Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE2.2.1.2
Test Cases and Content Description, First update of DE2.2.1

Version: 2.3
Date: 03/06/2006
Responsible: FUPF (silvia.llorente@upf.edu, victor.torres@upf.edu)
(revised and accepted by Coordinator)

Project Number: IST-2-511299 Project Title: AXMEDIS Deliverable Type: Public Visible to User Groups: Yes Visible to Affiliated: Yes Visible to Public: Yes
Deliverable Number: DE2.2.1.2 Contractual Date of Delivery: Actual Date of Delivery: 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

AXMEDIS Copyright Notice

The following terms (including future possible amendments) set out the rights and obligations licensee will be requested to accept on entering into possession of any official AXMEDIS document either by downloading it from the web site or by any other means.

Any relevant AXMEDIS document includes this license. PLEASE READ THE FOLLOWING TERMS CAREFULLY AS THEY HAVE TO BE ACCEPTED PRIOR TO READING/USE OF THE DOCUMENT.

1. **DEFINITIONS**
   - i. "**Acceptance Date**" is the date on which these terms and conditions for entering into possession of the document have been accepted.
   - ii. "**Copyright**" stands for any content, document or portion of it that is covered by the copyright disclaimer in a Document.
   - iii. "**Licensor**" is AXMEDIS Consortium as a de-facto consortium of the EC project and any of its derivations in terms of companies and/or associations, see www.AXMEDIS.org
   - iv. "**Document**" means the information contained in any electronic file, which has been published by the Licensor's as AXMEDIS official document and listed in the web site mentioned above or available by any other means.
   - v. "**Works**" means any works created by the licensee, which reproduce a Document or any of its part.
2. **LICENCE**
   1. The Licensor grants a non-exclusive royalty free licence to reproduce and use the Documents subject to present terms and conditions (the **Licence**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.
   2. In consideration of the Licensor granting the Licence, licensee agrees to adhere to the following terms and conditions.
3. **TERM AND TERMINATION**
   1. Granted Licence shall commence on Acceptance Date.
   2. Granted Licence will terminate automatically if licensee fails to comply with any of the terms and conditions of this Licence.
   3. Termination of this Licence does not affect either party's accrued rights and obligations as at the date of termination.
   4. Upon termination of this Licence for whatever reason, licensee shall cease to make any use of the accessed Copyright.
   5. All provisions of this Licence, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this Licence and shall continue in full force and effect.
   6. Notwithstanding License termination, confidentiality clauses related to any content, document or part of it as stated in the document itself will remain in force for a period of 5 years after license issue date or the period stated in the document whichever is the longer.
4. **USE**
   1. Licensee shall not breach or denigrate the integrity of the Copyright Notice and in particular shall not:
      - i. remove this Copyright Notice on a Document or any of its reproduction in any form in which those may be achieved;
      - ii. change or remove the title of a Document;
      - iii. use all or any part of a Document as part of a specification or standard not emanating from the Licensor without the prior written consent of the Licensor; or
      - iv. do or permit others to do any act or omission in relation to a Document which is contrary to the rights and obligations as stated in the present license and agreed with the Licensor
5. **COPYRIGHT NOTICES**
   1. All Works shall bear a clear notice asserting the Licensor's Copyright. The notice shall use the wording employed by the Licensor in its own copyright notice unless the Licensor otherwise instructs licensees.
6. **WARRANTY**
   1. The Licensor warrants the licensee that the present licence is issued on the basis of full Copyright ownership or re-licensing agreements granting the Licensor full licensing and enforcement power.
   2. For the avoidance of doubt the licensee should be aware that although the Copyright in the documents is given under warranty this warranty does not extend to the content of any document

which may contain references or specifications or technologies that are covered by patents (also of third parties) or that refer to other standards. AXMEDIS is not responsible and does not guarantee that the information contained in the document is fully proprietary of AXMEDIS consortium and/or partners.

3. Licensee hereby undertakes to the Licensor that he will, without prejudice to any other right of action which the Licensor may have, at all times keep the Licensor fully and effectively indemnified against all and any liability (which liability shall include, without limitation, all losses, costs, claims, expenses, demands, actions, damages, legal and other professional fees and expenses on a full indemnity basis) which the Licensor may suffer or incur as a result of, or by reason of, any breach or non-fulfilment of any of his obligations in respect of this Licence.

7. **INFRINGEMENT**

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

8. **GOVERNING LAW AND JURISDICTION**

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

Please note that:

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

Table of Contents

1 Executive Summary and Report Scope

Market and end-users are pressing content industry to reduce prices. This is presently the only solution to setup viable and sustainable business activities with e-content. Production costs have to be drastically reduced while maintaining product quality. Content providers, aggregators and distributors need innovative instruments to increase efficiency. A solution is automating, accelerating and restructuring the production process to make it faster and cheaper. The goals will be reached by: (i) accelerating and reducing costs for content production with artificial intelligence algorithms for content composition, formatting and workflow, (ii) reducing distribution and aggregation costs, increasing accessibility, with a P2P platform at B2B level integrating content management systems and workflows, (iii) providing algorithms and tools for innovative and flexible Digital Rights Management, exploiting MPEG-21 and overcoming its limits, supporting several business and transactions models.

AXMEDIS consortium (producers, aggregators, distributors and researcher) will create the AXMEDIS framework with innovative methods and tools to speed up and 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 some 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 a revised version of the early use cases it is related to all the deliverables of WP2 which is devoted to the continuous collection and analysis of user requirements. This activity is performed by setting up a user group of experts and by considering the content production models, educational paradigms, entertainment models, distribution paradigms and protection innovative aspects of the project. The work includes the adoption of interviews and the identification of use cases, description of the test cases, (while the corresponding collection of reference content for stressing key problems and for the eventual verification and validation of corresponding solutions is performed in WP8), collection of current practices (best practices) in using media technologies and solutions (processes, tools, methodologies, equipment, etc), identification of distribution processes and models.

