor 2 It is checked if actor has permission to save the object 3 Actor does not have permission. He is informed of the error
Expected results	None
Variations	None
Issues	None
Test case Scope/Type	Backend/BlackBox

12.6.1.2 Decryption

TCId	TC12.6.1.2a
Test case	Decryption of AXMEDIS object
Initial conditions	None
Configuration description	None
Description of functionality to be tested	An actor wants to open an AXMEDIS protected object
Partners, people involved	Any actor that can view protected AXMEDIS objects
Validator(s) skill	High, Technical
Data set used	AXDS-PMS1
Steps	1 An actor wants to open an AXMEDIS protected object (either by double click or inside AXMEDIS Editor / Viewer) 2 It is checked if actor has permission to open the object 3 If so, actor has permission, key for decrypting the object is recovered 4 The object is decrypted and AXMEDIS Editor or Viewer can show it
Expected results	AXMEDIS object is shown to the actor
Variations	None
Issues	None
Test case Scope/Type	Backend/BlackBox

TCId	TC12.6.1.2b
Test case	Decryption of AXMEDIS object
Initial conditions	None
Configuration description	None
Description of functionality to be tested	An actor tries to open a protected AXMEDIS object, but he has no permission
Partners, people involved	Actor that can use AXMEDIS viewer or AXMEDIS editor
Validator(s) skill	High, Technical

Data set used	AXDS-PMS1
Steps	1 An actor wants to open an AXMEDIS protected object (either by double click or inside AXMEDIS Editor / Viewer) 2 It is checked if actor has permission to open the object 3 The actor has no permission, the application informs of the error
Expected results	The AXMEDIS object is not shown to the user
Variations	None
Issues	None
Test case Scope/Type	Backend/BlackBox

12.6.1.3 Encryption of symmetric key

TCId	TC12.6.1.3
Test case	Encryption of AXMEDIS object symmetric key using public key techniques
Initial conditions	Symmetric and asymmetric keys exist
Configuration description	None
Description of functionality to be tested	A symmetric key for an AXMEDIS object is encrypted with asymmetric encrypting techniques for secure storage
Partners, people involved	Content creator
Validator skill	High, Technical
Data set used	AXDS-ENCDEC1
Steps	1 Symmetric key for AXMEDIS object is encrypted with the public component of the creator's asymmetric key
Expected results	Symmetric key is encrypted
Variations	None
Issues	None
Test case Scope/Type	Backend/BlackBox

12.6.1.4 Decryption of symmetric key

TCId	TC12.6.1.4
Test case	Decryption of AXMEDIS object symmetric key using public key techniques
Initial conditions	Symmetric and asymmetric keys exist
Configuration description	None
Description of functionality to be tested	A symmetric key for an AXMEDIS object is decrypted using asymmetric encrypting techniques to allow AXMEDIS object decryption
Partners, people involved	Actor that wants to use a protected AXMEDIS object
Validator skill	High, Technical
Data set used	AXDS-ENCDEC1
Steps	1 Symmetric key for AXMEDIS object is decrypted with the private component of the actor's asymmetric key
Expected results	Symmetric key is decrypted
Variations	None
Issues	None
Test case Scope/Type	Backend/BlackBox

12.6.1.5 Storage of security information

TCId	TC12.6.1.5
Test case	Storage of encryption/decryption information for a protected AXMEDIS object
Initial conditions	Symmetric, asymmetric keys and cryptographic information exist
Configuration description	None
Description of functionality to be tested	The encryption/decryption information of a given AXMEDIS object is stored
Partners, people involved	Content creator
Validator skill	High, Technical
Data set used	AXDS-ENCDEC1
Steps	1 Symmetric key is encrypted by means of public key techniques (see Encryption of symmetric key test case) 2 Cryptographic information regarding protected AXMEDIS object has to be stored 2.1 Symmetric key for encrypting / decrypting 2.2 AXMEDIS objectAlgorithm used
Expected results	Encryption/decryption information is stored
Variations	None
Issues	None
Test case Scope/Type	Backend/BlackBox

12.6.1.6 Retrieval of security information

TCId	TC12.6.1.6
Test case	Retrieval of encryption/decryption information for a protected AXMEDIS object
Initial conditions	Symmetric, asymmetric keys and cryptographic information exist
Configuration description	None
Description of functionality to be tested	The encryption/decryption information of a given AXMEDIS object is retrieved
Partners, people involved	Actor that wants to use a protected AXMEDIS object
Validator skill	High, Technical
Data set used	AXDS-ENCDEC1
Steps	1 Cryptographic information regarding protected AXMEDIS object has to be retrieved 1.1 Symmetric key for encrypting / decrypting AXMEDIS object 1.2 Algorithm used 2 Symmetric key is decrypted by means of public key techniques (see Decryption of symmetric key test case)
Expected results	Encryption/decryption information is retrieved
Variations	None
Issues	None
Test case Scope/Type	Backend/BlackBox

12.7 Protection tools integration test cases

12.7.1.1 Content consumption of an AXMEDIS Object (connected)

TCId	TC12.7.1.1a
Test case	Content consumption of an AXMEDIS Object (connected)
Initial conditions	A User wants to use an AXMEDIS Tool It is not his first use of the tool on the device
Configuration description	None
Description of functionality to be tested	The user is able to consume an object according to his licenses for the first time
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1, AXDS-PMS2, AXDS-DRMSupport7
Steps	3 An AXMEDIS User tries to perform an action on an AXMEDIS Object using an AXMEDIS Tool running on a device 4 The AXMEDIS Tool (AXOM) sends some needful information to PMS: AXDS-CertVer1 5 The DRM Support inside the PMS of reference contacts the AXCS sending him the received information 6 The Certification and verification (inside AXCS) checks that it is not the first use of the tool by the user on the device. 7 Retrieve the tool operation history fingerprint that is stored in the AXCS database (using AXCS Database interface) and check that it matches to the received one 8 The DRM Support (inside the PMS) verifies the DRM using the Authorization Support (inside PMS) 9 PMS requests the authorisation to the authorisation support. It sends an authorisation request that includes the user identification, the right, the resource, optionally the license(s) or its(their) identifier(s) and the status information. The authorisation server obtains the licenses associated to the user from the database of DRM licenses, if necessary, and performs the authorisation. 10 As the end-user has the appropriate license, the authorisation is positive. 11 Insert Action Log into AXCS reporting database 12 The PMS responds to AXMEDIS Tool with the key needed to perform the requested action on the AXMEDIS Object 13 The AXMEDIS Tool is now ready to perform the action requested by the AXMEDIS User
Expected results	The PMS is notified that the user, tool, device data and operation history have been verified
Variations	None
Issues	None
Test case Scope/Type	Backend

TCId	TC12.7.1.1b
Test case	Content consumption of an AXMEDIS Object (connected)
Initial conditions	A User wants to use an AXMEDIS Tool It is not the first use of the tool on the device
Configuration	None

description	
Description of functionality to be tested	The user is able to consume an object according to his licenses for the first time
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1, AXDS-PMS2, AXDS-DRMSupport7
Steps	1 An AXMEDIS User tries to perform an action on an AXMEDIS Object using an AXMEDIS Tool running on a device 2 The AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: AXDS-CertVer1 3 The DRM Support inside the PMS of reference contacts the AXCS sending him the received information 4 The Certification and verification (inside AXCS) checks that it is not the first use of the tool by the user on the device. 5 Retrieve the tool operation history fingerprint that is stored in the AXCS database (using AXCS Database interface) 6 Check that the retrieved FP does not match to the received one 7 A new fingerprint is computed, derived from the previous one (stored in the database) and the operation history (sent by the user). 8 The new fingerprint is compared to the fingerprint provided by the user and it matches. 9 The DRM Support (inside the PMS) verifies the DRM using the Authorization Support (inside PMS) 10 PMS requests the authorisation to the authorisation support. It sends an authorisation request that includes the user identification, the right, the resource, optionally the license(s) or its(their) identifier(s) and the status information. The authorisation server obtains the licenses associated to the user from the database of DRM licenses, if necessary, and performs the authorisation. 11 As the end-user has the appropriate license, the authorisation is positive. 12 Insert Action Log into AXCS reporting database 13 The DRM Support responds to AXMEDIS Tool with a Grant signal and the key needed to use the perform the requested action on the AXMEDIS Object 14 The AXMEDIS Tool is now ready to perform the action requested by the AXMEDIS User
Expected results	The PMS is notified that the user data, tool data and operation history have been verified
Variations	None
Issues	None
Test case Scope/Type	Backend

TCId	TC12.7.1.1c
Test case	Content consumption of an AXMEDIS Object (connected)
Initial conditions	A User wants to use an AXMEDIS Tool It's not his first use of the tool on the device The tool is connected or reconnected
Configuration description	None
Description of functionality to be tested	Negative verification of user data, tool data and tool operation history consistency

Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1, AXDS-PMS2, AXDS-DRMSupport7
Steps	1 An AXMEDIS User tries to perform an action on an AXMEDIS Object using an AXMEDIS Tool running on a device 2 The AXMEDIS Tool (AXOM) sends some needful information to PMS: AXDS-CertVer1 3 The DRM Support inside the PMS of reference contacts the AXCS sending him the received information 4 The Certification and verification (inside AXCS) checks that it is not the first use of the tool by the user on the device. 5 Retrieve the tool operation history fingerprint that is stored in the AXCS database (using AXCS Database interface) 6 Check that the retrieved FP does not match to the received one 7 A new fingerprint is computed, derived from the previous one (stored in the database) and the operation history (sent by the user). 8 The new fingerprint is compared to the fingerprint provided by the user and it does not match
Expected results	The PMS is notified that the tool operation history is not consistent. The PMS must then send an Event Report to the AXMEDIS Supervisor so as to block the user and/or tool
Variations	None
Issues	None
Test case Scope/Type	Backend

TCId	TC12.7.1.1d
Test case	Content consumption of an AXMEDIS Object (connected)
Initial conditions	A User wants to use an AXMEDIS Tool It is not his first use of the tool on the device
Configuration description	None
Description of functionality to be tested	Negative authorisation of the user
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1, AXDS-PMS2, AXDS-DRMSupport7
Steps	1 An AXMEDIS User tries to perform an action on an AXMEDIS Object using an AXMEDIS Tool running on a device 2 The AXMEDIS Tool (AXOM) sends some needful information to PMS: AXDS-CertVer1 3 The DRM Support inside the PMS of reference contacts the AXCS sending him the received information 4 The Certification and verification (inside AXCS) checks that it is not the first use of the tool by the user on the device. 5 Retrieve the tool operation history fingerprint that is stored in the AXCS database (using AXCS Database interface) and check that it matches to the received one 6 The DRM Support (inside the PMS) verifies the DRM using the Authorization Support (inside PMS) 7 PMS requests the authorisation to the authorisation support. It sends an

	authorisation request that includes the user identification, the right, the resource, optionally the license(s) or its(their) identifier(s) and the status information. The authorisation server obtains the licenses associated to the user from the database of DRM licenses, if necessary, and performs the authorisation. 8 Some license is missing, has expired or the conditions are not accomplished. The authorisation is negative. 9 Insert Action Log into AXCS reporting database 10 The PMS responds to AXMEDIS Tool with the negative response 11 The AXMEDIS Tool receives the negative authorisation
Expected results	The PMS is notified that the user has not been authorised
Variations	None
Issues	None
Test case Scope/Type	Backend

12.7.1.2 Content consumption of an AXMEDIS Object (unconnected)

TCId	TC12.7.1.2
Test case	Content consumption of an AXMEDIS Object (unconnected)
Initial conditions	A User wants to use an AXMEDIS Tool It is not his first use of the tool on the device It is not the first use of the AXMEDIS object
Configuration description	None
Description of functionality to be tested	The user is able to consume an object according to his licenses for the first time in an unconnected environment
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1, AXDS-PMS2, AXDS-DRMSupport7
Steps	1 An AXMEDIS User tries to perform an action on an AXMEDIS Object using an AXMEDIS Tool running on a device 2 The AXMEDIS Tool (AXOM) sends some needful information to PMS: AXDS-CertVer1 3 The DRM Support (inside the PMS) verifies the DRM using the Authorization Support (inside PMS) 4 PMS requests the authorisation to the authorisation support. It sends an authorisation request that includes the user identification, the right, the resource, optionally the license(s) or its(their) identifier(s) and the status information. The authorisation server obtains the licenses associated to the user from the database of DRM licenses, if necessary, and performs the authorisation. 5 As the end-user has the appropriate license, the authorisation is positive. 6 Successful content consumption is stored in the secure cache manager. 7 The PMS responds to AXMEDIS Tool with the key needed to perform the requested action on the AXMEDIS Object 8 The AXMEDIS Tool is now ready to perform the action requested by the AXMEDIS User
Expected results	The PMS is notified that the user, tool, device data and operation history have been verified
Variations	None
Issues	None

Test case Scope/Type	Backend

12.7.1.3 Content Consumption inside a Domain

TCId	TC12.7.1.3a
Test case	Content consumption inside a domain
Initial conditions	A User wants to perform an action on an AXMEDIS Object using an AXMEDIS Too inside a Domain It's not his first use of the Tool on the device The Tool is connected or reconnected
Configuration description	None
Description of functionality to be tested	Positive verification of user data, tool data and tool operation history consistency
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	User ID, Tool ID, Tool fingerprint, Tool operation history, Tool operation history fingerprint
Steps	1 An AXMEDIS User tries to perform an action on an AXMEDIS Object using an AXMEDIS Tool running on a device 2 The AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: where UID, TID, device FP, tool operation history and tool operation history FP, email and Domain 3 The DRM Support inside the PMS of reference contacts the Domain Manager sending it the received information 4 The Domain Manager performs the verification of TID Domain Consistency and it is positive 5 The DRM Support receives the positive response from the Domain Manager and sends the information to AXCS (AXMEDIS Certification and Verification) 6 The AXMEDIS Certification and Verification continues with the steps explained in the "Verification of AXMEDIS users using AXMEDIS tools during content consumption" test cases
Expected results	The PMS is notified that the user, tool, device data and operation history have been verified in the Domain
Variations	• Same as in "Verification of AXMEDIS users using AXMEDIS tools during content consumption" test cases.
Issues	None
Test case Scope/Type	Backend

TCId	TC12.7.1.3b
Test case	Content consumption inside a domain
Initial conditions	A User wants to perform an action on an AXMEDIS Object using an AXMEDIS Too inside a Domain It's not his first use of the Tool on the device The Tool is connected or reconnected
Configuration description	None
Description of functionality to be	Negative verification of user data, tool data and tool operation history consistency

tested	
Partners, people involved	AXMEDIS User
Validator(s) skill	High, technical
Data set used	User ID, Tool ID, Tool fingerprint, Tool operation history, Tool operation history fingerprint
Steps	1 An AXMEDIS User tries to perform an action on an AXMEDIS Object using an AXMEDIS Tool running on a device 2 The AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: where UID, TID, device FP, tool operation history and tool operation history FP, email and Domain 3 The DRM Support inside the PMS of reference contacts the Domain Manager sending it the received information 4 The Domain Manager performs the verification of TID Domain Consistency and it is negative 5 The DRM Support receives the negative response from the Domain Manager 6 Insert Action Log into AXCS reporting database 7 The AXMEDIS Tool is not allowed to perform the action requested by the AXMEDIS User
Expected results	The AXMEDIS Tool is notified that it is not allowed to perform the action requested by the AXMEDIS User in the Domain
Variations	• If the AXMEDIS Certification and Verification doesn’t authenticate the Device, the device must be deactivated immediately sending it a “Deactivation Signal” and marking it as “blocked”
Issues	None
Test case Scope/Type	Backend

13 AXMEDIS Player (WP4.1, WP4.6: EPFL, SEJER, DSI)

13.1 AXMEDIS Player on PC, Tablet PC (EPFL, SEJER, DSI)

13.1.1 Content Recording for Playtime Shift

TCId	TC13.1.1
Test case	Content Recording for Playtime Shift
Initial conditions	None
Configuration description	An AXMEDIS Player must be available supporting audiovisual playback and connected to a storage device, such as hard disk or other. The Player must be properly installed and configured for download.
Description of functionality to be tested	A user stores audiovisual content in a backup support to possibly play this content with a time shift from the moment when it is downloaded.
Partners, people involved	Actor that can use the AXMEDIS player
Validator(s) skill	Medium, Experienced User
Data set used	AXDS-PlMulti, AXDS-PlAu
Steps	1 The user selects from a content distributor catalogue an AXMEDIS Object to download containing an audiovisual file for which viewing and time shifting (recording) license can be available 2 The client terminal, if license terms for the AXMEDIS Object allow this, activate the Backup/Record Function 3 The user specifies the “title” with which the AXMEDIS content has to be recorded 4 The user executes the Backup/Record Function and download begins. 5 At a later time, after download end, the Player is started in playback mode to play a selected recorded “title”
Expected results	The AXMEDIS object is correctly played at a later time than download time.
Variations	• A simpler case can be tested with only audio content
Issues	None
Test case Scope/Type	GUI / WhiteBox

13.1.2 Fast-forward of Content in Internal Players/Viewers

TCId	TC13.1.2a
Test case	Fast-forward of Content in Internal Players/Viewers
Initial conditions	The AXMEDIS Player is active and an audiovisual sequence (in an AXMEDIS Object) is open.
Configuration description	An AXMEDIS Player must be available supporting audiovisual playback. The Player must be properly installed and configured.
Description of functionality to be tested	The User wants to play a digital resource faster for a quick preview or for fast access to a later sequence.
Partners, people involved	Actor that can use the AXMEDIS player
Validator(s) skill	Low, End User
Data set used	AXDS-PlMulti, AXDS-PlAu
Steps	1 The User selects the Play command 2 The User selects the fast-forward command, each time speed in increased more 3 The activated viewer/player inside the AXMEDIS Player starts skipping frames at appropriate rate to speed-up playback

	4 When the User releases the fast-forward command (or select play, according to the player), the viewer/player returns to normal playback mode
Expected results	The AXMEDIS object is correctly played at faster speed. Audio is only perceivable until a speed factor of 2.
Variations	• A simpler case can be tested with only audio content
Issues	None
Test case Scope/Type	GUI/ WhiteBox

TCId	TC13.1.2b
Test case	Fast-backward of Content in Internal Players/Viewers
Initial conditions	The AXMEDIS Player is active and an audiovisual sequence (in an AXMEDIS Object) is open.
Configuration description	An AXMEDIS Player must be available supporting audiovisual playback. The Player must be properly installed and configured.
Description of functionality to be tested	The User wants to play a digital resource faster backward for fast access to a previous sequence.
Partners, people involved	Actor that can use the AXMEDIS player
Validator(s) skill	Low, End User
Data set used	AXDS-PlMulti, AXDS-PlAu
Steps	1 The User selects the Play command 2 The User selects the fast-backward command, each time speed in increased more 3 The activated viewer/player inside the AXMEDIS Player starts playing backwards skipping frames at appropriate rate to speed-up playback. 4 When the User releases the fast-backward command (or select play, according to the player), the viewer/player returns to normal playback mode.
Expected results	The AXMEDIS Object is played back at fast speed. Audio is not perceivable.
Variations	• A simpler case can be tested with only audio content
Issues	This case is not as simple as the fast forward, especially with coded material. In fact, many formats are based on forward prediction, so backward playback may be simple only at predefined speeds (only Intra frames, etc.)
Test case Scope/Type	GUI/ WhiteBox

13.1.3 Local adaptation of Content in Internal Players/Viewers

TCId	TC13.1.3
Test case	Local adaptation of Content in Internal Players/Viewers
Initial conditions	The AXMEDIS Player is active and several audiovisual sequences (in AXMEDIS Objects) are open.
Configuration description	An AXMEDIS Player must be available supporting audiovisual playback with appropriate degradation techniques. The Player must be properly installed and configured.
Description of functionality to be tested	The Actor wants to play more digital items possibly requiring a system resource management in real-time
Partners, people involved	Actor that can use the AXMEDIS player as a skilled tester
Validator(s) skill	High, Technical
Data set used	AXDS-PlMulti, AXDS-PlAu
Steps	1 The User select the Play command

	2 The system activates the proper internal player/viewer 3 The User select again the Play command for a second Object 4 The activated viewers/players inside the AXMEDIS Player receive monitoring about resource availability: they possibly start skipping frames at appropriate rate to maintain system stability 5 New objects playbacks are possibly started until resource saturation is reached 6 When the User stops one of the object playbacks, the viewer/player returns to normal playback mode
Expected results	The AXMEDIS objects are played at reduced quality but the overall stability is not broken.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

13.1.4 Annotate for personal use

TCId	TC13.1.4a
Test case	Annotate for personal use
Initial conditions	None
Configuration description	An AXMEDIS Player must be available supporting audiovisual playback and textual annotations. The Player must be properly installed and configured for download.
Description of functionality to be tested	The user adds a personal text annotation to a video.
Partners, people involved	Actor that can use the AXMEDIS player
Validator(s) skill	Medium, Experienced User
Data set used	AXDS-PlVid, AXDS-PlAu
Steps	1 The Actor opens the AXMEDIS Object that contains a video using the Player, and starts playing it 2 The Actor selects the Annotations button of the Player 3 A simple text editor is opened 4 The user writes a commentary during a sequence 5 The user saves the commentary. The Player takes care of associating the written text to the content references
Expected results	The commentary is saved and associated to the content.
Variations	• Other kinds of content can be used, audio content for a simpler case.
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC13.1.4b
Test case	Annotate for personal use
Initial conditions	None
Configuration description	An AXMEDIS Player must be available supporting audiovisual playback and textual annotations. The Player must be properly installed and configured for download.
Description of functionality to be tested	The user views a personal text annotation to a video previously stored.
Partners, people involved	Actor that can use the AXMEDIS player
Validator(s) skill	Medium, Experienced User

Data set used	AXDS-PlMulti, AXDS-PlAu
Steps	1 The Actor opens the AXMEDIS Object that contains a video using the Player and starts playing it 2 The Actor selects the showAnnotation command of the Player 3 When the reference is reached with Annotated content, a text window opens in a corner showing the annotation
Expected results	The annotation is correctly displayed
Variations	• Other kinds of content can be used, audio content for a simpler case.
Issues	None
Test case Scope/Type	GUI / BlackBox

13.1.5 Local User Profiles

TCId	TC13.1.5
Test case	Local User Profiles
Initial conditions	The final user is the administrator of the platform where the player is installed
Configuration description	An AXMEDIS Player must be available supporting audiovisual playback and different user profiles (with different privileges). The Player must be properly installed and configured for download.
Description of functionality to be tested	The user sets up a Player profile configuration
Partners, people involved	Actor that can use the AXMEDIS Player
Validator(s) skill	Medium, Experienced User
Data set used	None
Steps	1 The user clicks on the Profile configuration button of the Player 2 The user selects that the Player cannot be used from 12 pm to 7 am for all normal users 3 The user saves the profile
Expected results	The non-administrator users will not be able to play any content from 12 pm to 7 am
Variations	• Other properties of the Player can be stored in the profile like: default volume or types of content allowed
Issues	None
Test case Scope/Type	GUI / BlackBox