Main deliverables are:

- DE2.3.1.2 – User Group Maintenance (M13) -- responsible UNIVLEEDS -- report on the activity related in the management and improvement of the user group, enlargement of it, analysis of the coverage of the UG with respect to the project topics, etc., activity to be performed in the next months (evolution of DE2.3.1);
- DE2.1.1.2.1 -- User Requirements, first update (M16) – responsible DSI -- this deliverable contains the revised and updated version of the user requirement produced in DE2.1.1; This deliverable has been planned since the Annex I and it has been split in the following two deliverables that initially have been planned to be under the same number. They were too large document to be considered single documents in this phase;
- DE2.1.1.2.2 – Use Cases and Scenarios, first update (M17) – responsible DSI -- this deliverable contains the revised and updated version of the Use cases and scenarios produced in DE2.1.1;
- DE2.2.1.2 – Test cases and content description, first update (M20) – responsible FUPF -- this deliverable contains the revised and updated version of the test cases for research functionalities and AXMEDIS tool

validation, starting from the DE2.2.1; In this case, the description of test cases will be more precise since the first results coming from the WP8 will be available and thus effective links from what can be done for testing and which content has to be used will be possible.

- DE2.3.1.3 – User Group Maintenance, first update (M20) – responsible UNIVLEEDS -- report on the activity related in the management and improvement of the user group, its enlargement, analysis of the coverage of the UG with respect to the project topics, etc., activity to be performed in the next months (evolution of DE2.3.1.2);

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

WP2.3 -- Set up and management of a AXMEDIS User Group -- responsibility UNIVLEEDS -- the established user group of experts will be enlarged and kept informed. The members will receive updated news about project evolution and will constitute a source for requirements and use cases collection and validation; moreover they will contribute to testing and validating produced results. The user group presently presents several experts representing the different users of AXMEDIS tools both at business and consumer levels. They include content producers, content integrators, content designers, usability experts, content distributors, content aggregators, publishers, etc.

WP2.4 -- Updating requirements analysis after first period -- responsibility DSI – (M16-M17) In this WP, the updating of the requirements and of use cases and test cases in the periods after the first is performed. Updating means to revise requirements, use cases and test cases, in order to see if they need to be revised and/or improved and/or deleted and/or added according to changes in the state of the art, needs of the context and users, etc. This process of updating is continuously performed and after each intermediate validation such as that of M14 and during the final validation. Requirements, use cases and test cases will be updated by considering the points of view of content designers, multimedia producers, integrators, final users, taking into account project partners, their experts, and experts of the user group by using specific interview based on guidelines produced by the consortium. Other sources of information will be the monitoring and participation to MPEG-21, DMP, and other groups. The use cases and test cases will be structured according to UML methodology creating and updated also scenarios as performed in the first period. As a result a new version of related deliverables will be produced updating and expanding those collected and reported in the deliverables of the first period. 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. In this case, the test cases will be more precise since the first results coming from the WP8 will be available and thus effective links from what can be done for testing and which content has to be used will be possible. The Content for the test cases will be collected and/or produced in WP8. The test cases will be structured according to structure of the AXMEDIS framework and tools as defined and developed in the first 12 months of the project.

Use cases and user requirements have been already defined in a different deliverable.

2 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.

2.1 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 AXMEDIS Object editing (DSI, EXITECH, EPFL)

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

3.1.1 Creation of a new AXMEDIS object

TCId	TC3.1.1
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 mandatory metadata
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

3.1.2 Load and save AXMEDIS objects

TCId	TC3.1.2
Test case	Load and save AXMEDIS objects
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

3.1.3 Navigating through AXMEDIS objects (DSI, Bellini)

TCId	TC3.1.3
Test case	Navigating through AXMEDIS objects
Initial conditions	An AXMEDIS object has been loaded form file or from Database
Configuration description	None
Description of functionality to be tested	AXMEDIS Editor allows to browse the structure of the AXMEDIS object
Partners, people involved	Content provider, Content Integrator,
Validator skill	None
Data set used	AXDS-Editor1
Steps	1 The user can see the structure of the AXMEDIS object using the Hierarchy Editor/Viewer
Expected results	The structure is correctly represented
Variations	None
Issues	None
Test case Scope/Type	GUI, BlackBox

3.1.4 Adding AXMEDIS elements to an existing AXMEDIS object

TCId	TC3.1.4
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…" or Insert element and then chooses the type of element he/she wants to add 4 If the user has the needed rights: 4.1 The element is added and the hierarchal view shows to the user a new node representing the element in the chosen position 4.2 To verify the modification have been really made the user has to execute Test Case "Save an AXMEDIS object" 5 Else: 5.1 The system shows a dialog box to inform the user about why he/she cannot add the element
Expected results	The element is correctly added or inserted
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

3.1.5 Extracting AXMEDIS elements

TCId	TC3.1.5
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

3.1.6 Removing an element from an AXMEDIS Object

TCId	TC3.1.6
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

3.1.7 Moving an element within the AXMEDIS Object

TCId	TC3.1.7
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	A "Move up" and "Move down" option in the menu can be used
Issues	None
Test case Scope/Type	GUI, BlackBox

3.1.8 Adding a resource

TCId	TC3.1.8
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

3.1.9 Managing/Modifying resources

TCId	TC3.1.9
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

3.1.10 Navigating and understanding DRM Rules and PAR (FUPF)

TCId	TC3.1.10
Test case	Navigating and understanding DRM Rules and PAR
Initial conditions	None
Configuration description	None
Description of functionality to be tested	Navigating and understanding DRM Rules and PAR
Partners, people involved	Contet provider, content integrator.
Validator skill	None special.
Data set used	AXDS-DRMSupport1
Steps	1. Open a license file 2. View its fields and freely navigate through the information.
Expected results	The user can easily browse through the DRM Information.
Variations	1. Using DRM Editor and Viewer (standalone application) 2. Using axeditor. A more detailed could be done by comparing the data offered in the application with the data observable by inspection of the XML license.
Issues	None
Test case Scope/Type	GUI

3.2 AXMEDIS Internal Viewers (DSI)

3.2.1 Invoking an internal viewer/editor

TCId	TC3.2.1
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

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

TCId	TC3.2.2
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

3.2.3 Closing an Internal viewer/editor

TCId	TC3.2.3
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	Closing an Internal viewer/editor and updating of the resource

tested	
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

3.2.4 Displaying HTML pages with internal resources (SEJER)

TCId	TC3.2.4
Test case	Displaying HTML pages with internal resources
Initial conditions	An Internal viewer/editor is running and displaying an HTML resource belonging to an AXMEDIS object.
Configuration description	An internal viewer has been invoked by the system
Description of functionality to be tested	Displaying HTML pages with internal resources, such as images and stylesheets.
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	AXDS-DB3
Steps	1 The actor chooses to display an HTML resource linked with internal resources as images and CSS.
Expected results	The HTML page is displayed correctly
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

3.3 AXMEDIS visual and behavioural viewer (EPFL, DSI)

3.3.1 Editing the visual scene for SMIL resource

3.3.1.1 Creating and deleting the visual scene for SMIL resource (EPFL)

TCId	TC3.3.1.1
Test case	Creating and deleting the visual scene for SMIL resource
Initial conditions	An internal visual editor and viewer has been invoked by the system
Configuration description	
Description of functionality to be tested	Creating and deleting the visual scene for SMIL resource
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to create/delete an Element in a SMIL scene component 2 The user clicks with left mouse button on the palette on the left of the Visual Editor. 3 When the Rectangular button is toggled, the user can create an Element (e.g., rectangular shape) on the canvas of the visual editor. 4 The user left clicks in the canvas to create an element (e.g., rectangular shape) to represent the element area in the SMIL scene. 5 The user clicks “remove” to delete an Element from canvas.
Expected results	The visual scene is correctly created and deleted
Variations	None
Issues	None
Test case Scope/Type	GUI

3.3.1.2 Resizing and moving the visual scene for SMIL resource (EPFL)

TCId	TC3.3.1.2
Test case	Resizing and moving the visual scene for SMIL resource
Initial conditions	An internal visual editor and viewer has been invoked by the system
Configuration description	
Description of functionality to be tested	resizing and moving the visual scene for SMIL resource
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 the user wants to resize the Elements for the media resources. 2 The user left clicks on the visual element (e.g., rectangular shape) on the canvas of the Visual Editor. 3 When the Rectangular is selected a) the user can resize the shape of the visual Element by dragging its edges in different directions to reach the desirable size. b) the user can move the visual Element to any position on the canvas.
Expected results	The visual scene is correctly resized and moved

Variations	None
Issues	None
Test case Scope/Type	GUI

3.3.1.3 Changing the background colour of the visual scene (EPFL)

TCId	TC3.3.1.2
Test case	Changing the background colour of the visual scene
Initial conditions	An internal visual editor and viewer has been invoked by the system
Configuration description	
Description of functionality to be tested	Changing the background colour of the visual scene
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to change the background colour of visual elements. 2 The user left clicks on the rectangular shape on the canvas of the Visual Editor. 3 When the rectangular is selected, the user can choose the button or right click to select the option of changing the background colour of the visual element.
Expected results	The background color of the visual scene is correctly changed
Variations	None
Issues	None
Test case Scope/Type	GUI

3.3.1.4 Inclusion of media resources into the visual scene for SMIL resource (EPFL)

TCId	TC3.3.1.4
Test case	Inclusion of media resources into the visual scene for SMIL resource
Initial conditions	An internal visual editor and viewer has been invoked by the system
Configuration description	
Description of functionality to be tested	Inclusion of media resources into the visual scene for SMIL resource
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to associate media resources with the Elements of SMIL scene 2 The user left clicks on the rectangular to choose an Element for holding the media resources. 3 The user can click on the right button of the mouse (or choose the menu on the frame, in both cases) with the options of “adding text”, “adding audio”, “adding video”, “adding image” to include different types of media resources. 4 In alternative the user may browse the AXMEDIS hierarchy for selecting a resource and drag it to an Element and drop it for associating the resource to the Element in this manner. 5 In alternative, the user can browse the media resources in the AXMEDIS object hierarchy and select one of them to associate it to the region.
Expected results	Media resources is correctly included into the visual scene
Variations	Right click on the visual scene of SMIL resource; system will prompt a menu with

	the same options on the frame
Issues	None
Test case Scope/Type	GUI

3.3.2 Editing the temporal information of media resources (EPFL)

3.3.2.1 Editing the unit and length of timeline (EPFL)

TCId	TC3.3.2.1
Test case	Editing the unit and length of timeline
Initial conditions	• The SMIL scene is already created and the AXMEDIS resources have already been associated with each Element using visual editor • An internal behaviour editor and viewer has been invoked by the system
Configuration description	
Description of functionality to be tested	Editing the unit and length of timeline
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to edit the time ruler by clicking the behaviour editor and viewer button. 2 The user will have a contextual menu with a list of the properties of the timeline and the user can edit the scale of time by inputting the time unit (in second, 10seconds, etc). 3 The user can then edit the entire displaying time length of the timeline by changing the length of the ruler. 4 After clicking “OK” of the contextual menu, the timeline will change according to the new information.
Expected results	The unit and length of timeline is correctly edited
Variations	None
Issues	None
Test case Scope/Type	GUI

3.3.2.2 Editing the displaying time boundary of each media resource (EPFL)

TCId	TC3.3.2.2
Test case	Editing the displaying time boundary of each media resource
Initial conditions	• The SMIL scene is already created and the AXMEDIS resources have already been associated with each Element using visual editor • An internal behaviour editor and viewer has been invoked by the system
Configuration description	
Description of functionality to be tested	Editing the displaying time boundary of each media resource
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	

Steps	1 The user wants to edit the displaying time for the AXMEDIS resources associated to Elements in the SMIL scene by clicking the behaviour editor button. 2 The behaviour editor and viewer will show the time line of the AXMEDIS resources and a temporal window of activation (a rectangular shape) along the time axis 3 The user can modify the length and position of the rectangle on the time line to indicate the different starting time and display duration.
Expected results	displaying time boundary of each media resource is correctly edited
Variations	None
Issues	None
Test case Scope/Type	GUI

3.3.3 Previewing the SMIL resources after editing (EPFL)

TCId	TC3.3.3
Test case	Previewing the SMIL resources after editing
Initial conditions	• The SMIL scene is already created and the AXMEDIS resources have already been associated with each Element of the SMIL Scene or sequence of scenes by using visual editor. • The temporal information of AXMEDIS resources for the elements of the SMIL Scene has been already edited using behaviour editor.
Configuration description	
Description of functionality to be tested	Previewing the SMIL resources after editing
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to preview SMIL Scene or a sequence of them related to subsequence links via buttons of for direct connection after it is being created, edited or modified. 2 The user could left click on the button “Preview” on the menu of the frame to preview the SMIL Scene by the internal SMIL player, starting from the current SMIL Scene under editing. And there would be a new frame popping up to show. 3 The user could left click on the button “Stop” on the menu of the frame to stop previewing SMIL Scene. 4 The user could close the frame to stop previewing SMIL Scene
Expected results	The SMIL resources after editing can be previewed
Variations	None
Issues	None
Test case Scope/Type	GUI