13.1.6 History of the last played contents

TCId	TC13.1.6
Test case	History of the last played contents
Initial conditions	AXMEDIS Player is active with an AXMEDIS Object open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured.
Description of functionality to be tested	The user plays an AXMEDIS Object and this object is recorded in the list of last played objects
Partners, people involved	Actor that can use the AXMEDIS Player
Validator(s) skill	Low, End User
Data set used	AXDS-PlMulti, AXDS-PlVid, AXDS-PlAu

Steps	1 The Actor plays the content, the Player records the name and location of the played content 2 The Actor closes the Player 3 The Actor opens again the Player and selects the recent Playlist
Expected results	The list of the last played contents contains in the first place the previously played Object. Clicking on any item of this lists makes the Player to open the selected object
Variations	• A list may be available for most played objects
Issues	None
Test case Scope/Type	GUI / BlackBox

14 AXMEDIS for Distribution via Internet (WP4.6, WP9.4: TISCALI)

14.1 Back Office Management

14.1.1 Creating a New Mediaclub

TCId	TC14.1.1
Test case	Creating a new Mediaclub setup
Initial conditions	The system is up and running and fully configured; Actors have network access to the management interface (web). All techinical info needed to configure the mediaclub are provided by the Content distributor
Configuration description	MediaClub server is up and running on a standard PC based server architecture A PC with a browser installation connected to internet.
Description of functionality to be tested	Set up a new MediaClub in the cms
Partners, people involved	System Manager (sys mng)
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	None
Steps	1 (sys mng) Log in to the system and add a new project (name and description) 2 (sys mng) Configure the MediaClub website publishing targets and publishing modes (static pages, dynamic, etc) 3 (sys mns) Create the projects content repository witch will contains the contents types definition and all contents that will be included in the project 4 (sys mng) Create the project media repository witch contains binaries content as images, video stream, audio stream, etc 5 (sys mng) Define feed import rules 6 (sys mng) Define referred publishing rules, if needed 7 (sys mng) Configure the project administrator 8 (sys mng) Save configuration
Expected results	A mediaclub project is created
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.1.2 Mediaclub Setup

TCId	TC14.1.3
Test case	Mediaclub set up
Initial conditions	The system is up and running and fully configured; actors have network access to the management interface (web)
Configuration description	A PC with a browser installation connected to internet.
Description of functionality to be tested	Define all mediaclub feactures in the cms
Partners, people involved	Project Manager (prj mng)
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	AXDS-MCProject

Steps	1 (prj mng) Log in to the system and load the project settings form (name and description) 2 (prj mng) Configure the MediaClub website sections 3 (prj mng) Create the projects content types (xsl schema; xsl target and taget layout) 4 (prj mng) Create content categories and media categories three
Expected results	The MediaClub front-end GUI scheleton is created
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.1.3 Mediaclub Accounts and Permission Management

TCId	TC14.1.3
Test case	Mediaclub accounts and permissions
Initial conditions	The system is up and running and fully configured; actors have network access to the management interface (web) permissions
Configuration description	A PC with a browser installation connected to internet.
Description of functionality to be tested	Manage a MediaClub accounts and their permissions
Partners, people involved	Project Manager (prj mng)
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	AXDS-MCProducer
Steps	1 (prj mng) Log in to the system and load the project account management form (n 2 (prj mng) Create a new project account defining personal details, user id, password 3 (prj mng) Define account permission (Editor, publish authorizer, project manager …..)
Expected results	Users are able to access the back-office and performe editorial, publishing and project tasks
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.1.4 Mediaclub Project Uploading and publishing contents

TCId	TC14.1.4
Test case	Mediaclub publishing
Initial conditions	The system is up and running and fully configured; actors have network access to the management interface (web)
Configuration description	A PC with a browser installation connected to internet.
Description of functionality to be tested	Upload contents in the cms and publish them in the related mediaclub site
Partners, people involved	Project Manager (prj mng)
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some

	steps suggested by the technical support
Data set used	AXDS-MCObject
Steps	1 (editor) Log in to the system and loads the select new content action 2 (editor) Choose the content type and define a content name 3 (editor) Fill all fields required from the defined content type 4 (editor) Save content and choose one or more publishing targets 5 (editor) Submit content to authorization for publishing 6 (publisher) Authorize or reject the publish request
Expected results	Content is regularly updated in the system and MediaClub front-end
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.1.5 Mediaclub Project Acquiring AXMEDIS content +

TCId	TC14.1.5
Test case	Mediaclub and AXMEDIS content
Initial conditions	The system is up and running and fully configured; actors have network access to the management interface (web)
Configuration description	A PC with a browser installation connected to internet.
Description of functionality to be tested	Set up a new MediaClub in the cms
Partners, people involved	Project Manager (prj mng)
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	AXDS-MCObject
Steps	1 (prj mng) Search a specific content on a AXMEDIS p2p network 2 (prj mng) Select AXMEDIS content and view all meta data infos 3 (prj mng) Acquire license (if needed) and refer the object in the MediaClub contents
Expected results	Content is regularly inserted in the system and MediaClub front-end
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.1.6 Mediaclub Project define payment gateway entry

TCId	TC14.1.6
Test case	Mediaclub payments system setup
Initial conditions	The system is up and running and fully configured; actors have network access to the management interface (web)
Configuration description	A PC with a browser installation connected to internet.
Description of functionality to be tested	Enable the payment gateway to provide payment service to the specific mediaclub
Partners, people involved	System Manager (sys mng)
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support

Data set used	AXDS-MCPayShop, AXDS-MCPayMethod
Steps	1 (sys mng) Log in to the system and go in to payment management section 2 (sys mng) Configure a new mediaclub shop in the payment gateway giving (name, description, other details) 3 (sys mng) Define payment methods available for the mediaclub 4 (sys mng) Configure the shop administrator 5 (sys mng) Save configuration
Expected results	Payment methods are regularly associated to correct shop
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.1.7 Mediaclub Shop payment Management

TCId	TC14.1.7
Test case	Mediaclub shop payments configuration
Initial conditions	The system is up and running and fully configured; actors have network access to the management interface (web)
Configuration description	A PC with a browser installation connected to internet.
Description of functionality to be tested	Configure a mediaclub shop in the payment gateway
Partners, people involved	Shop Manager (shop mng)
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	AXDS-MCPayShop, AXDS-MCPayMethod
Steps	1 (shop mng) Log in to the system and go in to payment management section 2 (sys mng) Configure mediaclub call back URL for success, failure and error transaction 3 (shop mng) Choose payment methods available for the mediaclub 4 (sys mng) Upload schema and graphical components needed to build the payments transaction pages that will be shown to the end user
Expected results	Chosen payment methods are regularly functioning and consumers can operate transactions on the MediaClub
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.1.8 Mediaclub Shop Management refound a transaction

TCId	TC14.1.8
Test case	Mediaclub refound management
Initial conditions	Customer have provided transaction details and is proven that he hasn't had the digital goods
Configuration description	A PC with a browser installation connected to internet.
Description of functionality to be tested	Refound a payment transaction in a MediaClub shop
Partners, people	Shop Manager (shop mng)

involved	
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	AXDS-MCTransaction
Steps	1 (shop mng) Search the transaction id and or the user id in the transaction list 2 (shop mng) Load the transaction details and check if everything is ok 3 (shop mng) Starts transaction refound process
Expected results	Transaction is regularly refunded
Variations	None
Issues	None
Test case Scope/Type	GUI, Blackend / BlackBox

14.2 End User Client configuration

14.2.1 User Software Installation

TCId	TC14.2.1
Test case	User Software Installation
Initial conditions	The user has completed software installation steps.
Configuration description	A PC with an AXMEDIS end user client installation connected to internet.
Description of functionality to be tested	The user installs the AXMEDIS Client Application The user is ready to use the MediaClub service and access the published Content. (Access can be restricted only to some components). Well functioning of the AXMEDIS Client after installation: • The AXMEDIS Client is able to run correctly; • The AXMEDIS Client does not create any conflicts with the previously installed applications; The AXMEDIS Client is able to stop its execution.
Partners, people involved	End-user
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	AXDS-Video, AXDS-PCClientIcon
Steps	1 The User launches the MediaClub using the desktop shortcut; 2 The User checks that the MediaClub service is working correctly: 2.1 He opens the GUI and the catalogue listing is displayed and correctly updated; 2.2 The AXMEDIS client plug-in is launched correctly within the Internet browser; 3 The User stops the AXMEDIS Client Application.
Expected results	The user can connect to the MediaClub service, but not to restricted sections. The AXMEDIS Client plug-in (limited to basic functionalities) works fine: • The AXMEDIS Client starts/stops and behaves correctly; • All previously installed application still work fine during AXMEDIS Client is running
Variations	• The AXMEDIS Client plug-in can automatically be launched at system start up.
Issues	In case of problems, the User should contact the technical support for troubleshooting.
Test case Scope/Type	GUI / BlackBox

14.2.2 User Registration

TCId	TC14.2.2
Test case	User Registration
Initial conditions	The MediaClub service is working and accessible by the user. The AXMEDIS Client is well installed and it works fine. The User has followed step by step the registration wizard. The backend shall be up and running and able to treat all incoming registration request from the Users.
Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the server for registering to the MediaClub.
Description of functionality to be tested	The user register himself in order to access the MediaClub service. The User has finished his registration procedure and now is able to completely access the MediaClub Service.
Partners, people involved	End User, MediaClub Backend
Validator(s) skill	Users should be familiar with computers and web based services.
Data set used	AXDS-MCTestUser
Steps	1 The User verifies that the Registration finishes with no errors. 2 The user verifies reception of registration confirmation email 3 The MediaClub Backend ha successfully received the user email confirmation 4 The User verifies the correct reception of all Authorizations associated with the test login. 5 The User has regular access to the restricted sections of the MediaClub service.
Expected results	The User has successfully finished the Client Registration; he has completely received all related Authorizations; The User can entirely access to the complete MediaClub service.
Variations	None
Issues	None
Test case Scope/Type	GUI

14.3 Catalogue Browsing

14.3.1 Browsing content

TCId	TC14.3.1
Test case	Catalogue Listing
Initial conditions	The user has an Internet Connection. The User has registered to the MediaClub.
Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The user accesses the catalogue listing. The User browses the content listed in order to find some interesting contents.
Partners, people involved	Content consumers
Validator(s) skill	User should be familiar with Internet Browsing.
Data set used	AXDS-Video
Steps	1 The user reaches MediaClub catalogue listing 2 The user accesses content according to all criteria available (type, author, content producer, production date) 3 The user performs keyword or free-text based searches

	4 The user accesses individual content pages 5 The user reads all available information (contained in the AXMEDIS Info) associated to the AXMEDIS Object 6 The user selects content and is prompted to chose wether to pre-download or direcly purchase 7 user choses one of the above options and is directed to the appropriate post-condition
Expected results	The user can browse the content on the web page, search and access content.
Variations	• Use different criteria to search and browse the content.
Issues	None
Test case Scope/Type	GUI / BlackBox

14.3.2 User Page

TCId	TC14.3.2
Test case	User Page
Initial conditions	The user has an Internet Connection. The User has registered to the MediaClub.
Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The User accesses the MediaClub user page The User provides his/her preferences about AXMEDIS content The User choose what should and what should not be included in his/her preferences profile The User saves his/her profile The user views transaction and license information
Partners, people involved	AXMEDIS End User, the MediaClub FE
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-MCTransaction, AXDS-MCTestUser, AXDS-MCTestUser
Steps	1 The User opens the User Page 2 The User verifies that initial manual User preferences set up finishes with no errors 3 The User verifies the correct effects of his preferences 4 The User verifies that he/she is able to manually change his/her preferences 5 The User verifies the possibility to avoid sending private information to the server 6 The User verifies correct reporting of transactions and licenses 7 The MediaClub FE verifies the correct reception of the User Profiles 8 The MediaClub FE verifies the correct storing of the User Profiles
Expected results	The User can access his/her User Page and manually correct it. The User can choose if remove private information from his/her User Page. The MediaClub FE successfully receives the User Profiles from the Client Application. The MediaClub FE successfully stores the User Profiles
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.3.3 Available resources listing

TCId	TC14.3.3
Test case	Catalogue Browsing
Initial conditions	AXMEDIS Player is accessible within a VLE (Virtual Learning Environment) that has an agreement with an AXMEDIS distributor (*i.e.* distribution channel)

	The Actor is registered as a valid VLE system Agreement between the VLE Provider provides valid licenses for some product for the Actor
Configuration description	PC within the domain covered by the VLE VLE client software installed and properly configured Distribution Portal actually has some Object for the User Actor correctly registered in the VLE system and AXCS etc. properly configured to interact with VLE authentication system.
Description of functionality to be tested	Display of the list of resources available to a user by the distribution portal
Partners, people involved	Actor is the final User of the software
Validator(s) skill	Low, End User
Data set used	AXDS-Editor1
Steps	1 The Actor logs into the VLE 2 The Actor launch the AXMEDIS Player available within the VLE
Expected results	The actor's Player displays the distributor portal, with a list of resources available to the Actor The actor is able to select a view one of these resources
Variations	• No Object available to this User o The Portals displays a Notice explaining to the User that he has to consult is teacher or a specific contact person to check the situation
Issues	The test case may vary depending on the way the AXMEDIS Framework manages the corresponding Use Case.
Test case Scope/Type	Backend/UnitTest

14.4 Catalogue Content Purchase

14.4.1 Content Fecthing

TCId	TC14.4.1
Test case	Content Fecthing
Initial conditions	The user has selected an AXMEDIS Object distributed in the Content Catalogue. This may happen directly after catalogue content access or after Catalogue Content transaction.
Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	As the user selects content fetching the AXMEDIS plug-in opens and Content delivery starts. User can select the 3 different delivery modes: - **Streaming.** Similar to a broadcast experience, user acquires license and subsequently starts streaming content. Recommended only for higher bandwidth (450kb/s or above). - **Download.** After acquiring a license, the user can download the media (up to 10Mb/s encoding). Media can be viewd from the user's computer after the downloading process (can take 1-8 hours according to user access) - **Pre-Download**. User can first download content and then is prompted to purchase license. The user can check any time that the progress bar, indicating the download state, is advancing.

Partners, people involved	The Content Consumer (user) AXMEDIS plug-in
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-MCObject
Steps	1 The user selects delivery mode: pre-download, download, progressive download, streaming 2 The AXMEDIS plug-in opens and content delivery starts according to the delivery mode chosen by the user 3 The user opens the jobs panel where all current downloads are displayed 4 The user reads the remaining time for the end of transmission 5 The user can open the folder where the content is being received 6 The user can interrupt the reception of a given content
Expected results	The system shall have entered the next procedural step
Variations	None
Issues	The user, after opening the folder where the content is being received, deletes an incomplete and/or temporary file. This could put the AXMEDIS Client Application in an inconsistent state. The use may also activate a previously purchased license while fetching content in progressive download.
Test case Scope/Type	GUI, Backend / BlackBox

14.4.2 User Authentication Form

TCId	TC14.4.2
Test case	User Authentication Form
Initial conditions	The user has accessed to the Catalogue
Configuration description	A PC with an AXMEDIS end user client installation connected to internet.The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The user will be requested to authenticate in order to start any content fetch or transaction
Partners, people involved	The Content Customer (user) (involved in the purchase/rental operation) The MediaClub (entity performing all required checks to ensure that purchase/rental operations are valid and legal)
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-MCTestUser
Steps	1 The user enters his identification information (this does not necessarily mean personal details, it will be sufficient to have proper credentials, e.g., login/password) 2 The user credentials are sent to the MediaClub for verification 3 The user waits for the server response 4 If the user is identified as a regular one permission to proceed is granted and user can access all restricted areas of the Mediaclub that enable to fetch, purchase and acquire licenses for content, otherwise purchase procedure is aborted and user is sent back to browsing
Expected results	The system shall have entered the next procedural step
Variations	• This Authentication Form could be published by third party distributor (e.g., OD2, iLabs, Sejer, etc.). XML data will enable lay-out flexibility on the third party distributor website.
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.4.3 Catalogue Content Transaction

TCId	TC14.4.3
Test case	Catalogue Content Transaction
Initial conditions	The user has selected the Catalogue content
Configuration description	A PC with an AXMEDIS end user client installation connected to internet.The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The user is prompted with multiple payment options. Te user confirms the intention of purchasing the selected AXMEDIS Content. The user provides payment related information along with data needed to ensure legal validity of requested operation.
Partners, people involved	The Content Consumer (user) The MediaClub Payment Gateway
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-MCPayMethod
Steps	1 The MediaClub Payment Gateway shows to the user all billing information available including: 1.1 Price 1.2 Conditions for each selected item 1.3 Related use licence 1.4 Scope and limitations 1.5 Possible constraints 2 The MediaClub Payment Gateway asks the user to verify and accept presented terms 3 If the user accepts procedure continues otherwise is aborted and user is sent back to browsing 4 The user shall finalise billing information 5 Once billing information are provided the user is requested to select the payment method (credit card, electronic wallet, pre paid card, pre assigned tokens or similar) 6 The MediaClub Payment Gateway requires clearance to the AXMEDIS Distributor for the provided payment ID. 7 If payment ID is cleared the user will be charged the cost 8 The MediaClub Payment Gateway provides the system the proper clearance and the license delivery is authorized. 9 The user receives confirmation of transaction OK on a web page 10 The user receives an email notification that transaction has been succesful 11 User can start fetching content and come back subsequently in the user page for license activation. Alternatively the user can immediately activate license and start viewing content during content fetching
Expected results	The system shall have entered the next procedural step
Variations	• A supplementary actor could be a bank or other institution that will handle the money transaction and has to be a third trusted party for both the user and the AXMEDIS Certifier.
Issues	Certain methods of payment, such as premium phone or premium SMS, could be valid only within certain countries
Test case Scope/Type	GUI, Backend / BlackBox

14.4.4 Content Access

TCId	TC14.4.4
Test case	Content Access
Initial conditions	The AXMEDIS Content is successfully received.

Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The user accesses his local cache containing several AXMEDIS Objects.
Partners, people involved	The Content Consumer (user)
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-MCObject
Steps	1 The user accesses the AXMEDIS Object for playing it 2 The AXMEDIS Object is delivered to either the AXMEDIS Viewer or the standard application (with an additional AXMEDIS plug-in) 3 The application detects if the Object needs to acquire a license 4 The application finds a pre-acquired license for the Object and play it 5 The application needs a new license for the Object and tries to contact the MediaClub.
Expected results	The system shall have entered the next procedural step
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

14.4.5 Content Preview

TCId	TC14.4.5
Test case	Content Preview
Initial conditions	The AXMEDIS Object has been integrally received.
Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The user browses one/more AXMEDIS Object(s). The user opens and plays some short previews (if they are available) integrated with t\he received AXMEDIS Object. The user decides to buy or not the received AXMEDIS Content.
Partners, people involved	The Content Consumer (user)
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-MCObject
Steps	1 The user opens the AXMEDIS Object locally stored in his local cache 2 The user browses the AXMEDIS Object, using the AXMEDIS Info associated to the Object 3 The user reaches a preview available for the Object 4 The user plays the AXMEDIS Object Preview
Expected results	The system shall have entered the next procedural step
Variations	• One or more previews (depending on the internal structure of the AXMEDIS Object) should be available for the final user, in order to help him in the content evaluation before purchasing it.
Issues	None
Test case Scope/Type	GUI / BlackBox

14.4.6 License Acquisition

TCId	TC14.4.6
Test case	License Acquisition
Initial conditions	The user is logged-in to the MediaClub The user has selected to play an Axmedis content

Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The user plays the content
Partners, people involved	The Content Consumer (user)
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-MCObject, AXDS-DRMSupport3
Steps	1 The user opens the protected part of the AXMEDIS Object 2 The Object is delivered to the application/viewer charged to open/play it 3 The Application/Viewer has an internal plug-in able to detect if the Object to open needs a license 4 The AXMEDIS Viewer, using the internal plug-in, contacts the MediaClub in a protected mode (a secure connection is established with the MediaClub) 5 The MediaClub authorizes the AXMEDIS Certifier and Supervisor to provide the user with a license corresponding to the business rule associated to product purchased by the user 6 The user receives the AXMEDIS license useful to open the protected part of the AXMEDIS Object 7 The user receives a confirmation page that license has been successfully issued 8 The user consumes the AXMEDIS Object following the rules contained in the AXMEDIS license
Expected results	The user receives a license for playing the content
Variations	• Security, privacy and transparency are key requirements.
Issues	None
Test case Scope/Type	Backend / BlackBox

14.4.7 Multi-device license activation and back-up

TCId	TC14.4.7
Test case	Multi-device license activation and back-up
Initial conditions	The device must be supported by the AXMEDIS Client plug-in Any Content copy or backup has to be expressly authorized in the license terms.
Configuration description	A PC with an AXMEDIS end user client installation connected to internet. The Internet Connection is able to reach the MediaClub service.
Description of functionality to be tested	The user copies some interesting content in a a device other than initial PC
Partners, people involved	The Content Consumer (user)
Validator(s) skill	Users should be familiar with computers.
Data set used	None
Steps	1 The user opens the copy/backup interface of the AXMEDIS Client plug-in 2 The user selects all Objects involved in the copy operation 3 The user specifies the device where the AXMEDIS Content has to be copied. 4 the user can start a new license activation procedure (if he has right to activate license on new device) or else purchase new license for new device
Expected results	The system shall have entered the next procedural step
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

14.4.8 Pre ordering and registration for a group of students

TCId	TC14.4.8
Test case	Catalogue content purchase
Initial conditions	Pre ordering has been performed by the teacher (this is process not involving AXMEDIS) Teacher has an activation number for the product.
Configuration description	PCs with SEJER player installed and configured with the AXMEDIS client plug-ins (AXOM, PMS)
Description of functionality to be tested	Automatic registration of users
Partners, people involved	Actor is the student
Validator(s) skill	Low, End User
Data set used	AXDS-PCDist1
Steps	1 The Actor launch SEJER's player 2 The Actor enter the URL of the Object 3 The player displays an activation form, asking for the activation number 4 The Actor enters the activation Number the teacher has given to him and submit the form
Expected results	The Player displays the expected resource and the Actor is able to consult it according to his rights. The Actor has been automatically registered into the AXCS using a mix of computed identification data.
Variations	• Number of product associated with the Activation Number has already been Activated and the Actor tries to activate one more. o The Portals displays a Notice explaining to the User that he has to consult is teacher or a specific contact person to check the situation o The User is NOT registered into PMS • Actor enters a false Activation Number o The Portals displays a Notice explaining to the User that he has to consult is teacher or a specific contact person to check the situation o The User is NOT registered into PMS
Issues	
Test case Scope/Type	Backend/UnitTest

14.5 Business Models

14.5.1 Wallet

TCId	TC14.5.1
Test case	Wallet
Initial conditions	The customer is already registered to the distributor DB
Configuration description	Internet connection, browser, AXMEDIS Client plug-in installed, customer account
Description of functionality to be tested	Functionality to be tested: - Wallet generation - Wallet use - Wallet recharge - Wallet termination - ecare
Partners, people involved	End user
Validator skill	User should be familiar with content download, ecommerce transaction..
Data set used	AXDS-MCTestUser

Steps	1 Wallet creation with the first deposit and the generation of secure key 1.1 In the wallet ecare 1.2 try to check the balance 1.3 to Recharge 1.4 to Check the statement (List of deposits, List of the purchases done) 1.5 to Change the secure key 1.6 to Change payment method used 2 To access to the wallet ecare, to make payments, and to recharge the wallet, the security key is always requested. 3 To test a payments controlling the credit coverage 4 To control the wallet termination for expiration or real termination
Expected results	See 'description of functionality'
Variations	None
Issues	License is not issued because of license server failure after Media Club subtracts credits
Test case Scope/Type	GUI, Backend / BlackBox

14.5.2 Pay per minute

TCId	TC14.5.2
Test case	Business Model: Pay per minute (or per Kb/or per day)
Initial conditions	• The selection/query of the contents available is already done • The test on the transmission of the licence and of the key is already done • The test of the download, streaming… of the Axmedis Object is already done • The customer is a registered customer with billing relationship with the distributor
Configuration description	Internet connection, distributor subscription, browser, Axmedis Client plug-in installed, customer account
Description of functionality to be tested	Functionality to be tested: - The authentication works (the customer is certified as a subscribed user) - The business model is well translated from the business rules implemented in the licence to the distributor billing system (the billing Media Club application that provides the consumption information works) - Axmedis client plug in is able to track and report the number of minutes/Kb/day used by the customer (reported through the Media Club application that provides the consumption information)
Partners, people involved	End user
Validator skill	User should be familiar with content download, ecommerce transaction.
Data set used	AXDS-MCObject, AXDS-DRMSupport8
Steps	1 To use the AXMEDIS Object in different sessions, to track the number of minutes/Kb/day and to control the sum of the minutes/Kb/day is the same as the one transferred in the Media Club application that provides the consumption information)
Expected results	See 'description of functionality'
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.5.3 Rental

TCId	TC14.5.3

Test case	Business model: Rental
Initial conditions	• The selection/query of the contents available is already done • The test on the transmission of the licence and of the key is already done • The test of the download, streaming… of the Axmedis Object is already done
Configuration description	Internet connection, distributor subscription, browser, Axmedis Client plug-in installed, customer account
Description of functionality to be tested	Functionality to be tested: - Before providing the licence for the downloaded content or before starting the streaming session, the customer is required to provide the payment information - The expiration time is respected
Partners, people involved	End user
Validator skill	User should be familiar with content download, ecommerce transaction..
Data set used	AXDS-MCObject, AXDS-DRMSupport8
Steps	1 To try to access to the AXMEDIS object without the licence 2 To introduce the payment information 3 To activate the licence 4 To access to the content several time during the availability period 5 To access to the content after the availability period
Expected results	See ‘description of functionality’
Variations	None
Issues	License is not issued because of license server failure after payment gateway gives transaction ok
Test case Scope/Type	GUI, Backend / BlackBox

14.5.4 Sell through

TCId	TC14.5.4
Test case	Business model: Sell through
Initial conditions	• The selection/query of the contents available is already done • The test on the transmission of the licence and of the key is already done • The test of the download, streaming… of the Axmedis Object is already done
Configuration description	Internet connection, distributor subscription, browser, Axmedis Client plug-in installed, customer account
Description of functionality to be tested	Functionality to be tested: - Before providing the licence for the downloaded content or before starting the streaming session, the customer is required to provide the payment information - There is no expiration time
Partners, people involved	End user
Validator skill	User should be familiar with content download, ecommerce transaction..
Data set used	AXDS-MCObject, AXDS-DRMSupport8
Steps	1 To try to access to the AXMEDIS object without the licence 2 To introduce the payment information 3 To activate the licence 4 To access to the content several time and never the access s blocked due to expiration licence (as the licence never expires)
Expected results	See ‘description of functionality’
Variations	None

Issues	License is not issued because of license server failure after payment gateway gives transaction ok
Test case Scope/Type	GUI, Backend / BlackBox

14.5.5 Gift certificates

TCId	TC14.5.5
Test case	Gift Certificates
Initial conditions	• The customer making the gift is registered in the distributor DB • The customer receiving the gift has a mail account
Configuration description	Internet connection, distributor subscription, browser, Axmedis Client plug-in installed, customer account for the customer buying the gift, the email address of the gifted customer
Description of functionality to be tested	Functionality to be tested: - The gift certificate is generated with the right economic value - The pin code is generated with the value - The mail is sent - The process to redeem the credit works - The eventual credit can be redeemed later - The credit expires
Partners, people involved	*CustomerA*: buys a credit for a friend *CustomerB*: is the friend who receive the gift
Validator skill	User should be familiar with content download, ecommerce transaction..
Data set used	AXDS-MCObject, AXDS-MCTestUser
Steps	1 Gift certificate purchase 1.1 In the web site shop *customerA* clicks on a link ‘Buy a gift certificate’. This link is part of the portal. 2 Payment details 2.1 The gift certificate application asks to the customer: 2.2 the amount to buy 2.3 mail address of the friend 2.4 payment details 2.4.1 (these information are stored by the Distributor adding a PIN code) 3 An application sends an email to customerB 4 Procedure to redeem the gift certificates 4.1 CustomerB clicks on the link present in the mail reaching the Distributor application that recognize the PIN and knows the credit related; the credit is shown to the customer inviting him to start the standard purchase procedure (selection of staff to buy and ok to the kart content) 4.2 When customerB approves the Kart content, there is a control about the amount to pay and the value of the kart with 3 different situation: 4.2.1 gift value=value to purchase 4.2.1.1 the customer sees a confirmation page + receives an email 4.2.2 gift value>value of the kart 4.2.2.1 the customer can use the credit available in following purchases. Technically the value of the PIN code assigned to the customer is decreased ex. Gift certificate value = 50 € (that is the value associated to the PIN generated for that gift) customerB buys 30 € in Tiscali music club the new value of the PIN code is 20 € available for new purchases

	the customer sees a confirmation page reminding the credit available + receives an email with the link where to redeem the credit available 4.3 gift value<value of the kart 5 the customer is required to chose a payment method to pay the difference or to come back to the kart to remove some items
Expected results	See ‘description of functionality’
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.5.6 Subscription

TCId	TC14.5.6
Test case	Subscription
Initial conditions	• There is a subscription package available
Configuration description	Internet connection, browser, AXMEDIS Client plug-in installed
Description of functionality to be tested	Functionality to be tested: - The data collection is successful - The customer information are correctly controlled - After the registration the customer is able to access to AXMEDIS Objects without extra charges/extra information requests; eventually authentication only - The subscriber is not able to access to other contents not part of the subscription without paying
Partners, people involved	End user
Validator skill	User should be familiar with content download, subscription procedure
Data set used	AXDS-MCObject, AXDS-MCTestUser
Steps	1 To use the AXMEDIS Object in different sessions, to track the number of minutes/Kb/day and to control the sum of the minutes/Kb/day is the same as the one transferred in the Media Club application that provides the consumption information)
Expected results	Subscription is successful and billing successful
Variations	To try a test with an active customer and with a new customer
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

14.5.7 Credits deduction

TCId	TC14.5.7
Test case	Credits deduction
Initial conditions	• The selection/query of the contents available is already done • The test on the transmission of the licence is already done • The test of the download, streaming… of the Axmedis Object is already done
Configuration description	Internet connection, distributor subscription, browser, Axmedis Client plug-in installed
Description of functionality to be tested	Functionality to be tested: - The licence transmission is blocked if the customer has not sufficient credits

	- The Media Club application that provides the consumption information export the number of credit to charge to the customer
Partners, people involved	End User
Validator skill	User should be familiar with content download, ecommerce transaction
Data set used	AXDS-MCObject, AXDS-MCTestUser
Steps	1 to acquire the AXMEDIS Object 2 to control that the number of credits exported are correct (by Media Club application that provides the consumption information)
Expected results	See 'description of functionality'
Variations	To try with a customer accounts having sufficient credits and with others with un-sufficient credits
Issues	Backend / BlackBox

14.6 User Login

14.6.1 User authentication through an external Single Sign On (SSO) system

TCId	TC14.6.1
Test case	User Login
Initial conditions	AXMEDIS Player is accessible within a VLE (Virtual Learning Environment) that has an agreement with an AXMEDIS distributor (*i.e.* distribution channel) The Actor is registered as a valid VLE system Agreement between the VLE Provider provides valid licenses for some product for the Actor
Configuration description	PC within the domain covered by the VLE VLE client software including AXMEDIS player installed and properly configured
Description of functionality to be tested	User authentication for the case where authentication is performed outside AXMEDIS
Partners, people involved	Actor is the final User of the software
Validator(s) skill	Low, End User
Data set used	None
Steps	1 The Actor logs into the VLE 2 The Actor launch the AXMEDIS Player available within the VLE
Expected results	The Actor does not have to authenticate itself again in the AXMEDIS Player The actor Player displays the distributor portal, with a list of resources available to the Actor The actor is able to select a view one of these resources
Variations	• The Actor is registered in the VLE but registration has not been propagated to the AXMEDIS Framework: o When launching the AXMEDIS Player, the Actor is prompted for its authentication
Issues	The test case may vary depending on the way the AXMEDIS Framework manages the corresponding Use Case.
Test case Scope/Type	Backend/UnitTest

15 AXMEDIS for Distribution towards Mobiles (COMVERSE)

15.1 Transcoding content

15.1.1 Successfully Transcode New Objects, on a New-Match Event

TCId	TC15.1.1
Test case	Successfully Transcode New Objects, on a New-Match Event
Initial conditions	• All of the AXMEDIS services are online and operational. • The producer's publication tool is running and connected to the AXMEDIS network. It is ready to publish five new Audio objects (AXAU1-5) and five new Graphic objects (AXGR1-5). • The Transcoding Platform has all the required plugins, extensions and codecs that are required to perform the transcoding successfully (with respect to source and destination formats) installed and operative. • The Transcoding Server is up and running, and connected to the AXMEDIS network. • The Transcoding Server is idle (no transcoding process is in progress). • The Transcoding Server has hooks on the Notification AXEPTool. It is waiting for a notification of new objects matching the Selection.
Configuration description	• All of the producer's new objects have never been published to the AXMEDIS network. They are ready and valid for publication. • The properties and characteristics of Objects AXAUO1-3 and AXGR1-3 match the criteria of the Transcoding Server's Selection. The rest of the new objects do not. • The Transcoding Server is configured to transcode all Audio objects to two formats – AUF1 (e.g. one minute long, high quality WMA) and AUF2. • The Transcoding Server is configured to transcode all Graphic objects to one format - GRF1 (e.g. 20x20 pixels, 16 colors GIF).
Description of functionality to be tested	Successfully transcode new AXMEDIS Audio and Graphic objects when they are published to the AXMEDIS network. Once the producer publishes the new objects, the AXEPTool notifies the Transcoding Server, which fetches the objects, formats them to the desired formats, and publishes the resulting files as new objects to the AXMEDIS network.
Partners, people involved	Producer (Steps), Transcoding Server Operator (Configuration and Validation)
Validator(s) skill	• Must be able to create new Audio and Graphic objects and publish them to the AXMEDIS network. • Must be familiar with the Transcoding Server configuration, deployment, database, logs. • Must be familiar with Audio and Graphic media formats.
Data set used	AXMEDIS objects and their (Audio and Graphic) content to transcode
Steps	1 The producer publishes all the new Audio and Graphic objects (AXAU1-5, AXGR1-5) to the AXMEDIS network.
Expected results	The GUI of the producer's publication tool indicates that all of the objects were published successfully. For each Audio object matching the transcoding criteria (AXAU1-3) the Transcoding Server has successfully published two new objects – one in format AUF1 and the other in AUF2. For each Graphic object matching the transcoding criteria (AXGR1-3) the Transcoding Server has successfully published a single object in format GRF1.

	The Transcoding Server updated the Selection criteria to exclude all of the objects that were successfully transcoded and published in the latest transcoding operation. The Transcoding Server appended the details of the latest transcoding process (begin and end time, objects transcoded, etc) to its log.
Variations	• Scheduled a transcoding process instead of triggering it on a New-Match event. • Publish new objects of different types (e.g. AUDIO, GRAPHIC, TEXT, ICONS, RINGTONES, etc) and formats (e.g. RTTTL, JPEG, GIF, WAV, MP3, ASCII, UNICODE, PDF, Word, OTB, IMY, etc, in different dimensions, colour-depth, bit rate, length, etc) to transcode; • Modify the criteria of the Transcoding Server Selection so that All, Some, or No objects will require transcoding. • Change the destination formats for each source object type.
Issues	• The criteria for objects-to-transcode must NEVER match the characteristics and properties of the destination formats, as it will engage the transcoding server in a never-ending, futile transcoding activity. • The Transcoding and Publishing processes are working asynchronous to the Transcoding engine. The Validator has to know when the processes ended before validating results (GUI, alerts, etc).
Test case Scope/Type	GUI, backend / BlackBox

15.1.2 Fail to Transcode New Objects on Scheduled Event

TCId	TC15.1.2
Test case	Fail to Transcode New Objects on Scheduled Event
Initial conditions	• All the AXMEDIS services are online and operational. • There are four new audio objects published to the AXMEDIS network (AXAU1-4). They were never transcoded by the Transcoding Server before. • The Transcoding Platform has all the required components that are required to perform the transcoding successfully (with respect to source and destination formats) installed and operative, with the exception of the components required for the encoding of format AUF2, which is not installed. • The Transcoding Server is up and running, and connected to the AXMEDIS network. • The Transcoding Server is idle (no transcoding process is in progress). • The Transcoding Server has NO hooks on the Notification AXEPTool. It's NOT waiting for notifications. • The Scheduler is running, and is about to prompt the Transcoding Server to begin the transcoding procedure. • A monitoring application (e.g. SNMP agent, e-mail client, SMS on mobile phone, GUI client) is waiting for alerts from the Transcoding Server.
Configuration description	• The properties and characteristics of all the new objects (AXAU1-4) match the criteria of the Transcoding Server's Selection. • All of the producer's new objects must contain Audio content in format AUF1, except for object AXAU3 that must contain content in format AUF2. • The following anomalies must be applied: ○ The protected object AXAU1 must be corrupted in a way the Transcoding Server would not be able to unprotect it or extract

	the content form it. ○ The object AXAU2 must indicate that it contains data in format AUF1, but the unprotected content data must contain garbage that the Transcoding Server can't open. • The Transcoding Server must be configured to transcode all objects in format AUF1 to format AUF3, and all objects in format AUF2 to format AUF4. • The component that is required to decode audio content to format AUF4 is not installed on the Transcoding Server. • The Transcoding Server must be configured to attempt to publish transcoded objects two times. • The Transcoding Server must be configured to attempt to transcode objects two times, in two separate sessions, before they are rejected. • The AXMEDIS platform must deny the option to publish from the Transcoding Server access. • The Transcoding server must be configured to send all failure alerts to the monitoring application (e.g. by SNMP, e-mail to supervisor mail address/mobile, etc).
Description of functionality to be tested	Failure to transcode and publish new AXMEDIS Audio objects due to common reasons, upon a scheduled event. The scheduler prompts the Transcoding Server to begin a transcoding session. The server fetches the files but fails to (a) unprotect an object (extract content); (b) open/load content; (c) convert content; and (d) publish the resulting objects to the AXMEDIS network.
Partners, people involved	Transcoding Server Operator (Configuration and Validation), Scheduler Operator.
Validator(s) skill	• Must be familiar with the Transcoding Server configuration, deployment, queues, database and logs. • Must be familiar with Audio media formats. • Must be familiar with the Scheduling tool (e.g. MS Scheduler).
Data set used	AXMEDIS objects and their (Audio and Graphic) content to transcode Transcoding Server Database for Failed transcoding attempts and Rejects
Steps	1 Scheduler tool fires the event to initiate the transcoding process.
Expected results	• The scheduler GUI indicates that the event scheduled to trigger the Transcoding Event was successfully fired. • The Transcoding Server did not publish any object to the AXMEDIS object (query the AXDB to verify). • A single AXMEDIS object - AXRE4 - was successfully transcoded from object AXAU4 to format AUF1 and protected. It is resident in the transcoding server cache, waiting for a second publication attempt. • The Transcoding Server's Publish queue contains the single entry, which indicates that AXRE4 failed publication one time, and is ready to attempt publication for a 2second (and final) time. • Records were added to the Transcoding Server's DB describing: ○ Object AXAU1 failed transcoding one time, because the server was unable to unpack/unprotect the content. ○ Object AXAU2 failed transcoding one time, because the content couldn't be open. ○ Object AXAU3 failed transcoding one time, because the required encoder/plugin was not available. ○ Object AXAU failed publishing one time, because the Transcoding Server was denied write access to the DB.

	- The Transcoding Server's Rejects bin is empty. - The Transcoding Server's log is added with records describing what is described in the DB. - The Transcoding Server updates the Selection's criteria to include all the objects that failed the transcoding (AXAU1-3). - The monitoring application received alerts of all the failures from the Transcoding Server.
Variations	- Make a second iteration to attempt and fail to transcode and publish the same objects, and validate they were added to the Rejects bin and excluded from the Selection criteria for good. - Trigger the transcoding process on a New-Match event rather than a scheduled event. - Publish new objects of different types (e.g. AUDIO, GRAPHIC, TEXT, etc) and formats (e.g. RTTTL, JPEG, GIF, WAV, MP3, ASCII, UNICODE, PDF, Word, OTB, IMY, etc, in different dimensions, colour-depth, bit rate, length, etc) to transcode; - Modify the criteria of the Transcoding Server Selection so that All, Some, or No objects will require transcoding. - Change the destination formats for each source type.
Issues	- The criteria for objects-to-transcode must NEVER match the characteristics and properties of the destination formats, as it will engage the transcoding server in a never-ending, futile transcoding activity. - The Transcoding and Publishing processes are working asynchronous to the Transcoding engine. The Validator has to know when the processes ended before validating results.
Test case Scope/Type	GUI, backend / BlackBox

15.2 Provisioning

15.2.1 Browse the Personalized Content tree through the WEB interface

TCId	TC15.2.1
Test case	Browse the Personalized Content tree through the WEB interface.
Initial conditions	- The Comverse Distribution system and all its components, including the WEB interface, Personalization Engine, and Handset Management Engine are running and operative. - The content tree – Categories and the associated content list – is loaded.
Configuration description	- The content tree is organized as follows: o Root ▪ genre1 • artist1 o song1-1 • artist2 o song2-1, dated 1980 o song2-2, dated 1985 • artist3 o song3-1, dated 1998 o song3-2, dated 2000 o song3-3x, dated 2002 o song3-4, dated 2004 ▪ genre2 • artist2

	o song2-1, dated 1980 o song2-2, dated 1985 • artist3 o song3-1, dated 1998 o song3-2, dated 2000 o song3-3x, dated 2002 o song3-4, dated 2004 ▪ genre3 • artist4 o song1-1x o song2-1x o Example: ▪ Root\Classic\J.S.Bach\Organ Works, 1987 ▪ Root\Psychedelic\Pink Floyd\Echoes, 1969 • The provisioning web site supports the Subscriber's browser. • The system requires the Subscriber to enter the destination handset type and model before he/she begins to browse. • The Subscriber's handset supports all the content that is not marked with an x. It does not support the content that is marked with an x. • The Subscriber has never browsed the content tree before. His/her profile in not personalized (this condition is only relevant for testing the default subscriber profile). • The default Subscriber profile requires that the content tree will be presented to the subscriber sorted alphabetically in ascending order. • The Personalization Engine must record and set the following preferences for each Subscriber in every session: o Sort field (title, date, etc) for song list o Sort Order of song list o Most recently browsed Artist and Genre Categories. • The Handset Management Engine must remove all items that are not supported by the Subscriber's handset, and all the empty Categories (removing all items will consequently leave the category empty).
Description of functionality to be tested	The Subscriber uses the WEB interface to browse a content tree that is adjusted by the Handset Management Engine and the Personalization Engine. The Subscriber browses the content tree through the WEB interface. The Handset Management Engine filters out all content items that are not supported by the Subscriber's handset. The Personalization Engine adjusts the content tree according to the subscriber's preferences/behaviour while he/she is browsing.
Partners, people involved	Subscriber, Comverse Distribution system Operator (Configuration and Validation)
Validator(s) skill	Must be familiar with the configuration and operation of the Comverse distribution system. Must be able to use a browser to browse the content tree through the system's WEB interface.
Data set used	Subscriber Profile in the Personalization Engine Database Handset Management Engine Database for filtering unsupported content and menu options Content Tree (AXMEDIS objects as Categories and the Associated Selections)
Steps	1 The Subscriber opens the browser and browse to the provisioning web site. 2 The Subscriber selects his handset model and type as the destination handset. 3 The Subscriber selects Category Genre1. 4 The Subscriber selects Category Artist 3.