3.3.4 Loading and saving the SMIL component into AXMEDIS object (EPFL)

TCId	TC3.3.4
Test case	Loading and saving the SMIL component into AXMEDIS object
Initial conditions	The visual editor is already activated and new SMIL component has been created/edited.
Configuration description	
Description of	Loading and saving the SMIL component into AXMEDIS object

functionality to be tested	
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1. The user wants to load the existent SMIL scene to edit or save it after editing by clicking load/save on the menu of frame. 2. After saving, the SMIL component would enclose inside a <Component> tag pair in AXMEDIS object. 3. The save of new SMIL scene has to be performed automatically providing the right position into the AXMEDIS object. The first SMIL scene of an AXMEDIS object saved is called Start (it is the starting point). 4. The user can load a SMIL scene and display it by double clicking on it from the AXMEDIS hierarchy of the AXMEDIS Editor.
Expected results	The SMIL component can be correctly loaded and saved into AXMEDIS object
Variations	None
Issues	None
Test case Scope/Type	GUI

3.4 Navigation and hyperlinking with multiple SMIL Scenes (EPFL; DSI)

TCId	TC3.4
Test case	Navigation and hyperlinking with multiple SMIL Scenes
Initial conditions	The set of SMIL scenes are in the AXMEDIS object. An SMIL Scene named as Start is located into the root of the object. The SMIL Scenes are connected one to another, some buttons and/or menus have been created to establish relationships among the difference SMIL Scenes into the AXMEDIS object.
Configuration description	
Description of functionality to be tested	Navigation and hyperlinking with multiple SMIL Scenes
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1. When one SMIL scene is under execution and it includes some buttons (realised as Elements, to create from simple to more complex menus) at which another SMIL Scene is connected via a link and thus the user wants to pass to another SMIL Scene by clicking “link” on the Element which is at the bottom of the SMIL Scene. On the other hand, the user does not need to do anything if he does not want to link to other resources. -- the SMIL Scene under execution complete the time line and pass to a next SMIL Scene as defined in Reference Link stated during the SMIL Scene editing. 2. When “link” is activated (in any case), the next SMIL Scene with related Elements and resources is loaded to be played replacing the previous one with all its related Elements.
Expected results	Multiple SMIL Scenes can be correctly navigated and hyperlinked
Variations	None
Issues	None

Test case Scope/Type	GUI

3.5 AXMEDIS OBJECT EDITOR AND VIEWERS (EPFL)

3.5.1 Opening annotations and comments of the media object

TCId	TC3.5.1
Test case	opening annotations and comments of the media object
Initial conditions	The object editor and viewer is activated.
Configuration description	
Description of functionality to be tested	opening annotations and comments of the media object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user selects a media object and clicks the object editor and viewer. 2 A list of annotations and comments of this media object will be shown. Each of them will be related to the original resource with a specific link.
Expected results	Annotations and comments of the media object can be opened
Variations	None
Issues	None
Test case Scope/Type	GUI

3.5.2 Adding annotations and comments of the media object

3.5.2.1 Adding audio annotations and comments of the media object

TCId	TC3.5.2.1
Test case	Adding audio annotations and comments of the media object
Initial conditions	The object editor and viewer is activated.
Configuration description	
Description of functionality to be tested	Adding audio annotations and comments of the media object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to add some comments or annotations for the media object. The user left clicks to select a media object in the AXMEDIS object resources 2 The user clicks “adding audio comments” 3 The user could record his/her own comments by using some available audio recorder/player 4 The user clicks “OK” to save this operation
Expected results	Audio annotations and comments of the media object can be added
Variations	None
Issues	None
Test case Scope/Type	GUI

3.5.2.2 Adding text annotations and comments of the media object

TCId	TC3.5.2.2
Test case	Adding text annotations and comments of the media object
Initial conditions	The object editor and viewer is activated.
Configuration description	
Description of functionality to be tested	Adding text annotations and comments of the media object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to add some comments or annotations for the media object. The user left clicks to select a media object in the AXMEDIS object resources 2 The user clicks “adding text comments” 3 The user could have a notebook to write his/her own comments. 4 The user clicks “OK” to save this operation
Expected results	text annotations and comments of the media object can be added
Variations	None
Issues	None
Test case Scope/Type	GUI

3.5.2.3 Adding graphical annotations and comments of the media object

TCId	TC3.5.2.3
Test case	Adding graphical annotations and comments of the media object
Initial conditions	The object editor and viewer is activated.
Configuration description	
Description of functionality to be tested	Adding graphical annotations and comments of the media object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to add some comments or annotations for the media object. The user left clicks to select a media object in the AXMEDIS object resources 2 The user clicks “adding graphical comments” 3 The user could attach some pictures to this media object. 4 The user clicks “OK” to save this operation
Expected results	graphical annotations and comments of the media object can be added
Variations	None
Issues	None
Test case Scope/Type	GUI

3.5.3 Saving annotations and comments of the media object

TCId	TC3.5.3
Test case	saving annotations and comments of the media object
Initial conditions	The object editor and viewer is activated.

Configuration description	
Description of functionality to be tested	saving annotations and comments of the media object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user has already changed the annotations and comments of the media object and wants to save the result by clicking “saving”. 2 The user will choose the directory and the name of the file to record the information of the annotations and comments of this media object 3 The comments and annotations are saved on the disk and not into the AXMEDIS objects if now decided by the user and if the user has the right to do it.
Expected results	annotations and comments of the media object can be saved
Variations	None
Issues	None
Test case Scope/Type	GUI

3.5.4 Removing annotations and comments of the media object

TCId	TC3.5.4
Test case	Removing annotations and comments of the media object
Initial conditions	The object editor and viewer is activated.
Configuration description	
Description of functionality to be tested	Removing annotations and comments of the media object
Partners, people involved	End User, Content Integrator, Content Distributor, Content Consumer
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The user wants to delete the current annotations and comments of media object by clicking “removing” 2 The annotations and comments will disappear and the file which record these annotations and comments will be deleted from the disk.
Expected results	annotations and comments of the media object can be removed
Variations	None
Issues	None