	5 The Subscriber selects to sort the list of items by date, in descending order. 6 The Subscriber browses back to Category Genre1. 7 The Subscriber browses back to the Root Category. 8 The Subscriber selects Category Genre2. 9 The Subscriber selects Artist2. 10 The Subscribers selects to sort the list in ascending order. 11 The Subscriber closes the browser.
Expected results	• The Subscriber’s browser presents a list of handset types and models to choose as the destination handset, before he/she can begin browsing the content. • The Subscriber’s browser presents a menu listing the Genre categories that are not empty, in alphabetical order: o Root ▪ Genre1 ▪ Genre2 • The Subscriber’s browser presents a menu listing the Artist categories in the Genre1, sorted alphabetically, for the subscriber to choose from. o Genre1 ▪ Artist1 ▪ Artist2 ▪ Artist3 • The Subscriber’s browser presents a menu listing the songs of Artist3 that are supported by the Subscriber’s handset, sorted alphabetically. o Artist3 ▪ Song3-1 ▪ Song3-2 ▪ Song3-4 • The list of items is re-sorted by date, in descending order, and is presented on the Subscriber’s browser. o Artist3 ▪ Song3-4 ▪ Song3-2 ▪ Song3-1 • The Subscriber’s browser presents a menu listing all the Artist categories in Genre1, sorted alphabetically. Artist3 is at the top of the list (MRU). o Genre1 ▪ Artist3 ▪ Artist1 ▪ Artist2 • The Subscriber’s browser presents a menu listing the Genre categories that are not empty, in alphabetical order. The Genre1 category is at the top of the list (MRU) • The Subscriber’s browser presents a menu listing the Artist categories in Genre2, sorted alphabetically. Artist3 is at the top of the list (MRU). o Genre2 ▪ Artist3 ▪ Artist2 • The Subscriber’s browser presents a menu listing the songs of Artist2, sorted by date in descending order. o Artist2 ▪ Song2-2 ▪ Song2-1 • The Subscriber’s browser presents a menu listing the songs of Artist2,

	sorted by date in ascending order. o Artist2 ▪ Song2-1 ▪ Song2-2 • The handset notifies the Subscriber that he/she ended the session. • The following preferences for browsing content are stored as the Subscriber's profile in the Personalization database: o The preferred sort field is Date o The preferred sort order is Ascending o The most recently used Artists are (i) Artist2 and (ii) Artist3, in that order. o The Most Recently used Genre categories are (i) Genre2 and (ii) Genre1, in that order.
Variations	• The Subscriber can browse the content tree through other supported interfaces (i.e. IVR, WAP, etc) • The Personalization Engine monitors and records other preferences for browsing content.
Issues	• The preference characteristics and behaviours for personalization are still being defined. The few suggested in this scenario (MRU, Sort Field, Sort Order) are the most probable.
Test case Scope/Type	GUI, backend / BlackBox

15.2.2 Sample Audio Content through the WEB interface

TCId	TC15.2.2
Test case	Sample Audio Content through the WEB interface.
Initial conditions	• The Comverse Distribution system and all its components, including the WEB interface, are running and operative. • The sample file AUS1 is available to the system in a format that is supported by the WEB interface (e.g. 30 seconds long, WMA). • The Subscriber is browsing the content tree and has already entered his/her handset type and model as the destination handset, and is browsing the content tree.
Configuration description	• The system supports the subscriber's handset type and model (e.g. Nokia 6100). • The Subscriber's browser is has the component/plugin that is required for playing AUS1 installed and ready. • The subscriber must be able to browse the content tree using the WEB interface. • The content tree includes the Audio Item AUI1. • The options menu that the system presents to the Subscriber when he/she reaches AUI1 must include the option to listen to a sample of AUI1 (i.e. AUS1).
Description of functionality to be tested	The Subscriber chooses to listen to a sample of a desired audio content. The system plays the content to the Subscriber.
Partners, people involved	• Subscriber. • Comverse Distribution system Operator (Configuration and Validation).
Validator(s) skill	• Must be familiar with the configuration and operation of the Comverse distribution system. • Must be able to use a browser to browse the content tree through the system's WEB interface.

Data set used	AUI1- the offered Audio content AUS1 - the sample of the offered Audio content AUI1
Steps	1 The Subscriber browses the content tree up to item AUI1. 2 The Subscriber chooses to listen to a sample of AUI1. 3 The Subscriber listens to AUS1 - the sample of AUI1.
Expected results	The Subscriber's browser presents the list of available actions, including the option to listen to a sample of AUI1. The Subscriber's browser plays the sample file – AUS1 – in the suitable component/plugin. The system added a record of the sample action details (time and date, Subscriber ID, etc) to the log.
Variations	• The Subscriber can ask for a sample media of different type (Graphic, Text, etc) and format (RTTTL, JPEG, MP3, UNICODE, PDF, etc). • The browser may offer the subscriber the option to download the content rather than playing it in the plugin. • The subscriber can choose to stop the sample while it's playing. • The subscriber can be presented with the option to purchase the content rather than sample it.
Issues	None
Test case Scope/Type	GUI, backend / BlackBox

15.2.3 Successfully purchase content through the SMS interface

TCId	TC15.2.3
Test case	Purchase content through the SMS interface.
Initial conditions	• The Comverse Distribution system and all its components, including the SMS interface, are running and operative. • The Audio content item AUI1 is available to the system in a format that is supported by the Subscriber's handset (e.g. 15 seconds long MIDI4 file). • The Billing Service is up and running. It's available to accept and process requests from the system.
Configuration description	• The system supports the subscriber's handset type and model (e.g. Nokia 6100). • The Subscriber's handset must be able to send an SMS to the system. • The system must push AUI1 to any handset that sends an SMS with the code CD1 to the system. • The system must bill the subscriber before pushing a purchased item.
Description of functionality to be tested	The Subscriber sends an SMS code to the system, asking to purchase a specific content item. The system updates the billing system and pushes the content to the Subscriber.
Partners, people involved	• Subscriber • Comverse Distribution system operator (Configuration and Validation) • Billing system operator.
Validator(s) skill	• Must be familiar with the configuration and operation of the Comverse distribution system. • Must know how to send SMS messages to the system • Must be familiar with the configuration and operation of the Billing system.
Data set used	AUI1 – the offered Audio content Billing system database
Steps	1 The Subscriber sends an SMS with the code CD1 to the system.

Expected results	A record was added to the billing system recording the purchase and its details (Time and date, Subscriber ID, AUI1, price, etc). The system pushed AUI1 to the Subscriber's handset. The Subscriber's handset indicates that the purchased content has arrived to the handset. A record was added to the system log with the purchase details (Time and date, Subscriber ID, AUI1, etc).
Variations	• If AUI1 is not available in a format that the subscriber's handset supports, the system must reply with a message stating this. • The subscriber can purchase content for a destination handset different from the one he/she used to send the SMS message. • Try this scenario on Pre-Paid and Post-Paid subscribers. • The Subscriber can purchase content through the other system interfaces (e.g. Web, IVR, WAP, etc)
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.4 Fail to send content purchased through the SMS interface to the Subscriber

TCId	TC15.2.4
Test case	Fail to send content purchased through the SMS interface to the Subscriber.
Initial conditions	• The Comverse Distribution system and all its components, including the SMS interface, are running and operative, with the exception of the WAP service, which is down. • The Audio content item AUI1 is available to the system in a format that is supported by the Subscriber's handset (e.g. 15 seconds long MIDI4 file). • The Billing Service is up and running. It's available to accept and process requests from the system.
Configuration description	• The system supports the subscriber's handset type and model (e.g. Nokia 6100). • The Subscriber's handset must be able to send an SMS to the system. • The system must push AUI1 to any handset that sends an SMS with the code CD1 to the system. • The system must bill the subscriber before pushing a purchased item.
Description of functionality to be tested	The Subscriber sends an SMS code to the system, asking to purchase a specific content item. The system fails to send the content to the Subscriber.
Partners, people involved	• Subscriber • Comverse Distribution system operator (Configuration and Validation) • Billing system operator.
Validator(s) skill	• Must be familiar with the configuration and operation of the Comverse distribution system. • Must know how to send SMS messages to the system • Must be familiar with the configuration and operation of the Billing system.
Data set used	AUI1 – the offered Audio content
Steps	1 The Subscriber sends an SMS with the code CD1 to the system.
Expected results	The Subscriber's handset received a message describing that the transaction failed, and that he/she was not billed. No record was added to the billing system. A record was added to the system log with the purchase failure details (Time and date, Subscriber ID, AUI1, etc).

Variations	• If AUI1 is not available in a format that the subscriber's handset supports, the system must reply with a message stating this. • The subscriber can purchase content for a destination handset different from the one he/she used to send the SMS message. • The Subscriber can purchase content through the other system interfaces (e.g. WEB, IVR, WAP, etc)
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

16 AXMEDIS for Distribution towards i-TV (WP4.8, WP9.3: EUTELSAT)

16.1 User Terminal Installation and Configuration

TCId	TC16.1
Test case	User Terminal Installation and Configuration
Initial conditions	The user has completed the hardware and software installation steps. The spectrum analyser (used by the installer of the satellite dish) indicates a good quality of signal of the satellite from which the Satellite Data Broadcast will be received. The backend shall be running and sending some basic data through the Satellite Channel.
Configuration description	A user client station connected to internet and to a satellite dish.
Description of functionality to be tested	The user is ready to use the AXMEDIS service and access the published Content. (Access can be restricted only to some components). Well functioning of the AXMEDIS Client after installation: • The Client is able to lock the signal coming from the appropriate transponder; • The Client is able to switch from a transponder to another; • The AXMEDIS Client is able to run correctly; • The AXMEDIS Client does not create any conflicts with the previously installed applications; The AXMEDIS Client is able to stop its execution.
Partners, people involved	Professional installers of satellite dish and on-line technical support.
Validator(s) skill	Users should be familiar with computers. Users have to be able to execute some steps suggested by the technical support
Data set used	AXDS-ITV4
Steps	1 The User launches the AXMEDIS Client using the desktop shortcut; 2 The User checks that the AXMEDIS Client Application is working correctly: 2.1 He opens the GUI and the Electronic Programme Guide is displayed and periodically refreshed; 2.2 The integrated DVB Tuner of the DVB Adapter is locking the signal by displaying a green light in the related Lock Box; 3 The User stops the AXMEDIS Client Application.
Expected results	The user can connect to AXMEDIS service, but not to restricted sections. The AXMEDIS Client (limited to basic functionalities) works fine: • The AXMEDIS Client starts/stops and behaves correctly; • All previously installed applications still work fine while AXMEDIS Client is running
Variations	• The AXMEDIS Client can automatically be launched at system start up.
Issues	In case of problems, the User should contact the technical support for troubleshooting.
Test case Scope/Type	GUI, Backend / WhiteBox

16.1.1 User Hardware Installation

TCId	TC16.1.1
Test case	User Hardware Installation
Initial conditions	The user's PC is connected to a satellite dish, correctly pointed to the satellite providing the Data Broadcast. The spectrum analyser (used by the installer of the satellite dish) indicates a good quality of signal of the satellite from which the Satellite Data Broadcast will be

	received. The user has installed a DVB Adapter on his PC by using a PCI slot, an Ethernet port or an USB connector.
Configuration description	The satellite cable is properly connected to the DVB Adapter.
Description of functionality to be tested	The DVB Adapter is properly installed and is locking the satellite signal.
Partners, people involved	Professional installers of satellite dish and on-line technical support.
Validator(s) skill	Users should be familiar with satellite antenna installation.
Data set used	AXDS-ITV5
Steps	1 The User connects the satellite cable to his DVB Adapter; 2 The User verifies the card blinking for the activity; 3 The User launches the DVB Tuner Application and tries to lock the satellite transponder where the AXMEDIS Service is transmitting;
Expected results	The DVB Adapter is correctly connected to the satellite cable; The DVB Adapter light is blinking to indicate an activity; The DVB Adapter is able to lock the satellite transponder.
Variations	None
Issues	None
Test case Scope/Type	Backend / WhiteBox.

16.1.2 User Software Installation

16.1.3 User Registration

This test case refers to use case 16.1.3 User Registration for the registration of the AXMEDIS Client Application

TCId	TC16.1.3
Test case	User Registration
Initial conditions	The AXMEDIS Client is well installed and it works fine (able to receive basic data from AXMEDIS Satellite Data Channel). The User has followed step by step the registration wizard (part of the B2C Application). The backend shall be up and running and able to treat all incoming registration request from the Users.
Configuration description	The Internet Connection is able to reach the server for registering the AXMEDIS Client Application.
Description of functionality to be tested	The User has finished the Client registration procedure and now is able to completely access the AXMEDIS Service. The User has received all authorizations in order to receive all type of contents. The User has received some filters in order to receive only the really interesting content.
Partners, people involved	AXMEDIS User, B2C Satellite Data Broadcaster.
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-ITVlogin, AXDS-ITVpreferences
Steps	1 The User verifies that the Registration finishes with no errors. 2 The User verifies the correct reception of all Authorizations associated with the test login (this verification could need a while to be finished because Authorizations are simultaneously distributed to all Users). 3 The User verifies the correct reception of all Filters associated with the test login.

	4 The User is able to see more content in the guide available for the selection.
Expected results	The User has successfully finished the Client Registration; he has completely received all related Authorizations; he has completely received all related Filters. The User can entirely access to the complete AXMEDIS Offer reserved to him.
Variations	None.
Issues	None.
Test case Scope/Type	GUI, Backend / WhiteBox

16.1.3.1 User Profiling – Server Side

TCId	TC16.1.3.1
Test case	User Profiling
Initial conditions	The server receives the User Profile about AXMEDIS contents, from some AXMEDIS Client Applications
Configuration description	The Internet Connection is able to connect the AXMEDIS Client Application to the server for storing the User Profile
Description of functionality to be tested	The Server receives the User Profile from some AXMEDIS Client Applications and stores them in its database.
Partners, people involved	AXMEDIS User, B2C Satellite Data Broadcaster.
Validator(s) skill	Familiarity with database management
Data set used	AXDS-ITVprofile
Steps	1 The Server verifies the correct reception of the User Profiles 2 The Server verifies the correct management of the database of User Profiles.
Expected results	The Server successfully receives the User Profiles from the Client Application. The Server successfully stores the User Profiles and manages the database of User Profiles.
Variations	None.
Issues	None.
Test case Scope/Type	Backend/WhiteBox

16.1.3.2 User Profiling – Client side

TCId	TC16.1.3.2
Test case	User Profiling
Initial conditions	The User is registered and gives his/her preferences about AXMEDIS contents
Configuration description	The Internet Connection is able to reach the server for storing the User Profile
Description of functionality to be tested	The User runs the AXMEDIS Client Application User Profiling procedure The User provides his/her preferences about AXMEDIS content The User choose what should and what should not be included in his/her profile The User saves his/her profile
Partners, people involved	AXMEDIS End User
Validator(s) skill	Users should be familiar with computers.
Data set used	AXDS-ITVpreferences. AXDS-ITVprofile
Steps	1 The User verifies that initial manual User Profiling set up finishes with no errors 2 The User verifies the correct effects of his preferences, behaviours, votes, and choices 3 The User verifies that he/she is able to manually change his/her preferences 4 The User verifies the possibility to avoid sending private information to the

	server
Expected results	The User has successfully finished the User Profiling procedure; the Client Application depends on the User preferences, behaviours, votes, and choices; The User can access his/her Profile and manually correct it. The User can choose if remove private information from his/her Profile.
Variations	None
Issues	None
Test case Scope/Type	GUI/WhiteBox

16.2 Content Listing

16.2.1 Content Web Listing

TCId	TC16.2.1
Test case	Content Web Listing
Initial conditions	The user has an Internet Connection. The User has registered the Application to the AXMEDIS project.
Configuration description	A user client station connected to internet and to a satellite dish. On the AXMEDIS web page, a list of content is available to be browsed.
Description of functionality to be tested	The user accesses the web page containing the list of the proposed AXMEDIS content. The User browses the content listed in order to find some interesting contents.
Partners, people involved	Content consumer (user)
Validator(s) skill	User should be familiar with Internet Browsing.
Data set used	None
Steps	1 The user reaches the AXMEDIS Content Web List 2 The user displays the proposed content using different criteria (type, author, content producer, production date) 3 The user inserts some key words for filtering Object potentially interesting for him 4 The user reads all available information (contained in the AXMEDIS Info) associated to the AXMEDIS Object, helpful for voting
Expected results	The user can browse the content on the web page, and filter it according to selected criteria.
Variations	• Use different criteria to filter the content.
Issues	None
Test case Scope/Type	GUI / BlackBox

16.2.2 Content Carousel Listing

TCId	TC16.2.2
Test case	Content Carousel Listing
Initial conditions	The user has a fully operational AXMEDIS Client Application; The list of content to be browsed has to be consistent. The backend has to be up and running. It regularly sends content in the AXMEDIS Channel.
Configuration description	A user client station connected to internet and to a satellite dish.
Description of functionality to be tested	The user consults from the AXMEDIS Client Application the list of the AXMEDIS Carousel currently in transmission. The User browses and previews the content listed in order to find some interesting contents.

Partners, people involved	Content consumer (user)
Validator(s) skill	User should be familiar with P2P-like Application (e.g. Kazaa) because of analogies with the AXMEDIS Client GUI.
Data set used	AXDS-ITVobjects, AXDS-ITVpackages, AXDS-ITVprofile
Steps	1 The user opens the AXMEDIS Client Application 2 The user uses some pre-defined functionalities to filter the content 3 The user applies his/her own profile (locally stored) to the AXMEDIS offer to best match his/her interest in the offered content 4 The user enters some key words in the content browsing 5 The user reads all available information (contained in the AXMEDIS Info) associated to the AXMEDIS Object, helpful for selection 6 The user plays some short previews (if this option is available) associated to the AXMEDIS Object, previously extracted from the AXMEDIS Info and added to the Electronic Programme Guide (constantly transmitted to AXMEDIS users) of the AXMEDIS Service.
Expected results	The user can browse the content currently transmitted, and filter it according to specific criteria.
Variations	Use different criteria to filter the content.
Issues	None
Test case Scope/Type	GUI / WhiteBox.

16.3 Content Voting

TCId	TC16.3
Test case	Content Voting
Initial conditions	The user can browse the content on the AXMEDIS web page, and filter it according to selected criteria.
Configuration description	User connected to Internet.
Description of functionality to be tested	The user expresses preferences on AXMEDIS Objects contained in the AXMEDIS Content Web List.
Partners, people involved	Content Consumer (user)
Validator(s) skill	User should be familiar with the Internet Browsing.
Data set used	AXDS-ITVobjects
Steps	1 The user chooses one or more AXMEDIS Objects he/she wishes to receive by push inside the AXMEDIS Carousel.
Expected results	The user receives a receipt about his vote expression and a notification saying if the AXMEDIS voted Object has entered in the AXMEDIS Carousel
Variations	• The user has exceeded the maximum number of preferences for the period: in this case the result of the test is negative. The user votes some AXMEDIS Objects not accepted in the AXMEDIS Carousel, he receives a negative notify.
Issues	None
Test case Scope/Type	Backend / WhiteBox

16.4 Content Selection

16.4.1 Manual Content Selection

TCId	TC16.4.1
Test case	Manual Content Selection
Initial conditions	The user can browse on the AXMEDIS Client Application the content currently

	transmitted, and filter it according to specific criteria. The backend is continuously transmitting content through the AXMEDIS Satellite Channel.
Configuration description	None.
Description of functionality to be tested	The user selects (manually) the scheduled content that will be received at the indicated time by push.
Partners, people involved	Content Consumer (user)
Validator(s) skill	The user should be familiar with P2P-like Application: AXMEDIS Client Interface will be similar to this type of Applications.
Data set used	AXDS-ITVobjects
Steps	1 The user double clicks on the AXMEDIS Object in order to select it for reception. 2 The user verifies that the Content has been selected by checking in the Downloading Panel of the AXMEDIS Client Interface.
Expected results	The user sees the selected AXMEDIS Object in the Downloading panel of the Client Application Interface. This proves that the content has been scheduled for reception.
Variations	• The Content Selection could be affected from a remote computer and the order passed to the local AXMEDIS Client Application.
Issues	The user has to leave turned on the computer (not the internet connection) during the time window of the selected transmission.
Test case Scope/Type	GUI / WhiteBox.

16.4.2 Automatic Content Selection

TCId	TC16.4.2
Test case	Automatic Content Selection
Initial conditions	The user has voted an AXMEDIS Object and received a message notifying the expected start date of the selected AXMEDIS Object.
Configuration description	An AXMEDIS Client Application up and running.
Description of functionality to be tested	The user automatically receives the AXMEDIS Object he/she voted, and that has been added to the AXMEDIS Carousel.
Partners, people involved	Content Consumer (user)
Validator(s) skill	User should be familiar with Computers
Data set used	AXDS-ITVobjects
Steps	1 The user turns on his AXMEDIS Client before the transmission starts.
Expected results	The AXMEDIS Object is downloaded on the Client Application automatically (no need to explicitly select it).
Variations	None
Issues	The user has to leave turned on the computer (not the internet connection) during the time window of the selected transmission.
Test case Scope/Type	Application Core Test / UnitTest

16.5 Content Reception

TCId	TC16.5
Test case	Content Reception
Initial conditions	The user has started a download, with automatic download or by manual select.

Configuration description	The AXMEDIS Client Application is up and running.
Description of functionality to be tested	The user can check the progression of downloads.
Partners, people involved	Content Consumer (user)
Validator(s) skill	User has a normal knowledge of Workstation.
Data set used	AXDS-ITVobjects
Steps	1 The user opens the jobs panel where all current downloads are displayed 2 The user reads the remaining time for the end of transmission
Expected results	The remaining time for the end of the transmission is displayed in the job panel, and it corresponds to the real end of the transmission. The Downloading Bar of the receiving content is progressing.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

16.6 Content Reparation

TCId	TC16.6
Test case	Content Reparation
Initial conditions	The user, trying to open an AXMEDIS Object from the access panel of the Client Application Interface, receives a pop-up saying that some packets were lost during the multicast transmission. Simulate the bad reception of the Object: 1. Start downloading an AXMEDIS Object 2. Break the satellite connection for few seconds and the re-plug it.
Configuration description	An AXMEDIS Client Application up and running, and an Internet connection
Description of functionality to be tested	Unicast reparation by pull of an AXMEDIS Object received incomplete.
Partners, people involved	Content Consumer (user)
Validator(s) skill	Normal usage of Software Application.
Data set used	None
Steps	1 The user starts to repair the Object via unicast clicking on the specific icon.
Expected results	After reparation the AXMEDIS Object is complete and can be correctly played.
Variations	• The checksum is not matching with the one kept by the server (reparation is impossible). • The Object is not available on the server: reparation cannot be done. • The reparation is still possible by waiting the next multicast transmission (if it is scheduled).
Issues	The Internet Connection is used to repair the content.
Test case Scope/Type	GUI, Application Core Test, Backend / BlackBox

16.7 Content Access

TCId	TC16.7
Test case	Content Access
Initial conditions	The AXMEDIS Client Application is up and running. The cache of the AXMEDIS client application has successfully received some AXMEDIS Objects
Configuration	The AXMEDIS Client Application is the Cache-based Distribution on i-TV

description	(T4.8.2)
Description of functionality to be tested	The user access the AXMEDIS Objects in the cache and plays them or stores them in the hard disk, according to the DRM rules
Partners, people involved	Content Consumer (user)
Validator(s) skill	Skill of the people involved in the test during the validation with end-users
Data set used	AXDS-ITVobjects
Steps	1 The user opens and plays some AXMEDIS Objects which are in his/her local cache 2 The user stores in the hard disk some AXMEDIS Objects from his/her local cache 3 The AXMEDIS Client Application detects if the AXMEDIS Object needs to acquire a license 4 The AXMEDIS Client Application finds pre-acquired license for the Object and plays it
Expected results	The user can open, play and/or store AXMEDIS Objects present in the local cache.
Variations	None
Issues	Internet Connection required.
Test case Scope/Type	GUI / WhiteBox.