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

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

TCId	TC3.6.1
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 contact the External Editor AXMEDIS Plug-in 5.1 Authenticate the plug-in (which is responsible of authenticating the underlying External Editor) 5.2 Negotiate the "enforceable" rights 5.3 If negotiation is not satisfying (i.e. cannot enforce save as…) 5.3.1 The system displays an verification failure message on screen 5.3.2 The Test Case ends 6 AXMEDIS Editor saves on disk an AXMEDIS Object with the interested resource in a protected form 7 The external tool load the AXMEDIS Object and if user is authorized 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 The Editor shows failure messages if the plug-in is not able to enforce some rights.
Variations	None
Issues	None
Test case Scope/Type	GUI, BlackBox

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

TCId	TC3.6.2
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

3.6.3 Closing an External Tool Session

TCId	TC3.6.3
Test case	Closing External Tool Session
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

3.6.4 Updating a digital resource modified by an External Tool

TCId	TC3.6.4
Test case	Updating a digital resource modified by 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 is using the AXMEDIS plug-in
Description of functionality to be tested	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	5 The actor wants to update the resource, being modified in the external tool, in the AXMEDIS Editor 6 The user clicks with left mouse button on the “update” button of the external tool menu 7 The digital resource id saved as an AXMEDIS Object by the AXMEDIS plug-in 8 The user click on “refresh” regarding the resource 9 The AXMEDIS Editor loads the resource including in the container object
Expected results	The resource in the AXMEDIS object 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

4 AXMEDIS Plug in Definition (DSI Bruno)

4.1 Defining a AXCP Plugin

TCId	TC4.1
Test case	Defining a AXCP plugin
Initial conditions	The actor has identified a set of content processing algorithms to be developed and works with the AXMEDIS Framework. The actor uses the xml schema related to edit the profile of functions that the plugin will expose
Configuration Description	• AXCP Rule Editor • AXMEDIS Plugin Manager
Description of functionality to be tested	• The AXCP plugin is recognised by the AXMEDIS Plugin manager
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Computer Programming, GUI user
Data set used	
Steps	1 The Actor develops a library of functions according to the identified content processing algorithms 2 The actor starts to define and implement a dynamic library for the development of AXMEDIS Plugin 3 For each functions of the library 3.1 The Actor defines and develops an AXCP Function class. The class has to be called using the name of the native content processing function 3.2 The Actor maps the set of in/out parameters of the native function signature into the AXCP Parameter classes according to data types 3.3 The Actor implements a method called “execute” where he puts the call to the native content processing function of library 4 The Actor implements an entry function that manages a selector of single instances of AXCP Functions 5 The Actor has to exports the entry function as entry point of a dynamic library 6 The Actor has to edit an xml file describing each function of the plugin 7 The Actor puts the plugin and the xml in the plugin directory of the AXCP tools
Expected results	• A *dll* or *so* library is created • An xml profile has been associated with the plugin • The AXCP plugin is recognised by the AXMEDIS Plugin manager • The xml is displayed in the AXCP Rule Editor
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5 AXMEDIS Production Tools (DSI, EXITECH, EPFL)

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

5.1.1 AXMEDIS Content Processing Engine (WP4.3.1: DSI, WP5.4.1: DSI)

5.1.1.1 Firing an AXCP rule

TCId	TC 5.1.1.1
Test case	Firing an AXCP rule

Initial conditions	The Internal Scheduler of the AXCP Rule Engine has almost an activated AXCP rule
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	The TC describes steps related to the firing of a rule in the AXCP Rule Engine
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1 The Internal Scheduler periodically checks if the firing condition of rules are verified. 2 If the firing conditions are verified 2.1 A Remote Executor is associated with the AXCP rule 2.2 The rule is sent to the Executor to be run
Expected results	The fired rule is running on the Remote Executor
Variations	The AXMEDIS Workflow manager sends a request of running a specific AXCP rule. The association of Remote Executor with the rule is performed by matching the required capabilities by the rule with the executor profile capabilities
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.2 Searching for a rule Executor

TCId	TC 5.1.1.2
Test case	Searching for a rule Executor
Initial conditions	AXCP Rule Executors are running in the GRID Environment
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	The TC describes steps related to the discovering of an AXCP Rule Executor in the AXCP Rule Engine Grid Environment
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The internal scheduler performs periodically a network exploration in order to discover new AXCP Remote Executor 2 If a new executor is found 2.1 the Remote Executor is put into the list of available executor
Expected results	A new AXCP Remote Executor is available
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.3 Automatic production (DSI: Bruno)

TCId	TC 5.1.1.3

Test case	Automatic production
Initial conditions	An active content processing rule is ready to be executed
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	Some AXMEDIS objects are created automatically
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1. The Use Case begins when the Internal Scheduler of the AXCP Rule Engine activates a rule from the AXCP Rules List. 2. The Internal Scheduler sends a Rule execution request and the corresponding rule to the selected AXCP Rule Executor. 3. The AXCP Rule Executor executes the submitted rules by: 3.1. recovering all the specified AXMEDIS objects from AXMEDIS Database 3.2. verifying the compatibility of DRM and licensing 3.3. compounding AXMEDIS objects as described into selected rule 3.4. interacting with Formatting, Fingerprint, Adaptation and Protection tools 3.5. storing all new created AXMEDIS objects into AXMEDIS Database (AXMEDIS Objects repository) 4. If the AXCP Rule Executor runs correctly the rule 4.1. sends an End Process notification to the Internal Scheduler 5. otherwise it sends the occurred errors to the Internal Scheduler 6. The Test Case ends
Expected results	AXMEDIS Objects are created or error notifications are sent back
Variations	• The AXCP Rule Engine receives a request coming from the AXMEDIS Workflow Manager. • The AXCP Rule Engine send a notification or the occurred error to the AXMEDIS Workflow Manager. • The formatting process can be executed by an external tool if specified in the formatting rule
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.4 Verification of the compatibility of DRM associated with digital resources (DSI: Bruno)

TCId	TC 5.1.1.4
Test case	Verification of the compatibility DRM associated with digital resources
Initial conditions	
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	An AXCP rule could include the verification request of DRM rules related to all digital resources with a DRM target specified by the Content Integrator. In this case The AXCP Engine verifies that DRM rules are compatible with the DRM rules and/or conditions specified in the rule
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user