16.8 Content Preview

TCId	TC16.8
Test case	Content Preview
Initial conditions	The cache of the AXMEDIS client application has successfully received some AXMEDIS Objects
Configuration description	Some AXMEDIS Objects are already received. The AXMEDIS Client Application is up and running.
Description of functionality to be tested	The user browses the AXMEDIS Objects and plays their previews. The user decides to buy or not the received AXMEDIS Content.
Partners, people involved	Content Consumer (user)
Validator(s) skill	Normal usage of Software Applications
Data set used	AXDS-ITVobjects
Steps	1 The user opens the AXMEDIS Object locally stored in his local cache 2 The user browses the AXMEDIS Object, using the AXMEDIS Info associated to the Object 3 The user reaches a preview available for the Object 4 The user plays the AXMEDIS Object Preview
Expected results	The user can open, browse, reach, and play previews of the AXMEDIS Objects.
Variations	None
Issues	No Internet Connection required.
Test case Scope/Type	GUI / WhiteBox

16.9 License Acquisition

TCId	TC16.9
Test case	Licence Acquisition
Initial conditions	The user has received an AXMEDIS Object and can browse and preview it. The backend (AXCS) is up and running.
Configuration	An AXMEDIS Client Application up and running.

description	AXMEDIS Certifier running. Some AXMEDIS Objects received and previewed. The user station should support all secure protocols.
Description of functionality to be tested	The user tries to purchase a license for playing the protected part of an AXMEDIS Object received and previewed.
Partners, people involved	Content Consumer (user).
Validator(s) skill	Familiarity in playing multimedia content in computers.
Data set used	AXDS-ITVlicenses, AXDS-ITVlogin, AXDS-ITVpayments
Steps	1 A user tries to access to a protected AXMEDIS Object 2 The user identifies himself in the AXCS 3 The user chooses a type of license proposed by the AXCS for the given Object
Expected results	The user receives the AXMEDIS Authorization useful to open the protected part of the AXMEDIS Object The user can consumes the AXMEDIS Object following the rules in the AXMEDIS Authorization
Variations	The user abandons interaction with the AXCS.
Issues	None.
Test case Scope/Type	Application Core Test, Backend / WhiteBox

16.9.1 User Identification

TCId	TC16.9.1
Test case	User Identification
Initial conditions	The User has successfully performed the Registration procedure.
Configuration description	AXMEDIS Client Application up and running. AXMEDIS Certifier running
Description of functionality to be tested	The user will be requested to identify and provide credentials needed to ensure that the requested transaction (purchase/rental) is valid and legal.
Partners, people involved	Content Consumer (user)
Validator(s) skill	Familiarity with e-commerce transactions.
Data set used	AXDS-ITVlogin, AXDS-ITVcredentials
Steps	1 The user enters his identification information (this does not necessarily mean personal details, it will be sufficient to have proper credentials, e.g., login/password) 2 The user credentials are sent to the AXCS for verification 3 The user receives a response from the server
Expected results	If the user is identified as a regular one, permission to proceed is granted; otherwise purchase procedure is aborted and user is sent back to browsing
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / WhiteBox

16.9.2 Billing

TCId	TC16.9.2
Test case	Billing
Initial conditions	The User is Regular for the AXCS. The User has enough credits to perform the transaction. The AXCS.
Configuration	An active Internet connection.

description	The AXMEDIS Client up and running. Some AXMEDIS Objects already stocked in the local hard disk.
Description of functionality to be tested	The user confirms the intention to purchase the AXMEDIS Content. The user provides payment related information along with data needed to ensure legal validity of requested operation.
Partners, people involved	Content Consumer; the AXCS
Validator(s) skill	Confidence in e-commerce transactions.
Data set used	AXDS-ITVpayments
Steps	1 The AXCS shows to the user all billing information available including: · Price · Conditions for each selected item · Related use licence · Scope and limitations · Possible constraints 2 The user accepts license terms and the procedure continues otherwise is aborted and user is sent back to browsing 3 The user finalises billing information (using Data Set) 4 The user selects the payment method (credit card, electronic wallet, pre paid card, pre assigned tokens or similar) 5 The user waits for the backend verifications. 6 The payment ID is cleared and the user can verify that his prepaid credit is decreased. 7 The user receives the license and he can play the AXMEDIS Content.
Expected results	The user plays the AXMEDIS Content, if he has enough credits to purchase the content.
Variations	• The user has not enough credits to complete the purchase.
Issues	The user accesses to the service on a prepaid subscription basis.
Test case Scope/Type	GUI, Backend / WhiteBox

16.10 Content Backup

TCId	TC16.10
Test case	Content Backup
Initial conditions	Some AXMEDIS Contents have been received.
Configuration description	The AXMEDIS Client Application is up and running.
Description of functionality to be tested	The user copies some interesting content in a backup support (internal or external).
Partners, people involved	Content Consumer (user)
Validator(s) skill	Familiarity with Backup operations.
Data set used	AXDS-ITVobjects
Steps	1 The user opens the backup interface of the AXMEDIS Client Application 2 The user selects all Objects involved in the backup operation (see Data Set) 3 The user specifies the backup unit where the AXMEDIS Objects will be copied
Expected results	The AXMEDIS Objects are copied on the specified backup support.
Variations	None.
Issues	None.
Test case Scope/Type	GUI / WhiteBox

16.11 Content Restore

TCId	TC16.11
Test case	Content Restore

Initial conditions	The user has backuped some Objects on another support.
Configuration description	The AXMEDIS Client Application is up and running. Some AXMEDIS Objects are copied on a backup support.
Description of functionality to be tested	The user restores some previously backuped AXMEDIS Objects from a backup support (internal or external).
Partners, people involved	Content Consumer (user)
Validator(s) skill	Familiarity with backup operations.
Data set used	AXDS-ITVobjects
Steps	1 The user opens the restore interface of the AXMEDIS Client Application. 2 The user selects all the Objects involved in the restore operation. 3 The user specifies the support from which the AXMEDIS Objects should be restored.
Expected results	The AXMEDIS Objects are again available on the AXMEDIS Client Application.
Variations	• The backup support is not available then the test fails.
Issues	None.
Test case Scope/Type	GUI / WhiteBox

16.11.1 Cache Preloading

TCId	TC16.11.1
Test case	Cache Preloading
Initial conditions	The AXMEDIS Client Application up and running. The user has successfully registered to the AXMEDIS service and has received the required authorisation and a filter according his/her user profile. The cache is empty because it has never been used yet, or because the user has finished a “cache cleaning” procedure.
Configuration description	The AXMEDIS Client Application is either the “Cache-based Distribution on i-TV” or the “Cache-based Personalised Content Distribution”
Description of functionality to be tested	The user activates the cache loading functionality to have the cache filled with AXMEDIS objects, according to the user profile. The active filter allows only the a of AXMEDIS objects compatible with the user profile to pass through it
Partners, people involved	Content Consumer
Validator(s) skill	Familiarity with cache management.
Data set used	AXDS-ITVobjects, AXDS-ITVprofiles, AXDS-ITVprofile
Steps	1 The user runs the AXMEDIS Client Set Up Application. 2 The user activates the Cache Preloading functionality 3 The user wait for the cache to be filled with AXMEDIS Objects, according to his/her user profile.
Expected results	The Cache is successfully filled with AXMEDIS Objects. The AXMEDIS Objects in the cache are compatible with the User Profile.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

16.12 Cache Cleaning

TCId	TC16.12
Test case	Cache Cleaning
Initial conditions	The AXMEDIS Client Application up and running. The user has successfully registered to the AXMEDIS service and has received the required authorisation and a filter according his/her user profile. The cache is full with AXMEDIS Objects.

Configuration description	The AXMEDIS Client Application is either the “Cache-based Distribution on i-TV” or the “Cache-based Personalised Content Distribution”
Description of functionality to be tested	The user activates the cache cleaning functionality to empty the cache.
Partners, people involved	Content Consumer
Validator(s) skill	Familiarity with cache management.
Data set used	AXDS-ITVobjects
Steps	1 The user runs the AXMEDIS Client Set Up Application. 2 The user activates the Cache Cleaning functionality 3 The user wait for the cache to be emptied of AXMEDIS Objects
Expected results	The Cache is successfully emptied of the AXMEDIS Objects.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

16.13 Cache–based Personalised Content Distribution specific Test Cases

16.13.1 Automatic Content Access Set Up

TCId	TC16.13.1
Test case	Automatic Content Access Set Up
Initial conditions	The AXMEDIS Client Application up and running. The cache is already full of AXMEDIS Objects.
Configuration description	The AXMEDIS Client Application is the “Cache-based Personalised Content Distribution”
Description of functionality to be tested	The user activates the functionality which performs the AXMEDIS default Channels composition (composition of AXMEDIS Objects can be temporal and/or spatial).
Partners, people involved	Content Consumer
Validator(s) skill	Familiarity with computers and basic applications
Data set used	AXDS-ITVobjects, AXDS-ITVprofile
Steps	1 The user runs the AXMEDIS Client Set Up Application. 2 The user activates the AXMEDIS default Channel Composition functionality 3 The user waits for the AXMEDIS default channels to be composed.
Expected results	The AXMEDIS default channels are successfully composed and ready to be played.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

16.13.2 AXMEDIS Channel Personalisation

TCId	TC16.13.2
Test case	AXMEDIS Channel Personalisation
Initial conditions	The AXMEDIS default Channels have been composed.
Configuration description	The AXMEDIS Client Application is the “Cache-based Personalised Content Distribution”
Description of functionality to be tested	The user activates the functionality, which performs the personalised AXMEDIS Channels composition, according to the User Profile.
Partners, people	Content Consumer

involved	
Validator(s) skill	Familiarity with computer and basic applications
Data set used	AXDS-ITVobjects, AXDS-ITVprofile
Steps	1 The user runs the AXMEDIS Client Set Up Application. 2 The user activates the Personalised AXMEDIS Channel Composition functionality 3 The user wait for the personalised AXMEDIS channels to be composed.
Expected results	The AXMEDIS default channels are successfully composed and ready to be played by streaming. The AXMEDIS Channels are compatible with the User Profile.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

16.13.3 Automatic Content Access

TCId	TC16.13.3
Test case	Automatic Content Access
Initial conditions	The end user has already made the Automatic Content Access set up and the AXMEDIS Channels (either default or personalised) have been composed.
Configuration description	The AXMEDIS Client Application is the “Cache-based Personalised Content Distribution”
Description of functionality to be tested	The AXMEDIS Client Application can be swithced on and works fine The end user is able to select an AXMEDIS channel and to play it
Partners, people involved	Content Consumer (end user)
Validator(s) skill	Skill of the people involved in the test during the validation with end-users
Data set used	AXDS-ITVchannels
Steps	1 The end user switch on the Cache-based personalised content distribution Application 2 The end user selects some AXMEDIS Channel (some default AXMEDIS Channels and some personalised ones) 3 The end user plays some AXMEDIS Channels (some default AXMEDIS Channels and some personalised ones) 4 The end user switch off the Cache-based personalised content distribution Application
Expected results	The user can switch on the Cache-based personalised content distribution Application, select an AXMEDIS Channel , play it, change from an AXMEDIS Channel to another and switch off the Application.
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox.

16.13.4 AXMEDIS Channel PVR functionalities

TCId	TC16.13.4
Test case	Automatic Channel PVR functionalities
Initial conditions	Some AXMEDIS Channels have been composed (either the default or the personalised) and ready to play by streaming.
Configuration description	The AXMEDIS Client Application is the “Cache-based Personalised Content Distribution”
Description of functionality to be	The user activates the AXMEDIS Channel PVR functionalities which perform: stop

tested	play pause forward rewind record
Partners, people involved	Content Consumer
Validator(s) skill	Familiarity with the typical set of PVR operations
Data set used	AXDS-ITVchannels
Steps	1 The user runs the AXMEDIS Client Application 2 The user plays the AXMEDIS Channel. 3 The user stops the AXMEDIS Channel and then plays it again 4 The user pauses the AXMEDIS Channel and then plays it again 5 The user stops the AXMEDIS Channel and then activates the rewind functionality (which has a limited range of time after which the content could not be in the cache any more) 6 The user stops the AXMEDIS Channel and then activates the fast forward functionality 7 The user plays the AXMEDIS channel and then activates the rewind functionality (which has a limited range of time after which the content could not be in the cache any more) 8 The user plays the AXMEDIS channel and then activates the fast forward functionality 9 The user records some content by the AXMEDIS Channel 10 The user plays some content previously recorded
Expected results	All the PVR feature can be successfully performed
Variations	None
Issues	The rewind shall be limited in time: after a certain range of time (e.g. half an hour) the AXMEDIS Object could not be in the cache any more.
Test case Scope/Type	GUI / WhiteBox

17 AXMEDIS for Distribution to PDA via Kiosks (WP9.6: ILABS, DSI, EXITECH)

For the sake of clarity we would like to point out that in the description we have always reported as involved actors also those actors that are performing the monitoring of the system during test execution. This is expressed in the body of the test case as follows

Partners, people involved
1. The end user (performing the operation)
2. The Kiosk Manager (performing the execution monitoring)

In the test description when "user" is used with no other specification it is always the "end user" operating. When the "Kiosk Manager" is directly involved then the word user is not mentioned. It is given for granted that when the "end user" is operating and the "Kiosk Manager" is just monitoring, monitoring operation are nor reported in the test description in order to make the description shorter, more linear and simple to read. On the other hand as far as validator skills are concerned we have assumed that for the end user there are no special requirement (qualitative evaluation of results is expected) while for the monitoring personnel (nominally the Kiosk Manager) is necessary to have ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components. For operations that do refer to the publishing value chain the Kiosk Manager needs to have (in addition to previously mentioned skills also the typical skills of an editorial person.
As a last point is worth mentioning that when referring to "end user device" we intend any among a PDA a Smartphone (like QTEC, Treo, Motorola, Blackberry…) or a TabletPC.

17.1 Content Catalogue Creation

TCId	TC17.1
Test case	Content Catalogue Creation
Initial conditions	The local AXDBM, the AXEPTool and the AXMEDIS Content Production (Query support, Composition / Formatting Rules Editor, Composition / Formatting Engine) are available and properly functioning. The user has administrative rights and is able to operate with the specified tools and supporting components.
Configuration description	AXMEDIS Content Production: ❑ Query support, ❑ Composition Rules Editor, ❑ Formatting Rules Editor, ❑ Composition Engine ❑ Formatting Engine, ❑ Local AXDBM ❑ AXEPTool
Description of functionality to be tested	AXMEDIS Content Production: ❑ Query support, ❑ Composition Rules Editing, ❑ Formatting Rules Editing, ❑ Composition Rules application ❑ Formatting Rules application
Partners, people involved	The kiosk manager, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The kiosk manager is an editorial person with sufficient ICT skills to be able to perform implied operation and interpret system returned messages (both in case of positive or negative results). Has administrative rights and is able to operate with the specified tools and supporting components.

Data set used	AXDS-Kiosk1, AXDS-Kiosk2
Steps	1 The kiosk manager logs into the system. 2 The kiosk manager performs a query with the query user interface to retrieve the list of object suitable for being acquired and reported in the kiosk content catalogue: Select From AXEPTool (Where {Key1=XYZ, Key2=KKK…) 3 The query support system returns a AXOID list 4 The Kiosk manager browses the list and identifies the needed objects accessing to public metadata and preview samples stored in AXINFO for each AXOID of the received list. 5 The kiosk manager performs a new query to retrieve the wanted objects: Get From AXEPTool (Where {AXOID1=X, AXOID2=Y…) 6 The query support system returns a AXMEDIS objects list 7 The kiosk manager checks with the Composition Rules Editor presently available composition rules 7.1 If available rules are adequate proceeds to next step 7.2 If available rules need to be modified or are lacking then modifies / defines Composition Rules 8 The kiosk manager checks with the Formatting Rules Editor presently available formatting rules 8.1 If available rules are adequate proceeds to next step 8.2 If available rules need to be modified or are lacking then modifies / defines Formatting Rules 9 The kiosk manager activates the Composition Engine that creates a new AXMEDIS object starting from the collected objects 10 The kiosk manager activates the Formatting Engine that creates a new AXMEDIS object
Expected results	The kiosk manager is successfully logged into the system The queried AXEPTool returns a list of objects The kiosk manager is able to select & retrieve the AXMEDIS objects relevant to the generation of the kiosk catalogue The kiosk manager is able to check/generate the needed rules for composition & formatting The kiosk manager is able to properly format and compose retrieved AXMEDIS objects into a new one
Variations	• The procedure may be launched remotely from the kiosk management server
Issues	It is not clear yet if the login onto the system is limited to the local security (kiosk server or kiosk backend) or to AXMEDIS framework / AXEPTool
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.2 Content Catalogue Loading (Publication)

TCId	TC17.2
Test case	Content Catalogue Loading (publication)
Initial conditions	The local AXDB, the AXEPTool and the AXMEDIS Content Publication (Publication Rules Editor, Publication Engine, Client View Profiles, Distribution Server) are available and properly functioning. The user has administrative rights and is able to operate with the specified tools and supporting components.
Configuration description	AXMEDIS Publication environment: ❑ Publication Rules Editor, ❑ Publication Engine ❑ Formatting Engine, ❑ Client View Profiles, ❑ Distributor Server,

	❑ Local AXDB, ❑ AXEPTool
Description of functionality to be tested	AXMEDIS Content Publication: ❑ Composition Rules Editing, ❑ Formatting Rules Editing, ❑ Publication Rules application ❑ Client View Profiles manipulation & usage ❑ Distribution Server configuration
Partners, people involved	The kiosk manager, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The kiosk manager is an editorial person with sufficient ICT skills to be able to perform implied operation and interpret system returned messages (both in case of positive or negative results). Has administrative rights and is able to operate with the specified tools and supporting components.
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk9
Steps	1 The kiosk manager logs into the system. 2 The kiosk manager checks with the Publication Rules Editor presently available publication rules 2.1 If available rules are adequate proceeds to next step 2.2 If available rules need to be modified or are lacking then modifies / defines Publication Rules 3 The kiosk manager verifies clients view profiles and eventually updates them 4 The kiosk manager activates the Publication Engine that uses the Publication Rules defined in conjunction with the Client View Profiles to access the local AXDB (end eventually the Formatting Engine) to feed the Distributor server with the catalogue and top ten AXMEDIS object to be distributed
Expected results	The kiosk manager is successfully logged into the system The kiosk manager is able to check/generate the needed rules for publishing The kiosk manager is able to access and modify/exploit the Client View Profiles to condition the catalogue fruition by the user The kiosk manager is able to properly feed the Distributor Server to enable kiosk catalogue distribution / loading to kiosks
Variations	The procedure may be launched remotely from the kiosk management server
Issues	Backend and transfer functionalities should be available and in place. It is not clear yet if the login onto the system is limited to the local security (kiosk server or kiosk backend) or to AXMEDIS framework / AXEPTool
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.3 Content Catalogue Loading Update

TCId	TC17.3
Test case	Content Catalogue Loading Update
Initial conditions	The local Application Front End, the Local Management system, the local AXDBM, the AXEPTool and the AXMEDIS Distribution Server are available and properly functioning. The user has administrative rights and is able to operate with the specified tools and supporting components.
Configuration description	The kiosk manager accessing the kiosk starts the application that loads locally to the kiosk the presently available content catalogue and launches the content updating interface & procedures.
Description of functionality to be tested	Kiosk management: ❑ Application front end functioning, ❑ System management, ❑ System maintenance procedures functioning, ❑ Local storage management,

	❑ Local connectivity, ❑ Remote connectivity AXMEDIS Content Publication: ❑ Distribution Server functioning
Partners, people involved	The kiosk manager, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The kiosk manager is an editorial person with sufficient ICT skills to be able to perform implied operation and interpret system returned messages (both in case of positive or negative results). Has administrative rights and is able to operate with the specified tools and supporting components.
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk9
Steps	1 The kiosk manager logs into the system. 2 The kiosk manager perform a "switch to maintenance mode" for the target kiosk 3 The target kiosk system exits normal operational and enters in maintenance mode. 4 The application front-end loads all system maintenance application. 5 The kiosk manager launches the catalogue upload procedure which: 5.1 Contacts the Distribution Server requesting the send of the catalogue 5.2 Receives the catalogue 5.3 The procedure extracts from the catalogue the list of the top ten AXMEDIS objects 6 The system administrator activate the Content Top ten check procedure 6.1 The procedure retrieves the list o the top ten content 6.2 The procedure automatically removes from local storage all AXMEDIS objects out of the new top ten list 6.3 Contacts the Distribution Server requesting the send of the specified list of AXMEDIS objects 6.4 Receives the requested AXMEDIS objects 6.5 Once new top ten AXMEDIS objects are received locally the local storage is updated and the procedure ends 7 The system administrator launches the local system check procedure to verify if all is in order 8 The system administrator exits the maintenance mode
Expected results	The kiosk manager is successfully logged into the system The kiosk manager is able to start the "maintenance mode" The kiosk manager is able to activate maintenance procedures Maintenance procedure end up successfully Top ten AXMEDIS objects are loaded locally Kiosk Catalogue is loaded Kiosk applicative check list are successfully performed The kiosk manager is able to restore kiosk "normal" functionalities
Variations	The procedure may be launched remotely from the kiosk management server
Issues	Backend and transfer functionalities should be available and in place. It is not clear yet if the login onto the system is limited to the local security (kiosk server or kiosk backend) or to AXMEDIS framework / AXEPTool
Test case Scope/Type	GUI, Frontend, Backend /BlackBox

17.4 Kiosk Start-up

TCId	TC17.4
Test case	Kiosk start-up
Initial conditions	The local Application Front End, the Local Management system, the local AXDBM and the AXEPTool are available and properly functioning. The end user

	accessing the kiosk starts the application that loads locally to the kiosk the presently available content catalogue and launches the content browsing & previewing interface.
Configuration description	Kiosk: □ Local Application Front End, □ Local Management system, AXMEDIS Framework: □ Local AXDBM □ AXEPTool
Description of functionality to be tested	Kiosk management: □ Application front end functioning, □ System management, □ System maintenance procedures functioning, □ Local storage management, □ Local connectivity, □ Remote connectivity
Partners, people involved	□ The end user (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk10, AXDS-Kiosk11, AXDS-Kiosk12
Steps	1 The system at start-up shall load the application front-end. 2 The application front-end loads all application modules and performs a full system check encompassing: 2.1 Verify network connectivity 2.2 Verify backend availability 2.3 Verify local appliances functionality 3 Depending on system check results the system performs what follows: 3.1 The system is ready to be used or 3.2 Signalling out of service condition
Expected results	The local application should load the kiosk catalogue and allow the user to browse and navigate it.
Variations	None
Issues	e-commerce backend and transactional functionalities should be available and in place
Test case Scope/Type	GUI, Frontend, Backend /BlackBox

17.5 User registration to Kiosk

TCId	TC17.5
Test case	User registration to kiosk
Initial conditions	The end user is accessing the kiosk, but has never registered before to the AXMEDIS Framework; the user device (PDA or mobile) has not yet been identified.
Configuration description	Kiosk: □ Local Application Front End, □ Local Management system, □ Local authentication service AXMEDIS Framework: □ AXMEDIS Certifier & Supervisor
Description of	□ The kiosk authentication application

functionality to be tested	❑ The AXMEDIS Certifier & Supervisor (Registration Service)
Partners, people involved	❑ The end user (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk5, AXDS-Kiosk13
Steps	1 The system presents the user a registration form with the following data: 1.1 First Name [mandatory], 1.2 Last Name [mandatory], 1.3 Age [mandatory], 1.4 Address [mandatory]: 1.4.1 mail address, 1.4.2 phone, 1.4.3 mobile, 1.4.4 e-mail, 1.4.5 VAT code 1.4.6 … 1.5 Default Language [mandatory], 1.6 Preferred payment method: 1.6.1 pre-paid-cards, 1.6.2 credit card 1.6.3 … 1.7 Payment method: 1.7.1 card #, 1.7.2 validity from, 1.7.3 validity to, 1.7.4 type 1.7.5 … 1.8 Billing info: 1.8.1 mail address, 1.8.2 phone, 1.8.3 mobile, 1.8.4 e-mail, 1.8.5 VAT code 1.8.6 … 1.9 Preferred device: 1.9.1 PDA, 1.9.2 Smart phone, 1.9.3 Other 2 The user provides the required data 3 The user confirms input operation ending either pressing a button on the interface or any other widget. 4 The kiosk performs a check on data provided to verify completeness and correctness (as far as possible like for e-mail formats or number of digits for a VAT code or credit card…) 5 Depending on check results the system performs either operation: 5.1 Requires the user to re-input/correct data or add missing mandatory items 5.2 The kiosk presents the user a filled in form to require data confirmation or change 6 The user modifies or confirms provided data (in case modification apply steps

	1-5 have to be re-iterated) 7 The kiosk local application server properly formats the data and send a request to the AXMEDIS Registration Service 8 The kiosks prompts the user to wait for registration clearance 9 In case of success the AXMEDIS Registration Service sends back to the user final UID and password 10 The kiosks retrieves the registration clearance, informs the user of performed registration, stores provided UID and sends the confirmation e-mail to the user specified account 11 The kiosk system requests the user to log in 12 The user logs in 13 The application front end grants access to available services and presents the user a screen with the possible activities 13.1 Browse the catalogue 13.2 Modify own data 13.3 View support information 13.4 Logout
Expected results	The user should be registered The user should be assigned an AXMEDIS UID The system should be notified of the registration (via mail/sms) The user should be logged into the system
Variations	• The user is already registered and is simply performing a session registration or has changed device and therefore needs to download the application frontend on the new device (see User Login and Application Frontend installation)
Issues	In the kiosk scenario the case of a user registering for the 1st time has the major drawback that is not possible to provide the user with a direct access to his mail account to check the confirmation send back via mail. The usage of sms instead can be limited by environmental factors that are too risky to be left out.
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.6 User Login

TCId	TC17.6
Test case	User Login
Initial conditions	The end user is accessing the kiosk, but has already registered before to the AXMEDIS Framework.
Configuration description	Kiosk: □ Local Application Front End, □ Local Management system, □ Local authentication service AXMEDIS Framework: □ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	□ The kiosk authentication application □ The AXMEDIS Certifier & Supervisor (Registration Service)
Partners, people involved	□ The end user (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk5, AXDS-Kiosk13
Steps	1 The user interacts with the application front-end (selecting registration from

	language selection case) 2 The application front end invokes the local user login 3 The kiosk user management sends to the application front end the user data structure to be filled 4 The application front end asks the user to provide the login data (UID) 5 The user inserts the data and confirms. 6 Filled in data structure is sent back to the kiosk user management 7 The kiosk user management checks user information locally 8 The kiosk user management retrieves user related data via UID (in case the UID is not present the user will be requested to register) 9 The kiosk user management sends user data to the AXCS for verification (via AXCS web service interface) 10 The AXCS checks received info 11 The AXCS logs the registration event 12 The AXCS sends back to the kiosk user management a ACK 13 The kiosk user management confirms the login to the application front end 14 The application front end grants access to available services: application front-end presents the user a screen with the possible activities 14.1 Browse the catalogue 14.2 Modify own data 14.3 View support information 14.4 Logout 15 The user selects the desired activity
Expected results	The user should be registered The user should be logged into the system
Variations	• The user is not registered and therefore has to perform a full registration
Issues	In the kiosk scenario if something happens and the user is forced to log on anew on the system but has not yet accessed to the confirmation mail is necessary to use locally stored data to grant access if the initial registration procedure has been successful. Therefore the system will have to keep track of this and behave as previously specified.
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.7 Content Browsing & previewing

TCId	TC17.7
Test case	Content Browsing & Previewing
Initial conditions	The kiosk application is up and running, the back end is connected and functional and the end user uses the kiosk application to brows and preview the presently available content listed in the catalogue and eventually launches the selection process
Configuration description	Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB, ❑ AXEPTool, ❑ AXMEDIS viewers, ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk: ❑ Local Application Front End functioning, ❑ Local Management system functioning, AXMEDIS Framework: ❑ Local AXDB data provision,

	❑ AXMEDIS viewers rendering,
Partners, people involved	❑ The end user (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system presents the content list 2 The end user browses the list 3 The end user selects an item 4 The end user asks for content preview 5 Depending on content format a preview is presented as follows: 5.1 Brief description for text 5.2 Thumbnail for images 5.3 X sec sample for Audio (X will depend on IPR rules) 5.4 X sec sample for Video (X will depend on IPR rules) 5.5 X sec sample for Animations (X will depend on IPR rules) 5.6 X sec sample for Multimedia (X will depend on IPR rules) 6 The end users decides next step between: 6.1 Activate acquiring procedure 6.2 Returning to browsing
Expected results	The user can brose the catalogue The user can select objects The user can access to objects metadata (filtered by own profile) The user activate the acquisition procedure
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.8 Content Selection and Chart Management

TCId	TC17.8
Test case	Content Selection And Chart Management
Initial conditions	The end user has operated selections that condition the check out process
Configuration description	Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB, ❑ AXEPTool, ❑ AXMEDIS viewers, ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB data provision, ❑ AXMEDIS viewers rendering,
Partners, people involved	❑ The end user (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform

	checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The end user selects a specific content (it could be both in browsing or previewing mode) 2 The user requests content addition to the chart 3 The user requests to proceed either to check out or to continue browsing 4 Depending on previous step results the system enters one of the following to states: 4.1 Check out procedure activation 4.2 Browsing & previewing mode
Expected results	The selected content (AXOID) is added to the chart The user is able both to finalise the acquisition and to continue browsing & selecting
Variations	• In case of rental the chart can also be composed of a single item chart. Once the selection is operated the checkout procedure is automatically started in order to bring the user soon to fruition.
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.9 Check out procedure initiation

TCId	TC17.9
Test case	Check Out Procedure Initiation
Initial conditions	The chart holds selected AXOIDs and the user has requested to start a check out.
Configuration description	Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB, ❑ AXEPTool, ❑ AXMEDIS viewers, ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB data provision, ❑ AXMEDIS viewers rendering,
Partners, people involved	❑ The end user (initiating the operation) ❑ The system (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system enters protected mode 2 A secure connection is established with the certification authority
Expected results	The procedure is performed without any error detection
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend /BlackBox

17.10 Purchasing / Acquiring / Renting

TCId	TC17.10
Test case	Purchasing / Acquiring / Renting
Initial conditions	The user has initiated the check-out procedure
Configuration description	❑ Customer (involved in the purchase/rental operation), ❑ Kiosk Application Front End, ❑ Kiosk local management system, ❑ Kiosk billing management system ❑ AXMEDIS Certifier & Supervisor, ❑ Certification Authority (3rd trusted party like Verisign), ❑ A bank or other institution that will handle the money transaction
Description of functionality to be tested	Kiosk: ❑ Local Application Front End functioning, ❑ Local Management system functioning, ❑ Billing functioning, AXMEDIS Framework: ❑ Local AXDB data provision, ❑ AXEPTool data provision, ❑ AXMEDIS Certifier & supervisor functioning,
Partners, people involved	❑ The end user (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring) ❑ Certification Authority (3rd trusted party like VeriSign), ❑ A bank or other institution that will handle the money transaction
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system presents the customer billing information available (including price and conditions for each selected item, related use licence, scope and limitations, possible constraints…). Such data will be dependent on the UID and related profile. 2 The system asks the customer to verify and accept presented terms 3 If the customer accepts procedure continues otherwise is aborted and customer is sent back to browsing 4 Once accepted purchase/acquisition/renting conditions, the customer is requested to finalise billing information 5 The customer shall finalise billing information 6 Once billing information are provided the customer is requested to select the payment method (credit card, electronic wallet, pre paid card or similar) 7 The customer is requested to provide a valid ID for payment (credit card, electronic wallet, pre paid card or similar) 8 The Certification authority requires clearance to the third trusted party for the provided payment ID. 9 The thirds trusted party should provide clearance on payment ID (if this fails operation is aborted) 10 If payment ID is cleared the customer will charged the cost (including the third trusted party commission for service) 11 Certification authority provides the system the proper clearance and the delivery process can start.
Expected results	The transaction is properly performed and no errors are detected. Billing and payment info are collected and secularly stored for subsequent processing (notification to user and AXMEDIS Certifier & Supervisor)

Variations	• Initially (during internal test phases) no bank or Certification authority will be involved. Only once the system will be stable enough and the usage will be open to external actors these components of the value chain will be inserted
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.11 Repository Selection

TCId	TC17.11
Test case	Repository Selection
Initial conditions	The local Application Front End, the Local Management system, the local storage and AXDBM, the Query support, the AXEPTool and the AXMEDIS Certifier & Supervisor are fully operational and connected.
Configuration description	Kiosk: ❑ Local Application Front End, ❑ Local Management system, ❑ Local storage, AXMEDIS Framework: ❑ Local AXDBM, ❑ AXEPTool, ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk management: ❑ Application front end functioning, ❑ System management, ❑ Local storage management, ❑ Local connectivity, ❑ Remote connectivity AXMEDIS framework: ❑ Query support, ❑ Distribution Server functioning
Partners, people involved	❑ The end user (waiting for the operation to end) ❑ The system (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system checks each selected item for local / remote availability 2 In case of remote availability a secure channel is established 3 The system interacts via Query support wit the Kiosk AXDBM to check for data availability 3.1 If the AXMEDIS object is available it is sent to the kiosk for local storage and delivery to customer (once check out is performed) 3.2 If data is not available a request is placed to the Query support to retrieve it via AXEPTool 3.2.1 If the AXMEDIS object is available it is sent to the kiosk for local storage and delivery to customer (once check out is performed) 3.2.2 If the AXMEDIS object is unavailable an error message is generated, the check out is interrupted and a roll-back procedure is started
Expected results	The AXMEDIS object is located and locally cashed for subsequent delivery
Variations	None

Issues	If the content originally identified for being part of the catalogue is removed during catalogue lifetime an error code should be generated, the user should be informed, the check out interrupted and a rollback initiated. This should not happen yet if the kiosk catalogue has an average lifetime longer than the average lifetime / availability of AXMEDIS objects on the AXEPTool has to be taken into account
Test case Scope/Type	Backend / BlackBox

17.12 Destination Target Identification (Unique ID for Target – WIFI)

TCId	TC17.12
Test case	Destination Target Identification (Unique Id For Target – WiFi)
Initial conditions	The user has started the check out procedure (two similar system totally undistinguishable by the end user should be used in different trials to prove that identification is unique)
Configuration description	The local system and the PDA / mobile device
Description of functionality to be tested	Kiosk management: ❑ Application front end functioning, ❑ System management, AXMEDIS Framework: ❑ Device fingerprinting / identification ❑ AXMEDIS Certifier & Supervisor
Partners, people involved	❑ The end user (waiting for the operation to end) ❑ The system (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	None
Steps	1 The system identifies the end-user device and extracts a unique ID
Expected results	The user fruition system is properly identified
Variations	• In case a cross channel delivery is requested it will be necessary to access to additional info in order to properly secure delivery and fruition process on a device that is not available at transaction time
Issues	In case the user device is not capable to provide a unique identifier corresponding to some OS fingerprinting function or HW serialisation… we suggest using MAC addresses of the WiFi device as a fingerprint to be used in the license generation. In case both solutions are not pursuable and no unique ID can be provided by/ retrieved from the end-user device some other solution should be found to prevent that the delivered object can be improperly used/copied/duplicated/re-distributed once delivered
Test case Scope/Type	Backend / BlackBox

17.13 Delivery Template Selection (Depending on Device)

TCId	TC17.13
Test case	Delivery Template Selection (Depending On Device)
Initial conditions	The user has started the check out procedure and the delivery device has been identified
Configuration description	The local system, the end user fruition device, the AXMEDIS Formatting Engine, the AXMEDIS Composition Engine and the AXMEDIS Certifier & Supervisor
Description of	Kiosk management:

functionality to be tested	□ Application front end functioning, □ System management, □ Remote connectivity AXMEDIS Framework: □ Local AXDBM, □ AXEPTool, □ Composition Rules application, □ Formatting Rules application, □ DRM rule respect, □ AXMEDIS Certifier & Supervisor
Partners, people involved	□ The end user (waiting for the operation to end) □ The system (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk6, AXDS-Kiosk7, AXDS-Kiosk8, AXDS-Kiosk12
Steps	1 The system identifies the class of delivery device 2 The systems verifies the DRM rules related to the selected object in correlation to the end user device, kind of required usage, kind of available license composition and formatting rules applicable. 3 The system selects the template to be used for delivery (set of Composition Rules to be applied) to be applied by the Composition Engine to the selected AXMEDIS object 4 The system creates the delivery format by grouping a set of Formatting Rules to be applied by the Formatting Engine to the selected AXMEDIS object 5 The system requests to the Composition Engine to apply the required rules in order to achieve the needed delivery template 6 The system starts the preliminary checks necessary to ensure proper delivery
Expected results	The system is able to verify if the selected AXMEDIS object is available in a format suitable for delivery; which are the composition & formatting rules to apply. Foreseen composition rules are applied and formatting rules grouped for execution prior to delivery. DRM and composition rules for are properly respected, formatting rules are properly selected.
Variations	• In case a cross channel delivery is requested it will be necessary to access to additional info in order to properly secure delivery and fruition process on a device that is not available at transaction time
Issues	None
Test case Scope/Type	Backend / BlackBox

17.14 Delivery Format Selection (Depending on content)

TCId	TC17.14
Test case	Delivery Format Selection (Depending On Content)
Initial conditions	The user has started the check out procedure and previous preliminary checks are positive
Configuration description	The local system, the end user fruition device, the AXMEDIS Formatting Engine and the AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk management: □ Application front end functioning, □ System management, □ Remote connectivity

	AXMEDIS Framework: □ Local AXDBM, □ AXEPTool, □ Formatting Rules application, □ DRM rule respect, □ AXMEDIS Certifier & Supervisor
Partners, people involved	□ The end user (waiting for the operation to end) □ The system (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk6, AXDS-Kiosk7, AXDS-Kiosk8, AXDS-Kiosk12
Steps	1 Based on the end-user device identification and delivery template the system selects the delivery format (set of Formatting Rules) to be applied by the Formatting Engine to the selected AXMEDIS object 2 The system verifies if the required formatting rules can be applied on the selected AXMEDIS object according to DRM rules, user profile and requested operation (fruition, purchase, rent…) 3 The system requests to the Formatting Engine to apply the required rules in order to achieve the needed delivery format 4 The system starts the preliminary checks necessary to ensure proper delivery
Expected results	The formatting rules grouped in the previous step are applied and the selected AXMEDIS object is formatted for the delivery to end user according to required kind of device. DRM and formatting rules are properly applied.
Variations	• In case a cross channel delivery is requested it will be necessary to ensure that safe delivery conditions can be met
Issues	None
Test case Scope/Type	Backend / BlackBox

17.15 Device Compatibility (Roll Back in Case of Failure)

TCId	TC17.15
Test case	Device Compatibility (Roll Back In Case Of Failure)
Initial conditions	The user has started the check out procedure and previous preliminary checks are positive
Configuration description	The local system, the end user fruition device
Description of functionality to be tested	Kiosk management: □ Application front end functioning, □ System management, □ Local storage management, □ Local connectivity, □ Remote connectivity AXMEDIS Framework: □ Local AXDBM, □ AXEPTool, □ AXMEDIS Certifier & Supervisor
Partners, people involved	□ The end user (waiting for the operation to end) □ The system (performing the operation) □ The Kiosk Manager (performing the execution monitoring)

Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 Given the combination of selected content, fruition device, delivery template and format the systems performs a final consistency check 2 According to check results the system proceeds either: 2.1 In the delivery process or 2.2 Performs a roll back request (including billing cancelling and money refund)
Expected results	The user device is compatible with the selected AXMEDIS object and required usage
Variations	• In case a cross channel delivery is requested it will be necessary to ensure that safe delivery conditions can be met. A customer feedback will be necessary within 8 day from transaction execution date. Timestamps for this should be provided to the customer and stored both locally and at the AXMEDIS certification authority
Issues	None
Test case Scope/Type	Backend / BlackBox

17.16 Storage Availability (Roll Back in Case of Failure)

TCId	TC17.16
Test case	Storage Availability (Roll Back In Case Of Failure)
Initial conditions	The user has started the check out procedure and previous preliminary checks are positive
Configuration description	The local system, the end user fruition device
Description of functionality to be tested	Kiosk management: □ Application front end functioning, □ System management, □ Local storage management, □ Local connectivity, □ Remote connectivity
Partners, people involved	□ The end user (waiting for the operation to end) □ The system (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system checks if the end-fruition device has enough storage to host the selected content 2 According to check results the system proceeds either: 2.1 In the delivery process or 2.2 Performs a roll back request (including billing cancelling and money refund)
Expected results	The user has enough storage to host the selected AXMEDIS object
Variations	In case a cross channel delivery is requested it will be necessary to ensure that safe delivery conditions can be met
Issues	None

Test case Scope/Type	Backend / BlackBox

17.17 Billing

TCId	TC17.17
Test case	Billing
Initial conditions	The local Application Front End, the Local Management system, the local storage and AXDBM, the Query support, the AXEPTool and the AXMEDIS Certifier & Supervisor are fully operational and connected. All preliminary checks have been positively performed (no roll-back)
Configuration description	End user fruition device Kiosk: ❑ Local Application Front End, ❑ Local Management system, ❑ Local storage, AXMEDIS Framework: ❑ Local AXDBM, ❑ AXEPTool, ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk management: ❑ Application front end functioning, ❑ System management, ❑ E-payment transaction handling, ❑ Billing management, ❑ Remote connectivity AXMEDIS framework: ❑ AXMEDIS Certifier & Supervisor functioning
Partners, people involved	❑ The end user (performing the operation) The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system formalises the economic transaction into a proper bill 2 The system sends the billing info to the end-user (according to provided billing info) 3 The system sends the billing info to the AXMEDIS certification authority for the required subsequent processing steps
Expected results	The kiosk local managements handles the economic transaction with no errors The corresponding billing report is produced and given both to the end user and to the AXMEDIS Certifier & Supervisor No errors are detected
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.18 Data Delivery

TCId	TC17.18
Test case	Data Delivery
Initial conditions	The billing phase is closed positively. The user device is wireless connected and the front-end application is properly running on it.
Configuration	Customer device

description	Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB, ❑ AXMEDIS viewers, ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk: ❑ Local Application Front End, ❑ Local Management system, ❑ Local storage data provision, AXMEDIS Framework: ❑ Local AXDB data provision, ❑ AXMEDIS viewers rendering,
Partners, people involved	❑ The end user (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system requires the customer to initiate the content download 2 The customer selects the final storage target destination (if possible) 3 The customer activates the download procedure
Expected results	The AXMEDIS viewer is successfully uploaded, The selected AXMEDIS object download, No errors are detected
Variations	• The AXMEDIS viewer is not pushed to the PDA/ smartphone/tabletPC as it is already present and up to date having being loaded during the 1st registration step
Issues	None
Test case Scope/Type	GUI, Frontend, Backend /BlackBox

17.19 Check out procedure closure

TCId	TC17.19
Test case	Check Out Procedure Closure
Initial conditions	All steps related to delivery preliminary checks have been successfully completed and delivery has started.
Configuration description	Customer device Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB, ❑ AXEPTool, ❑ AXMEDIS viewers, ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk: ❑ Local Application Front End, ❑ Local Management system, AXMEDIS Framework: ❑ Local AXDB data provision, ❑ AXMEDIS viewers rendering,

Partners, people involved	□ The end user (waiting the operation to end) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The system notifies the customer that the checkout procedure has been terminated 2 The secure connection with the certification authority is released 3 The system exits protected mode
Expected results	The system returns to normal operation while the delivery procedure is proceeding.
Variations	None
Issues	In this case we are demanding to a separate step the eventuality of delivery failure as the process may have a sensible difference in duration depending on factors like object size, available bandwidth…
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.20 Successful Delivery Check (Recovery in Case of Failure)

TCId	TC19.20
Test case	Successful Delivery Check (Recovery In Case Of Failure)
Initial conditions	The check out procedure has been closed and delivery is in process
Configuration description	Customer device Kiosk: □ Local Application Front End, □ Local Management system, AXMEDIS Framework: □ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	Kiosk: □ Local Application Front End, □ Local Management system, AXMEDIS Framework: □ Local AXDB data provision, □ AXMEDIS viewers rendering,
Partners, people involved	□ The end user (waiting the operation to end) □ The system (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The local system should monitor the download process to ensure a smooth delivery in case of problems the system should: 2 Perform at least 3 retries 3 Inform the customer of the incurred problem 4 Ask the customer which choice is preferred among: 4.1 New set of delivery retry 4.2 Deferred delivery 4.3 Delivery cancel 5 The system should take note of customer decision and consequently proceed to:

	5.1 Activate a new set of delivery retry (maximum 3) 5.2 Deferred delivery 5.2.1 Ask the customer the time of next delivery 5.2.2 Schedule next delivery 5.2.3 Flag the process for possible cancellation & refund 5.3 Delivery cancel 5.3.1 Enter secure mode 5.3.2 Establish a secure connection with the AXMEDIS certification authority 5.3.3 Performs a roll back request (including billing cancelling and money refund) 5.3.4 The system notifies the customer that the delivery and related transaction has been annulated 5.3.5 The system notifies the customer that refund procedure has been activated 5.3.6 The secure connection with the certification authority is released 5.3.7 The system exits protected mode 6 The system goes back to normal operation mode allowing the customer to browse and select content
Expected results	Content is actually downloaded onto the fruition device and no error is reported
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