Data set used	AXDS-IVE2
Steps	1 The Use Case starts when the AXCP Rule Executor has to verify if the set of DRM rules match the DRM target specified in the rule. 2 If DRM are not compatible with the DRM and/or conditions specified in the rule. 2.1 The execution fails and a failure notification is generated 2.2 The Use Case ends. 3 The AXCP Rule Executor continues the rule execution 4 The Test Case ends The Use Case starts when the AXCP Rule Executor has to verify if the set of DRM rules match the DRM target specified in the rule. 5 If DRM are not compatible with the DRM and/or conditions specified in the rule. 5.1 The execution fails and a failure notification is generated 5.2 The Use Case ends. 6 The AXCP Rule Executor continues the rule execution 7 The Test Case ends
Expected results	The current composition is interrupted.
Variations	The AXCP Rule Engine send a notification or the occurred error to the AXMEDIS Workflow Manager.
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.5 Verification of rights for digital resources (FUPF DSI: Bruno)

TCId	TC 5.1.1.5
Test case	Verification of rights for digital resources
Initial conditions	The DRM rules of digital resources related to the Selection of objects are available
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	An AXCP rule could include the verification request of rights related to all digital resources. In this case, the AXCP Engine verifies that rights are compatible with the rights target specified in the rule.
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-IVE2
Steps	1 The Use Case starts when the AXCP Rule Executor has to verify if the set of rights match the rights specified in compositional rule. 2 If rights are not compatible with the rights specified in the rule. 2.1 The execution fails and a failure notification is generated 2.2 The Use Case ends. 3 The AXCP Rule Executor continues the rule execution 4 The Test Case ends
Expected results	The current composition is interrupted.
Variations	The AXCP Rule Engine send a notification or the occurred error to the AXMEDIS Workflow Manager.
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.6 Embedding a digital resource in the new AXMEDIS object (DSI: Bruno)

TCId	TC 5.1.1.6
Test case	Embedding a digital resource in the new AXMEDIS object
Initial conditions	Rules for creating AXMEDIS Object are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	AXCP Rule Engine embeds physically or by reference one o more digital resource in the new AXMEDIS object.
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1 The Use Case starts when the AXCP Rule Executor has to embed a digital resource in the new AXMEDIS object 2 If the embedding option is “physically” 2.1 The rule executor sends an embedding request and the resource to the AXOM 3 Else the composition engine sends an embedding request and the reference of resource to the AXOM 4 The resource is embedded 5 The Test Case ends
Expected results	The resource or the reference is embedded in the AXMEDIS object
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.7 New AXMEDIS objects generation (DSI: Bruno)

TCId	TC 5.1.1.7
Test case	New AXMEDIS object generation
Initial conditions	Rules for creating AXMEDIS Object are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	AXCP Rule Engine creates one o more new AXMEDIS objects and assigns them a new Object ID
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1 The Use Case starts when the AXCP Rule Executor creates a new AXMEDIS object following an AXCP rule 2 The AXCP Rule Executor asks for a new Object ID to the AXMEDIS OID Generator. 3 The ID is applied to the new object. 4 The Test Case ends
Expected results	The Object is created and a new ID has been assigned

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

5.1.1.8 Fingerprint estimation of a digital resource (DSI: Bruno)

TCId	TC 5.1.1.8
Test case	Fingerprint estimation of a digital resource
Initial conditions	The digital resource is available physically Rules for creating AXMEDIS Object are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	If a fingerprint request for a digital resource is specified in the AXCP rule the AXCP Rule Engine interacts with the Fingerprint tool asking for fingerprint estimation . The Fingerprint tool will return the content descriptors related to the digital resource.
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1 The Use Case starts when a fingerprint estimation request for the digital resource is specified in the AXCP rule. 2 The digital resource is sent to the Fingerprint tool 3 The Fingerprint tool returns the content descriptors associated with the digital resource. 4 The content descriptors are inserted as metadata associated with the digital resource 5 The Test Case ends
Expected results	The content descriptors are inserted as metadata associated with the digital resource
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.9 Formatting of AXMEDIS Objects (DSI: Vaccari, Bruno)

TCId	TC 5.1.1.9
Test case	Formatting of AXMEDIS Objects.
Initial conditions	• The AXMEDIS Object is available during the composition process; profiles for user preferences, device capabilities and delivery context should also be available. • The digital resource is available physically • Rules for creating AXMEDIS Object are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	If an automatic formatting request for an AXMEDIS Object is specified in the AXCP rule, the AXCP Rule Engine interacts with the Format Tool. The Format Tool will return the format descriptors most suitable for the digital resources contained in the Object.

Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1. The Use Case starts when an automatic format request for an AXMEDIS Object is specified in the AXCP rule. 2. The AXMEDIS Object is sent to the Format Tool 3. Profiles for user preferences, device capabilities and delivery context are sent to the Format Tool 4. The AXCP Rule Executor asks the Format Tool for automatically selecting a template. 5. The Format Tool returns an ordered list of descriptors for most suitable templates. 6. The AXCP Rule Executor asks the Format Tool for automatically selecting a style-sheet for the first template of the list. 7. The Format Tool returns an ordered list of descriptors for most suitable style-sheets. 8. The AXCP Rule Executor asks the Format Tool for optimizing the first style-sheet of the list. 9. The Format Tool returns the descriptor of the optimized style-sheet. 10. The AXCP Rule Executor requests the creation of the resulting format. 11. The resulting format descriptor is associated with the Object. 12. The Test Case ends
Expected results	A format descriptor is associated with the Object.
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.10 Adaptation of a digital resource (DSI: Bruno)

TCId	TC 5.1.1.10
Test case	Adaptation of a digital resource
Initial conditions	• The AXMEDIS Object is available during the composition process; profiles for user preferences, device capabilities and delivery context should also be available. • The digital resource is available physically • Rules for adapting AXMEDIS Object are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	If an adaptation request for a digital resource is specified in the AXCP rule the AXCP Rule Engine interacts with the Adaptation tool. The Adaptation tool will perform the adaptation specified in the composition rule for the a given digital resource.
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1 The Use Case starts when an adaptation request for the digital resource is specified in the AXCP rule 2 The digital resource is sent to the adaptation tool 3 The Adaptation tool returns the adapted resource 4 The Use Case ends