17.21 Content Fruition after Download on PDA

TCId	TC17.21
Test case	Content fruition after download on PDA
Initial conditions	The kiosk has successfully uploaded the required viewer onto the fruition device and the user has successfully loaded the object .
Configuration description	The local system, the AXOM, the domain PMS, the AXMEDIS Certification Supervisor, the end user and the fruition device
Description of functionality to be tested	□ AXMEDIS viewer managing of selected object via domain AXOM □ AXMEDIS viewer managing of selected object license via domain PMS □ Domain PMS managing of license via AXMEDIS Certifier & Supervisor □ AXMEDIS Certifier & Supervisor key generation in relation to selected object DRM & license. □ Domain PMS managing of keys generated via AXMEDIS Certifier & Supervisor □ AXMEDIS viewer rendering of selected object based on available license and provided keys
Partners, people involved	□ The end user (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator(s) skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk4
Steps	1 The user requests access to the downloaded content 2 The local viewer gets the license from the governed object 3 The local viewer gets the AXOID from the governed object 4 The local viewer gets the UID 5 The local viewer gets the device ID 6 The local viewer requires the domain PMS (via AXOM) the consistency of

	the required operation for the specified AXOID by the UID on the specific device with the given licence 7 The viewer informs the user of being performing a licensing check and enters a wait state for either the keys or a NACK 8 The domain PMS requires to the AXMEDIS Certification Supervisor to perform the check and if positive generate the related user keys 9 The domain PMS waits for the either the keys or a NACK 10 The AXMEDIS Certification Supervisor performs a license check on the basis of the requested usage, identified object, device and UID and decides whether the operation is feasible or not. According to check results it either: 10.1 Sends back to the requesting domain PMS needed user keys (in case of positive result) 10.2 Sends back to the requesting domain PMS a NACK 11 The domain PMS receives the reply and forwards it to the requesting viewer (via AXOM) 12 Depending on check results the viewer proceeds as follows: 12.1 Allows content fruition 12.2 Blocks content fruition
Expected results	Content is actually downloaded onto the fruition device Proper keys for content rendering are provided The end user can use the content according to the agreed license model and DRM fruition rules No error is reported
Variations	• Cross channel fruition will be specified in a second time as it has implications on how to identify the remote device and provide the key / activation code… as already highlighted in the "Destination Target Identification" case
Issues	This is a critical step as much of its feasibility depends on what the end-user device OS and computational ability are able to provide as working infrastructure
Test case Scope/Type	GUI, Backend /BlackBox

17.22 Application Frontend Installation on End User Device

TCId	TC17.22
Test case	AXMEDIS viewer installation on end user device
Initial conditions	The end user is logged-on and using the kiosk to install the Application Frontend (including the AXMEDIS viewer). The kiosk application is up and running, the back end is connected and functional.
Configuration description	Kiosk: ❑ Local Application Front End, ❑ Local Management system, ❑ Local authentication service AXMEDIS Framework: ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	❑ The kiosk authentication application ❑ The AXMEDIS Certifier & Supervisor (Registration Service)
Partners, people involved	❑ The end user (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk5, AXDS-Kiosk13

Steps	1 The user selects in the Content List the option “Other Services” 2 The system presents the user a menu with the following options: 2.1 Change profile 2.2 Application Frontend install 3 The user selects on the kiosk GUI the option “Application Frontend install” 4 The system presents instructions for installation and how to connect wirelessly to the kiosk 5 The system asks the user to connect via web to a specific URL from his device 6 The user accesses the specified URL 7 The page loaded initiates the download of the application & viewer 8 The system informs the user of download results.
Expected results	The viewer is properly installed
Variations	• The user is not registered and therefore has to perform a full registration
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.23 User Profile Change

TCId	TC17.23
Test case	User Profile Change
Initial conditions	The end user is logged-on and using the kiosk to change own profile. The kiosk application is up and running, the back end is connected and functional.
Configuration description	Kiosk: ❑ Local Application Front End, ❑ Local Management system, ❑ Local authentication service AXMEDIS Framework: ❑ AXMEDIS Certifier & Supervisor
Description of functionality to be tested	❑ The kiosk authentication application ❑ The AXMEDIS Certifier & Supervisor (Registration Service)
Partners, people involved	❑ The end user (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator skill	❑ The end user (no special requirement – qualitative evaluation of results is expected) ❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk5, AXDS-Kiosk13
Steps	1 The application front end has granted access to available services including: 1.1 Browse the catalogue 1.2 Modify own data 1.3 View support information 1.4 Logout 2 The user selects in the Content List the option “Modify own data” 3 The system presents the user the profile form with the following data: 3.1 First Name [mandatory], 3.2 Last Name [mandatory], 3.3 Age [mandatory], 3.4 Address [mandatory]: 3.4.1 mail address, 3.4.2 phone, 3.4.3 mobile, 3.4.4 e-mail,

	3.4.5 VAT code 3.4.6 … 3.5 Default Language [mandatory], 3.6 Preferred payment method: 3.6.1 pre-paid-cards, 3.6.2 credit card 3.6.3 … 3.7 Payment method: 3.7.1 card #, 3.7.2 validity from, 3.7.3 validity to, 3.7.4 type 3.7.5 … 3.8 Billing info: 3.8.1 mail address, 3.8.2 phone, 3.8.3 mobile, 3.8.4 e-mail, 3.8.5 VAT code 3.8.6 … 3.9 Preferred device: 3.9.1 PDA, 3.9.2 Smartphone, 3.9.3 Other 4 The user provides the required data 5 The user confirms input operation ending either pressing a button on the interface or any other widget.
Expected results	The user should be registered The user should be assigned an AXMEDIS UID The system should be notified of the registration (via mail/sms) The user should be logged into the system
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.24 Interface Language Selection

TCId	TC17.24
Test case	Interface Language selection
Initial conditions	The end user is selecting the application front-end interface language. The kiosk application is up and running.
Configuration description	Kiosk: □ Local Application Front End,
Description of functionality to be tested	□ Application front end adaptability (in terms of language)
Partners, people involved	□ The end user (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	None
Steps	1 The application front end exits from idle mode when a user interacts

	2 The application front-end presents the user a screen with the flags of the supported languages for the GUI 3 The user presses the selected language 4 The application front end sets-up the environment variable stating the GUI language 5 The application front end presents the user a page for log-in / registration
Expected results	The front-end application is operational in the selected language
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend / BlackBox

17.25 User Device Configuration

TCId	TC17.25
Test case	Interface Language selection
Initial conditions	The end user is configuring the application front end on own device. The kiosk application is up and running.
Configuration description	Kiosk: □ Local Application Front End, □ WiFi connection User device □ Operating system □ WiFi Connection
Description of functionality to be tested	□ Application front end functionality □ WiFi connectivity
Partners, people involved	□ The end user (performing the operation) □ The Kiosk Manager (performing the execution monitoring)
Validator skill	□ The end user (no special requirement – qualitative evaluation of results is expected) □ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk10
Steps	1 The user has access to a page (either printed or in push) with the following info: 1.1 How to connect the PDA / Tablet to the kiosk via WiFi (including how to test the connection) 1.2 How to download the Application client on the device (including how to test the client) 2 The user performs on the device the required operation to configure the WiFi connection 3 The user performs the suggested check to ensure that WiFi configuration is successful 4 Device connects to the kiosk application front end 5 The application front end returns a test display page 6 The user performs on the device the required operation to download the application client (following a specific URL returned in the previously provided test page) 7 The device downloads the application client 8 The user install the downloaded client 9 The user performs the suggested check to ensure that application client install is successful 10 Device connects to the kiosk application front end 11 The application front end returns a test display object and a link to bookmark

	for future access via device 12 The application client displays the test object 13 The application client bookmarks the provided URL to access via device 14 The installed AXMEDIS client connects to the domain PMS to perform the requested "Registration" & "Authentication" as described in overall scenarios V3.9 (slide 219-220)
Expected results	The end user device is properly configured and operational
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend / BlackBox

17.26 Content Update (Via Satellite)

TCId	TC17.26
Test case	Content Update (via Satellite)
Initial conditions	The end user is selecting the application front-end interface language. The kiosk application is up and running.
Configuration description	Kiosk: ❑ Local Application Front End, ❑ Kiosk Data Manager ❑ Local AXDB AXMEDIS framework ❑ AXMEDIS B2B Satellite Reception Listener ❑ AXMEDIS B2B Satellite Reception Content Checker ❑ AXMEDIS Action Manager
Description of functionality to be tested	❑ The kiosk front end ❑ The satellite connectivity ❑ The kiosk maintenance functionalities
Partners, people involved	❑ The kiosk (performing the operation) ❑ The Kiosk Manager (performing the execution monitoring)
Validator skill	❑ The Kiosk Manager (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk7, AXDS-Kiosk8, AXDS-Kiosk9, AXDS-Kiosk10, AXDS-Kiosk11, AXDS-Kiosk12, AXDS-Kiosk13
Steps	1 The checking time is over a Down-Link channel check has to be performed 2 The AXMEDIS B2B Satellite Reception Listener checks for data availability and behaves as follows: 2.1 Data is not available yet so a further check is scheduled and the application enters wait mode (cycling back to point 1) 2.2 Data is available therefore is downloaded (2.b.1) and progressively cached locally (2.b.2) 2.3 Received data is stored locally 3 The AXMEDIS B2B Satellite Reception Listener activates the AXMEDIS Action Manager to decide how to proceed 4 The AXMEDIS Action Manager invokes the AXMEDIS B2B Satellite Reception Content Checker to verify consistency check on received data 5 The AXMEDIS B2B Satellite Reception Content Checker proceeds as follows 5.1 Performs consistency check on received data 5.2 If result is positive returns ACK and control to the AXMEDIS Action Manager 5.3 If result is negative requires the distribution server to resend the damaged packages via Up-link as detailed here after: 5.3.1 Satellite Reception Content Checker requires missing or

	damaged packages via Up-Link 5.3.2 Satellite Reception Content Checker receives missing or damaged packages via Up-Link 5.3.3 Satellite Reception Content Checker returns ACK and control to the AXMEDIS Action Manager 6 The AXMEDIS Action Manager retrieves the data from the local storage 7 The AXMEDIS Action Manager extracts the content form the OpenSky package 8 The AXMEDIS Action Manager checks the received data to determine what it is and behaves consequently: 8.1 Received data are AXMEDIS Object: data is stored in the AXDB 8.2 Received data are system / application updates: invoke the kiosk data manager to store data locally according to needs 8.2.1 The kiosk data manager stores the received data locally in plain format
Expected results	The kiosk content is correctly updated.
Variations	None
Issues	None
Test case Scope/Type	Frontend / BlackBox

18 AXMEDIS Content Description: Data sets for test and validation

This section describes the data set used and referenced into the data set row of test cases.

18.1 AXDS-DB1

A group of at least 20 different AXMEDIS objects with at least version 1.0 and 1.1. These objects can include different type of contents: no limitations to that.

18.2 AXDS-DB2

A group of at least 100 different AXMEDIS objects with no limitation in the contents for which is known the result to a set of predefined queries.

18.3 AXDS-Editor1

An AXMEDIS Object.

18.4 AXDS-Editor2

Composite AXMEDIS Object containing several components, the grant to enrich that object.

18.5 AXDS-Editor3

Composite AXMEDIS object containing at least a resource, the grant to export that resource.

18.6 AXDS-Editor4

An AXMEDIS object containing at least one element.

18.7 AXDS-Editor5

Composite AXMEDIS object.

18.8 AXDS-Editor6

An AXMEDIS Object contains at least a resource, the grant to someway modify that resource.

18.9 AXDS-IVE1

An AXMEDIS Object containing at least one element, the grant to manage or someway modify that resource.

18.10 AXDS-IVE2

An AXMEDIS Object containing at least one resource, the grant to manage or someway modify the resource.

18.11 AXDS-Composition1

Set of multi-media objects for distribution and production from database, different set of compositional rules and Selections of relevant objects.

18.12 AXDS-Composition2

Set of multi-media objects for distribution and production from database, set of rules in the repository collection.

18.13 AXDS-Composition3

Collection of Composition rules.

18.14 AXDS-Composition4

Collection of active Composition rules.

18.15 AXDS-Formatting1

Formatting rules set to active.

18.16 AXDS-Formatting2

Set of multi-media objects for distribution and production from database, set of rules in the repository collection.

18.17 AXDS-Formatting3

Collection of formatting rules

18.18 AXDS-Formatting4

Collection of active formatting rules.

18.19 AXDS-Workflow1

An NPD process containing at least one component

18.20 AXDS-Workflow2

An NPD process containing at least two components

18.21 AXDS-Workflow3

An NPD process containing at least one component plus CPA data

18.22 AXDS-Workflow4

A set of NPDs with tasks assigned to at least one user

18.23 AXDS-CMS

A selection of content (more than 100) that represents the typical content distributed within AXMEDIS (media types: audio, images, video and (text) documents) with related metadata, coming from different CMSs.

18.24 AXDS-AXEPPR

The rules to apply for the automatic publication of AXMEDIS objects.

18.25 AXDS-AXEPLR

The rules to apply for the automatic loading of AXMEDIS objects.

18.26 AXDS-AXEPAS

An active selection of AXMEDIS objects.

18.27 AXDS-AXEPP2Pheaders

The headers involved in the P2P handshaking.

18.28 AXDS-AXEPQH

A set of query Hits.

18.29 AXDS-P&P1

A set of 10 or more selected test AXMEDIS objects from the AXMEDIS database.

18.30 AXDS-P&P2

A set of 3 sample Set of rules in the repository collectionset of rules (i.e. 3 programmes) in the test repository.

18.31 AXDS-P&P3

A selection of content (about 100) that represents the typical content distributed within AXMEDIS (media types: audio, images, video and (text) documents) with related metadata. A set of predefined queries (about 5 for each media type and distribution channel) addressing the scope of the selected/provided content.

18.32 AXDS-PTE1

A set of (unprotected) multi-media objects (about 20) ready for protection. A set of corresponding rules specification/description addressing the different protection options available within AXMEDIS. Each option must be covered by at least one specified rule.

18.33 AXDS-PTE2

A set of (unprotected) multi-media objects (about 20) ready for protection. A set of corresponding (created) rules addressing the different protection options available within AXMEDIS. Each option must be covered by at least one specified rule.

18.34 AXDS-PIE1

A set of protected and un-protected multi-media objects (about 20) ready for protection.

18.35 AXDS-ITV1

List of transponders frequencies, set of filenames, extract of log lines to be checked in a successful execution of B2B AXMEDIS Client.

18.36 AXDS-ITV2

List of files to be configured, parameters to be changed, modules to be installed in the OS.

18.37 AXDS-ITV3

Set of information needed to check the correct reception of content: location of the storage area where the received content is stored, location of log files tracing the download and decryption of content, examples of regular traced messages.

18.38 AXDS-ITV4

List of test packages to be visible in the Electronic Program Guide, list containing the position, form of the icons to be visible after installation.

18.39 AXDS-ITV5

Spectrum Image of the transponder to be pointed, list of Transponder frequencies to be tested, list of PIDs to be visible in the locked frequencies.

18.40 AXDS-ITVlogin

Login to access to the Opensky database, to be used in test cases.

18.41 AXDS-ITVloginB

Login to access to the Opensky database as a Distributor, to be used in test cases.

18.42 AXDS-ITVpreferences

Set of Authorizations and Filters associated to a test login.

18.43 AXDS-ITVprofile

Set of User profiles to be used in test cases.

18.44 AXDS-ITVobjects

Set of AXMEDIS Objects, some of which are enctrypted, that can be scheduled for transmission, updated, downloaded on the client side.

18.45 AXDS-ITVpackages

List of packages, some of which are surely present in the Carousel List, others coming from the application of a test user profile, others matching of some key words expressly provided.

18.46 AXDS-ITVstations

Set of B2B receiving stations and related data, such as whether they are controlled or enabled to receive AXMEDIS Objects or Updates.

18.47 AXDS-ITVschedule

Schedule parameters used to define transmission of Objects via Push system.

18.48 AXDS-ITVcredentials

List of credentials.

18.49 AXDS-ITVlicences

Set of licencesID.

18.50 AXDS-ITVpayments

Accepted form of payment to acquire licences.

18.51 AXDS-ITVchannels

Set of AXMEDIS channels made of AXMEDIS objects composition.

18.52 AXDS-AXCS1

A group of at least 30 representative AXCS related data (see pertinent database structure and Use Case). The included data should be the most various as possible, containing a 30% of null, inconsistent, duplicated and data-type critical data (i.e. for example 0 for numeric data type to be tested in mathematic division operand).

18.53 AXDS-AXCS2

A group of at least 30 representative Tool/Device related data (see pertinent database structure and Use Case). The included data should be the most various as possible, containing a 30% of null, inconsistent, duplicated and data-type critical data (i.e. for example 0 for numeric data type to be tested in mathematic division operand).

18.54 AXDS-AXCS3

A group of at least 30 representative Objects related data (see pertinent database structure and Use Case). The included data should be the most various as possible.

18.55 AXDS-AXCS4

A group of at least 30 representative Object related data (see pertinent database structure and Use Case). The included data should be the most various as possible, containing a 30% of null, inconsistent, duplicated and data-type critical data (i.e. for example 0 for numeric data type to be tested in mathematic division operand). Note that at least a 50% of data included in the present data set should be also included in AXDS-AXCS3.

18.56 AXDS-AXCS5

A group of at least 30 representative Object usage related data (see pertinent database structure and Use Case). The included data should be the most various as possible, containing a 30% of null, inconsistent, duplicated and data-type critical data (i.e. for example 0 for numeric data type to be tested in mathematic division operand). Note that at least a 80% of data included in the present data set should be related to Objects included in AXDS-AXCS3.

18.57 AXDS-AXCS6

A group of at least 30 representative AXCS related data (see pertinent database structure and Use Case). The included data should be the most various as possible.

18.58 AXDS-AXCS7

A group of at least 30 representative Users related data (see pertinent database structure and Use Case). The included data should be the most various as possible, containing a 30% of null, inconsistent, duplicated and data-type critical data (i.e. for example 0 for numeric data type to be tested in mathematic division operand).

18.59 AXDS-AXCS8

A group of at least 30 representative Users related data (see pertinent database structure and Use Case). The included data should be the most various as possible.

18.60 AXDS-AXCS9

A group of at least 30 passwords random generated. Password length should be the most various as possible and containing a 30% of null and data-type critical data (i.e. for example 0 for numeric data type to be tested in mathematic division operand).

18.61 AXDS-OIDGen1

A set of 20 ObjectID's and associated information.

18.62 AXDS-Supervisor1

A group of 20 user identifier (AXUID).

18.63 AXDS-Supervisor2

A group of 20 tool identifiers (AXTID).

18.64 AXDS-CertVer1 (Full data set

A group of at least 20 sets that include: user identifier (AXUID), Tool Identifier (AXTID), Tool fingerprint, Device fingerprint, Tool operation history and Tool operation history fingerprint.

18.65 AXDS-CertVer2 (Subset 1)

A group of at least 20 sets that include: user identifier (AXUID) and Device fingerprint.

18.66 AXDS-CertVer3 (Subset 2)

A group of at least 20 sets that include: user identifier (AXUID), Tool Identifier (AXTID), Tool fingerprint and Device fingerprint.

18.67 AXDS-ACCREP1

A group of at least 100 action-logs chosen with the following criteria: (i) they must be related to at least 10 different users; (ii) they must be related to at least 10 different AXMEDIS objects. The list of objects and user they refer to, must be known.

18.68 AXDS-DRMSupport1

A group of at least 25 sets with information related with personal data from the license issuer and license petitioner, right(s) to be granted, resource over which the rights are granted, conditions of use.

- UID (issuer)
- UID (grantee)
- right(s)
    - AXOID
    - *validityInterval
    - *countLimit
    - *validityRegion
    - *feeInformation
    - *adaptationRules

* Optional fields

18.69 AXDS-DRMSupport2

A group of at least 25 sets with licencenseID (existing in license DB), the UID (user identifier) of the authorisation petitioner, right to be authorised and resource identifier over which the action has to be taken.

- UID (petitioner)
- right
- AXOID
- *LicenseID

* Optional fields

18.70 AXDS-DRMSupport3

A group of at least 10 sets with two or more licencenseID (existing in license DB for creating a new object), personal data from the license issuer and license petitioner, right(s) to be granted, resource over which the rights are granted, conditions of use.

- LicenseID's (two or more)
- UID (issuer)
- UID (grantee)
- right(s)
    - AXOID
    - *validityInterval
    - *countLimit
    - *validityRegion
    - *feeInformation
    - *adaptationRules

* Optional fields

18.71 AXDS-DRMSupport4

A group of at least 15 licenses (stored by the user or in de central license DB) that are capable to be migrated from one device to another.

18.72 AXDS-DRMSupport5

A group of at least 15 sets of AXMEDIS object, Symmetric key and Cryptographic information.

18.73 AXDS-DRMSupport6

A group of at least 15 sets of AXMEDIS object, Symmetric key, Cryptographic information and license(s).

18.74 AXDS-DRMSupport7

- LicenseID's (two or more)
- UID (issuer)
- UID (grantee)
- right(s)
  - AXOID
  - *validityInterval
  - *countLimit
  - *validityRegion
  - *feeInformation
  - *adaptationRules

* Optional fields

18.75 AXDS-DRMSupport8

A group of at least 10 sets with two or more licencenseID (existing in license DB for creating a new object), personal data from the license issuer and license petitioner, right(s) to be granted, resource over which the rights are granted, conditions of use. Business rule includes time or data consumption.

- LicenseID's (two or more)
- UID (issuer)
- UID (grantee)
- right(s)
  - AXOID
  - *validityInterval
  - *countLimit
  - *validityRegion
  - *feeInformation
  - *adaptationRules

* Optional fields

18.76 AXDS-PMS1

An AXMEDIS object, Protection information (including keys).

18.77 AXDS-PMS2

Protection information related to an AXMEDIS Object.

18.77.1 AXDS-RET1

A group of at least 20 different licenses of every REL supported. These licenses should include different type of content and rights, valid and not valid: no limitations to that.

18.78 AXDS-ENCDEC1

A set of 15 symmetric and asymmetric key (key pair) and its associated cryptographic information.

18.79 AXDS-PIMulti

A set of at least 5 AXMEDIS Objects containing audio and video, at different sampling rates and frame sizes

18.80 AXDS-PIVid

A set of at least 5 AXMEDIS Objects containing video content with at least two different frame sizes (a high quality like TV size and a lower quality)

18.81 AXDS-PIAu

A set of at least 5 AXMEDIS Objects containing audio content with at least two different sampling rates (a high quality like CD and a lower quality).

18.82 AXDS-MCProject

The Media Club project definition data – typically running on a remote Xaura DB installation.

18.83 AXDS-MCProducer

The Media Club data definition of back-office users and permissions – typically running on a remote Xaura DB installation.

18.84 AXDS-MCObject

Set of AXMEDIS Objects, tailored to be managed by the Media Club, some of which are enctrypted, that can be updated, downloaded or streamed on the client side.

18.85 AXDS-MCShop

The Media Club data definition for the shop.

18.86 AXDS-MCPayMethod

The Media Club data specifying all available payment methods.

18.87 AXDS-MCTransaction

The Media Club transaction data.

18.88 AXDS-MCTestUser

1. A test login to be used in each test case.
2. test User e-mail
3. List of Authorizations associated with the test login.
4. Sample of user profile to be submitted.
5. wallet
6. gift certificate

18.89 AXDS-Video

A selection of content (more than 10) that represents the typical video content distributed within AXMEDIS with related metadata, coming from different CMSs. Some Items will include a preview.