Expected results	The initial digital resource is adapted on the basis of adaptation request specified in the rule
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.11 Protection of the new AXMEDIS object (DSI: Bruno)

TCId	TC 5.1.1.11
Test case	Protection of the new AXMEDIS object
Initial conditions	• The AXMEDIS Object is available during the composition process; • The digital resource is available physically • Rules for protecting AXMEDIS Objects are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	If a protection request for the new AXMEDIS object is specified in the AXCP rule the AXCP Rule Engine interacts with the Protection tool. The Protection tool will create an AXMEDIS protected object.
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-PTE1, AXDS- PTE2
Steps	1 The Test Case starts when the rule executor has to perform a protection. 2 The digital resource is sent to the Protection tool 3 The Protection tool returns the protected content 4 The Test Case ends
Expected results	The Axmedis object is protected
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.12 Merging component's DRM/PAR rules into a new AXMEDIS object (FUPF, DSI: Bruno)

TCId	TC 5.1.1.12
Test case	Merging component's DRM/PAR rules into a new AXMEDIS object
Initial conditions	• The AXMEDIS Object is available during the composition process; • The digital resource is available physically • Rules for merging DRM/PAR of AXMEDIS Objects are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	AXCP Rule Engine create a new AXMEDIS objects and merge component's DRM/PAR rules to create a new DRM/PAR rule
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-IVE1, AXDS- IVE2
Steps	1 The Use Case starts when the AXCP Rule Executor has to generate the DRM/PAR for the composite AXMEDIS object 2 AXCP Rule Executor merges component's DRM/PAR rules into the new AXMEDIS Objects

	3 The Use Case ends
Expected results	Component's DRM/PAR are merged
Variations	• The DRM/PAR of the whole new object depends on what could be the intersection of DRM/PAR rules related to each component • The actor can modify the DRM/PAR rule of the new AXMEDIS object without modify the DRM/PAR rules of the components, but they have to respect the existing. The verification can be invoked by exploiting services of the PMS client in conjunction with the AXOM
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.1.13 External Tools execute formatting operations (DSI: Vaccari, Bruno)

TCId	TC 5.1.1.13
Test case	External Tools execute formatting operations
Initial conditions	• The AXMEDIS Object is available during the composition process; profiles for user preferences, device capabilities and delivery context should also be available. • The digital resources to be formatted are available physically during the formatting process • Rules for formatting AXMEDIS Object are running
Configuration Description	The AXCP Rule Scheduler of the AXCP Rule Engine is running Some AXCP Rule Excutors are running on GRID
Description of functionality to be tested	If a request of services provided by formatting external tools is specified in the AXCP rule the AXCP Rule Engine will interact with the External Formatting tools. The External Tools will format or perform specific script languages. The external tool will be able to perform also adaptations specified in the formatting rule for digital resources.
Partners, people involved	Content Integrator, Content Distributor, Content Creator
Validator skill	Production editing, GUI user
Data set used	AXDS-Formatting4, AXDS-Composition4
Steps	1 The Test Case starts when the rule executor has to perform an external call to a formatting tool specified in the rule 2 The Rule Executor sends the digital resources and parameters specified in the external call to the external tool 3 The external tool performs functions specified in the rule 4 The external tool returns formatted digital resources 5 The Test Case ends
Expected results	The AXMEDIS Object is formatted according to the rule
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

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

5.1.2.1 Create a new AXCP rule (DSI: Bruno)

TCId	TC5.1.2.1
Test case	Create a new AXCP rule
Initial conditions	An empty AXCP rule is ready on the Rule Editor
Configuration	AXCP Rule Editor is running

Description	
Description of functionality to be tested	Editing of a new rule
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	Production editing, GUI user, Javascript knowledge
Data set used	
Steps	1 The Actor creates a Selection of digital resources based on queries to the AXMEDIS Database 2 The Actor defines the set of parameters necessary to run the rule 3 The Actor defines the plugins/external tools to be used 4 The Actor rules how these resources have to be processed 5 The Actor stores the created rule into the Rule repository
Expected results	An AXCP Rule is stored
Variations	1) The Actor defines a Selection by writing in the rule the scripting code (AXCP Rule Language) for queries to be executed when the rule will be run 2) The Actor can define a rule or writing it as scripting code (AXCP Rule Language) or in a Visual way.
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.2.2 Search and Select an AXCP rule (DSI: Bruno)

TCId	TC5.1.2.2
Test case	Search and Select an AXCP rule
Initial conditions	
Configuration Description	AXCP Rule Editor is running
Description of functionality to be tested	An Actor wants to select a specific compositional rule he should be enabled to make some search or browsing, they are organized in some ordering.
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The Actor search into the Repository of AXCP Rules a specific AXCP rule 2 The rules are ordered in some manner and simple queries can be performed 3 If the Actor founds the rule can : 3.1 Use it to create a compounded AXMEDIS object 3.2 Modify it 3.2.1 Then the Actor stores the new rule into the Repository by AXCP Rule Editor 3.2.2 Use the new rule to create a compounded AXMEDIS object 4 If the Actor doesn't found the rule can create a new oneThe Actor search into the Repository of AXCP Rules a specific AXCP rule 5 The rules are ordered in some manner and simple queries can be performed 6 If the Actor founds the rule can : 6.1 Use it to create a compounded AXMEDIS object 6.2 Modify it 6.2.1 Then the Actor stores the new rule into the Repository by AXCP Rule Editor 6.2.2 Use the new rule to create a compounded AXMEDIS object

	7 If the Actor doesn't found the rule can create a new one
Expected results	The Selected rule is opened into the rule editor
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.2.3 Activating an AXCP rule (DSI: Bruno)

TCId	TC5.1.2.3
Test case	Activating an AXCP rule
Initial conditions	The AXCP Rule Editor can access to the AXCP Rules Scheduler
Configuration Description	AXCP Rule Editor is running AXCP Rules Scheduler is running
Description of functionality to be tested	Installation and Activation of a AXCP Rule on the AXCP Rule Scheduler via Rule Editor
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	Production editing, GUI user
Data set used	
Steps	1 The Actor searches into the Repository of Rules a specific AXCP rule 2 If the Actor doesn't found the rule 2.1 The Actor can create a new one 3 The Actor selects "Activate" function 4 The rule is sent to the Active Rules Repository of the AXCP Rule Engine
Expected results	The AXCP Rule is intalled into the Scheduler
Variations	None
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