18.90 AXDS-PCDist1

An AXMEDIS Protected Object.

18.91 AXDS-Kiosk1

- AXMEDIS object stored in the local AXDBM
  - Sample 1 = Image + Text
  - Sample 2 = Image + Audio

  - Sample 3 = Video + Audio
  - Sample 4 = Text + Image + Audio
  - Sample 5 = Audio + Text
  - Sample 6 = Animation + Audio + Text

18.92 AXDS-Kiosk2

- AXMEDIS object exposed on the AXEPTool
  - Sample 7 = Image + Text
  - Sample 8 = Image + Text
  - Sample 9 = Image + Audio
  - Sample 10 = Image + Audio
  - Sample 11 = Text + Image + Audio
  - Sample 12 = Text + Image + Audio
  - Sample 13 = Video + Audio
  - Sample 14 = Video + Audio + Text

18.93 AXDS-Kiosk3

- Top-ten AXMEDIS objects stored locally
  - any of the previously mentioned objects or: Sample 1- 14

18.94 AXDS-Kiosk4

- The AXMEDIS object stored in the local fruition device
  - any of the previously mentioned objects or: Sample 1- 14

18.95 AXDS-Kiosk5

AXMEDIS Certifier & Supervisor user management data

18.96 AXDS-Kiosk6

Licenses of the AXMEDIS governed object

18.97 AXDS-Kiosk7

Composition & formatting rules

18.98 AXDS-Kiosk8

DRM rules

18.99 AXDS-Kiosk9

Selected Content List for Kiosk

18.100 AXDS-Kiosk10

The kiosk applications

18.101 AXDS-Kiosk11

The kiosk procedures

18.102 AXDS-Kiosk12

The kiosk catalogue

18.103 AXDS-Kiosk13

The kiosk local user management data

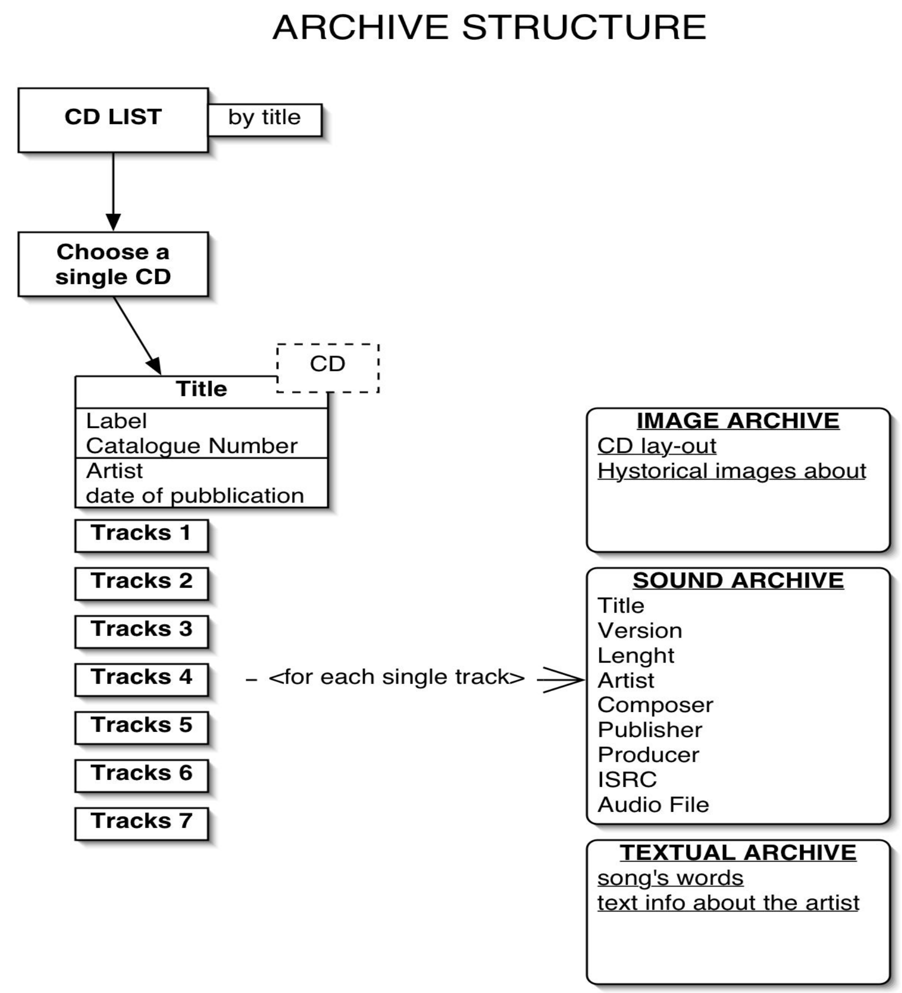
19 AXMEDIS Content used by partners

19.1 Study of current standards on content description

Historical archive: Cataloguing standard: EAD. Type of files: TIFFs, JPGs
Library
Archive of photographs: Cataloguing standard: scheda F ICCD customized. Type of files: TIFFs, JPGs
Sound archive: Cataloguing standards: SBN (The Italian national service for libraries). Type of files: Wave, MP3s
Musical instrument museum: Cataloguing standards still in evaluation. Should be defined within summer 2005. Cataloguing standard: None. Structure not defined yet, will be ready within 2005.
Metadata: ICCU (i.e. national) Mag schema 1.5 for books (2.0 upcoming with metadata for audio)
Art: Cataloguing standard as defined by IEEE-LOM/IMS: EAD. Type of files embedded into the LO: TIFFs, JPGs, MP3, WAW, QickTime, AVI, MPEG, TXT, HTML, Flash and VRML.
Medical: Cataloguing standard as defined by IEEE-LOM/IMS: EAD. Type of files embedded into the LO: TIFFs, JPGs, MP3, WAW, QuickTime, AVI, MPEG, TXT, HTML, Flash, QuickTimeVR.
Other: Cataloguing standard as defined by IEEE-LOM/IMS: EAD. Type of files embedded into the LO: TIFFs, JPGs, MP3, WAW, QuickTime, AVI, MPEG, TXT, HTML, Flash and VRML.
Sound archive: Cataloguing standards: ISRC- Type of files: Aiffs, MP3s, MP4, AAC

Images Archive: Type of files: JPG, TIFF
Text Archive: Type of files: txt, pdf

The structure of data are described in the diagram below:



19.2 Content description from ANSC

Contents from ANSC consist in different types of archives; they are mainly related to the ANSC history and activity and managed by means of a XML CMS (DAMS Digital Archives Memory Storage. Engine: Extraway).
Historical archive: contains documents from 1650 to present day. It is partially digitized (35.000 images ca). Digitization to be completed depending from funds available. Cataloguing standard: EAD
Library: Some music manuscript and early editions are going to be digitized within summer 2005. Cataloguing standards: SBN (The Italian national service for libraries)
Archive of photographs: Contains photographs (ca. 13000) from the end of XIX century to present day related mainly to ANSC concerts but also various artists, singers, actors, musicians, and so on. Cataloguing standard: Scheda F ICCD customized and transformed in XML.
Sound archive: 3 archives: Sound archive of concerts, Archive of oral traditional music, Archive of recordings. Cataloguing: Proprietary for oral traditional music, For recordings and concerts: National standard (Windj>Unimarc). Records of our concerts are not available for
Musical instrument museum: Documentation about ANSC collection of 297 musical instrument: Photographs, technical drawings, Cataloguing standard: None. Structure not defined yet, will be ready within 2005.
All cataloguing standards are (and will) be merged into our CMS.

19.3 Content description from AFI

Contents from AFI consist mainly in audio files that will be stored in CMS (php+mysql / GUI flash+html) in the filesystem. The CMS will be the same for many different production partners (AFI DB for all associates). Every partner will be identified by user/pass.

Archive: There will be 1 archive containing all Cds produced by AFI associated.
Inside the CD data you will find the subdata about all the recordings (audio tracks) included in the CD with their own related data (title, version (original, remix ec), length, artists, authors, composers, publisher, producer, ISRC code, catalogue number, label).
We will develop a part of the archive also for:

1- graphics part for a product (CD Lay-out) (IMAGE ARCHIVE)
2- Song's words (TEXTUAL ARCHIVE)
3- textual info about the artist. (TEXTUAL ARCHIVE)
4- Historical image material. (IMAGE ARCHIVE)

19.4 Content description from ILABS

Contents from ILABS consist in different types of learning objects and courses; they are mainly related to ILABS involvement in the educational and cultural environment. ILABS has developed an e-learning infrastructure covering the whole value chain from editing to delivery of educational content and offers to own customers also the possibility of "content development on demand". Form major chemical companies have been developed ad hoc training on clinical and medical subject too. So in the overall ILABS content covers the following kind of contents:
Art: fruition, understanding and restoration of art objects (to this category will belong most of the sample objects for the test case).
Medical: clinical training on breast cancer / first emergency intervention / privacy management in medical environment
Other: training courses for banks / financial institutions. Maintenance courses to be used on wearable computing.

19.5 Content Description from SEJER

Content produced by Nathan and Bordas are object called "pedagogical units", that can be compounded. The biggest pedagogical unit is in fact the whole schoolbook.

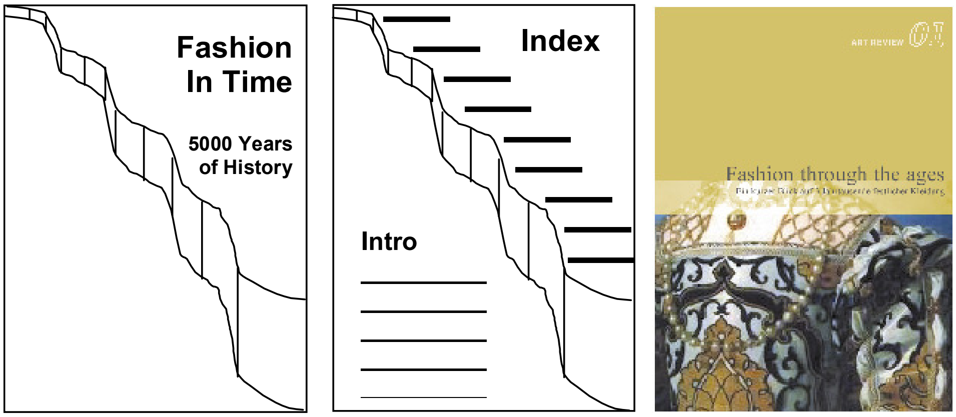
Pedagogical units are complex objects, mixing different kind of "atomic" formats: images, text etc. Besides, metadata are associated to each pedagogical unit.

The "atomic" content formats use to construct a pedagogical unit are those defined in 3.2 Content Description.

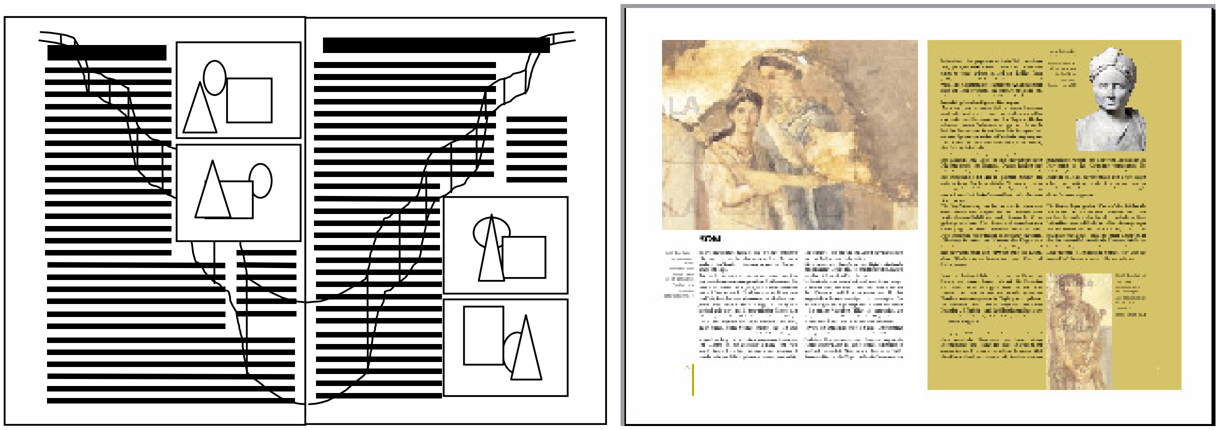
In addition, RDF and raw XML are used to describe metadata (depending on the capacities of the viewer that will be used to display the resource).

Appendix A. Samples of Content in editorial products

In this section are reported samples of traditional print oriented content starting from the description of the idea (objective of the content) up to samples of images form the finalized mock-up used for the final decision from the editorial board:

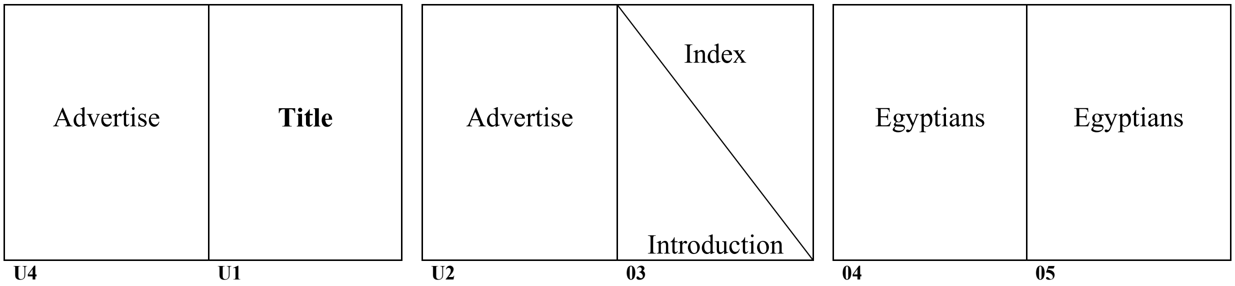


Sample of a cover / index page (sketch from the author) / Actual cover (final mock-up)

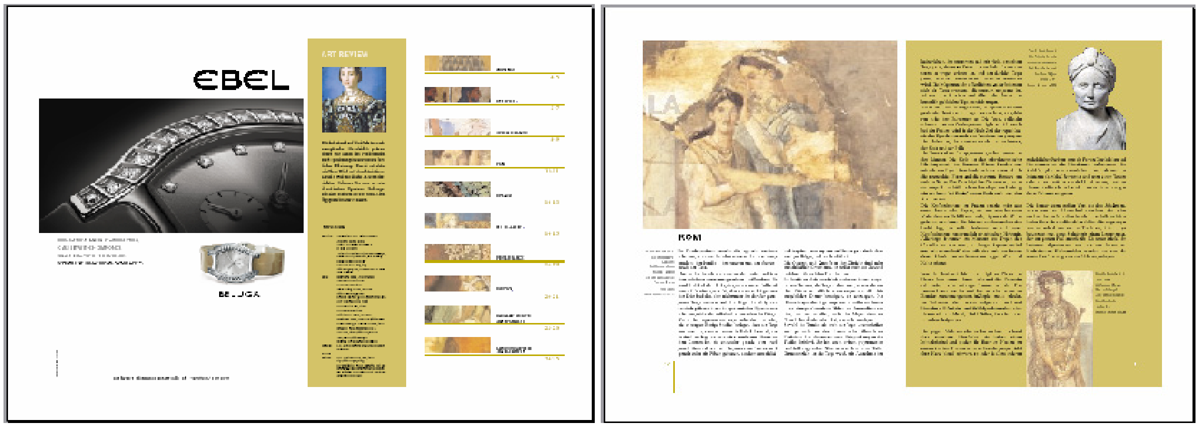


Sample of internal pages (sketch from the author) / an actual couple of pages (final mock-up)

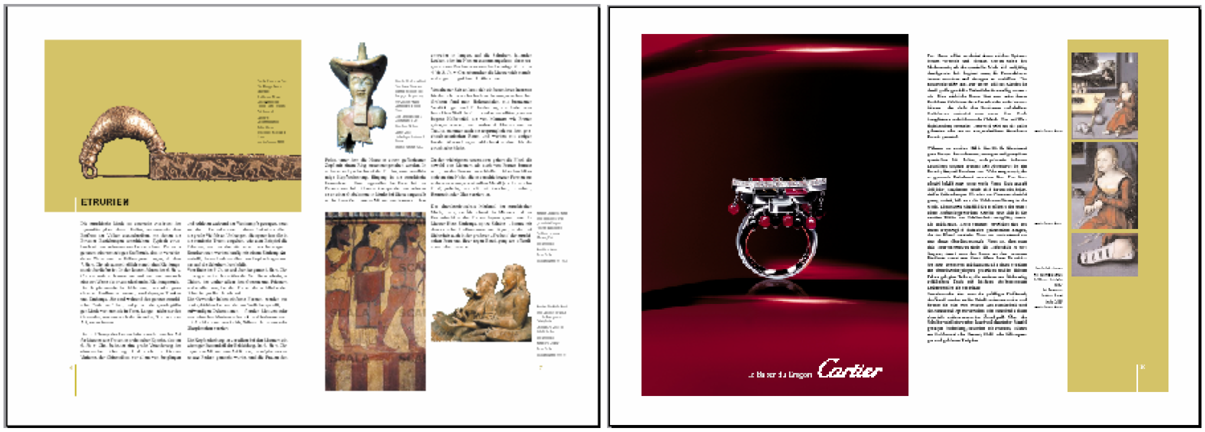
In the following pages are reported the layout schema used to produce a small 20 content page magazine including advertisement based on the author sketch just presented. Actual content is selected and placed into the pages where during the process are defined place holders for images, captions, drill-down text…



Greeks	Greeks	Etruscans	Etruscans	Romans	Romans
06	**07**	**08**	**09**	**10**	**11**
Byzantine	Byzantine	Middle Ages	Middle Ages	Renaissance	Renaissance
12	**13**	**14**	**15**	**16**	**17**
Baroque	Baroque	History of shoes	Curiosities	Imprint Credits	Advertise
18	**19**	**20**	**21**	**22**	**U3**

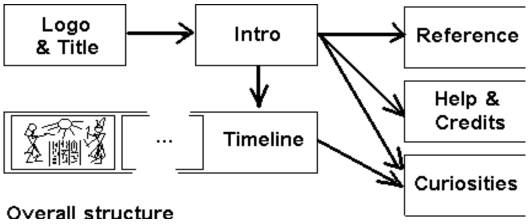


Sample of opposite pages layout for a magazine on the History of Fashion

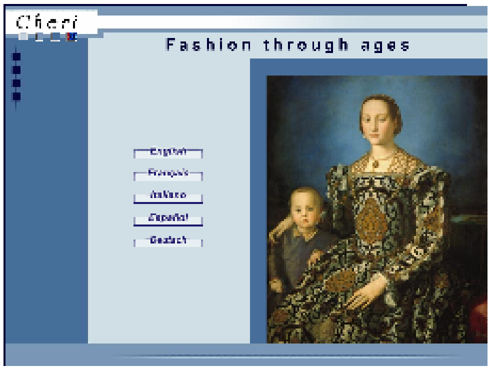


Sample of opposite pages layout for a magazine on the History of Fashion

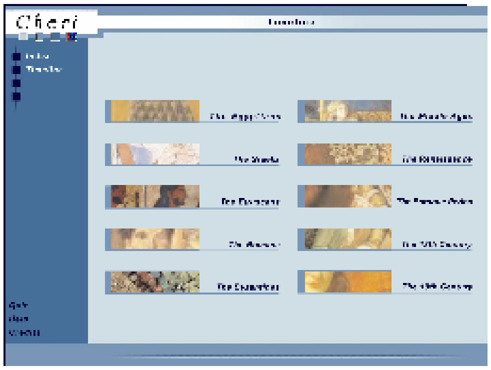
From a Book or a magazine is possible to extract a multimedia presentation. The process will exploit part of the work carried out in the design process of the other product. In the following picture is reported the navigation structure derived from the previously mentioned paper-based layout.
In the case of a multimedia presentation the staring point is to define the navigation process and the sections that will be available as independent ones. The typical structure foresee an introductory part where Logo of the producer and Title of the product appear and may / may not be accompanied by a music background or some audio (speech…). Once ended the product presentation starts the proper introduction where are accessible the various part of the presentation itself (as reported in the picture describing the process applied for the previous product when turning it into a CD). The introduction usually ends with a main menu page / section from where the user can access each section composing the product.



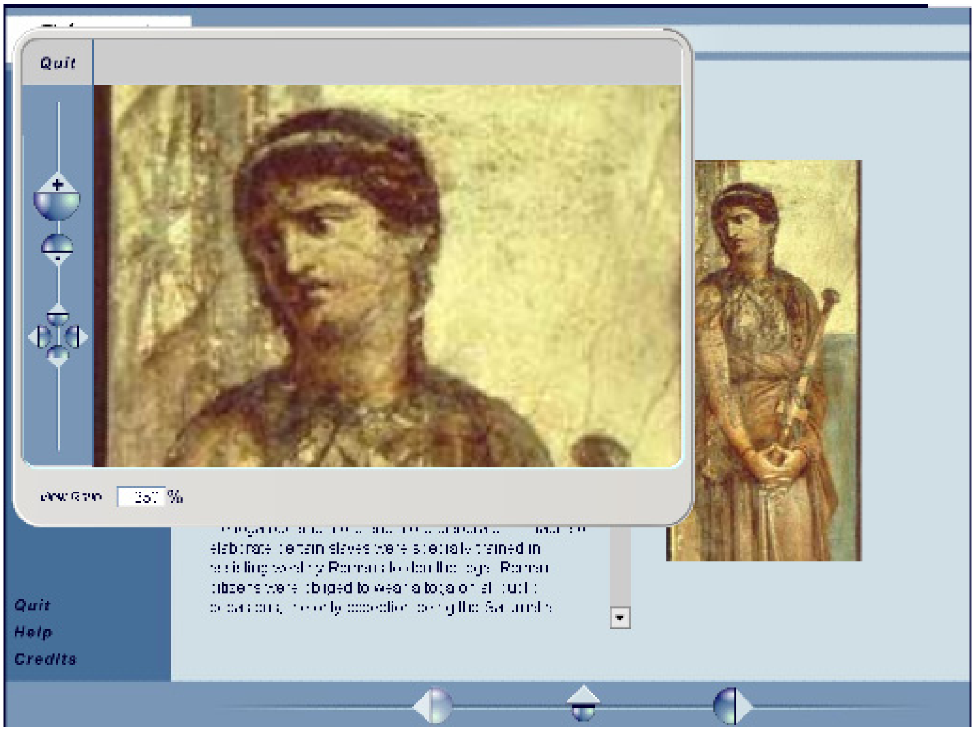
Navigation Structure derived for the CD



Language selection page



Navigation menu (timeline)



[illegible]
elaborate; certain slaves were specially trained in
assisting wealthy Romans to don the toga. Roman
citizens were obliged to wear a toga on all public
occasions, the only exception being the Saturnalia

Sample of a content page (Flash) with open pop-up window for browsing image detail