5.1.2.4 Debugging/simulation of an AXCP rule (DSI: Bruno)

TCId	TC5.1.2.4
Test case	Debugging/Simulation of an AXCP rule
Initial conditions	The script associated with the AXCP rule is available in the Rule Editor
Configuration Description	AXCP Rule Editor is running
Description of functionality to be tested	Debug or simulate of an AXCP rule
Partners, people involved	Content owner, Content Integrator, Content Distributor
Validator skill	Production editing, GUI user, Javascript knowledge
Data set used	
Steps	1 The Test Case starts when the Actor press "F5" or call "Start debug" 2 The Rule Editor enters in the Debugging/Simulation Mode 3 During the debugging mode the Actor can: 3.1 Check the statements of script step by step 3.2 Control the values of current variables 3.3 Add/Remove breakpoints by "F9" function key or "Insert/Remove Breakpoint" 3.4 "Step Over" lines and "Trace Into" functions ("F10" and "F11" function

	keys) 3.5 Exit from the debugging mode with “Stop Debug” function (“Shift + F5” shortcut)
Expected results	
Variations	The AXCP rule is simulated during the debugging session in order to extract information regarding the complexity, CPU request, workload, etc....
Issues	None
Test case Scope/Type	GUI/ backend, BlackBox

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

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

5.2.1.1 Automatic formatting process (DSI: Vaccari)

TCId	TC 5.2.1.1
Test case	Automatic formatting process
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available; • A Rule that needs Object formatting is executed by the AXCP Rule Engine.
Configuration description	
Description of functionality to be tested	This test-case tests the correct execution of all the steps involved in the automatic formatting process.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the script language and the SMIL language.
Data set used	
Steps	1. The test begins when a Rule sends a formatting request to the Format Tool. 2. The AXCP Rule Executor performs an automatic template selection (see TC 7.2.12). 3. The AXCP Rule Executor performs an automatic style-sheet selection and optimization (see TC 7.2.1.3). 4. The AXCP Rule Executor performs an automatic style-sheet optimization (see TC 7.2.1.4). 5. The AXCP Rule Executor performs a format creation (see TC 7.2.1.5). 6. The system creates the format descriptor.
Expected results	The system has to produce a valid SMIL document, which correctly includes all (or most of) the resources contained in the AXMEDIS Object. The SMIL document has to be suitable for the given profile descriptors.
Variations	• Profiles may be available or not.
Issues	
Test case Scope/Type	Backend/White box

5.2.1.2 Automatic template selection (DSI: Vaccari)+

TCId	TC 5.2.1.2
Test case	Automatic template selection
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user

	preferences, device capabilities and delivery context should also be available; • A Rule that needs a template selection is executed by the AXCP Rule Engine.
Configuration description	
Description of functionality to be tested	This test-case tests if the system correctly select templates suitable for the given resources and profiles.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the script language and the SMIL language.
Data set used	
Steps	1. The test begins when a Rule sends a template selection request to the Format Tool. 2. The AXCP Rule Executor provides resource references. 3. The AXCP Rule Executor may provide Profile references (User, Device and Context Profiles) 4. The AXCP Rule Executor may provide information about the type of document to be created. 5. The AXCP Rule Executor may provide information about the preferred output format. 6. The AXCP Rule Executor may provide information about the target device(s). 7. The AXCP Rule Executor may provide indications about the category of each resource. 8. The system produces an ordered list of descriptors for best templates.
Expected results	The system has to select one or more templates suitable for the given resources and profiles: all (or most of) the resources are used by a template that meets profiles and preferences.
Variations	• A template ID may be directly provided to the system; • Profiles may be provided or not; • Information about the type of document to be created may be provided or not; • Information about the output format may be provided or not; • Information about the target device may be provided or not; • Information about resources category may be provided or not
Issues	
Test case Scope/Type	Backend/Black box

5.2.1.3 Automatic style-sheet selection (DSI: Vaccari)

TCId	TC 5.2.1.3
Test case	Automatic style-sheet selection
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available; • A template has been selected; • A Rule that needs a template selection is executed by the AXCP Rule Engine.
Configuration description	
Description of functionality to be tested	This test-case tests if the system correctly select style-sheets suitable for the given template, resources and profiles.
Partners, people involved	Content owner, Content Integrator, Content Distributor.

Validator(s) skill	An appropriate familiarity with the script language and the SMIL language; a good knowledge of XSLT.
Data set used	
Steps	1. The test-case begins when the Rule sends a style-sheet selection request to the Format Tool. 2. The AXCP Rule Engine provides resource references. 3. The AXCP Rule Engine may provide Profile references (User, Device and Context Profiles). 4. The AXCP Rule Engine may give information about the target device(s). 5. The system produces an ordered list of descriptors for best style-sheets.
Expected results	The system has to select one or more style-sheets suitable for the given template, resources and profiles.
Variations	• A style-sheet ID may be directly provided to the system; • Profiles may be provided or not; • Information about the target device may be provided or not.
Issues	
Test case Scope/Type	Backend/Black box

5.2.1.4 Automatic style-sheet optimization (DSI: Vaccari)

TCId	TC 5.2.1.4
Test case	Automatic style-sheet optimization
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available; • A template has been selected; • A style-sheet has been selected; • A Rule that needs a style-sheet optimization is executed by the AXCP Rule Engine.
Configuration description	
Description of functionality to be tested	This test-case tests if the system provide optimized values for parameters defined in a style-sheet.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the script language and the SMIL language; a good knowledge of XSLT.
Data set used	
Steps	1 The test-case begins when the Rule sends a style-sheet optimization request to the Format Tool. 2 The AXCP Rule Engine provides resource references. 3 The AXCP Rule Engine may provide Profile references (User, Device and Context Profiles). 4 The AXCP Rule Engine may give information about the target device(s). 5 The system produces optimized values for parameters defined in the style-sheet.
Expected results	Values produced have to be suitable for the style-sheet optimization according to the given profiles.
Variations	• Information about profiles may be provided or not. • Information about the target device may be provided or not.
Issues	

Test case Scope/Type	Backend/Black box

5.2.1.5 Format creation (DSI: Vaccari)

TCId	TC 5.2.1.5
Test case	Format creation
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available; • A template has been selected; • A style-sheet has been selected (and optimized, if needed); • A Rule that needs a style-sheet optimization is executed by the AXCP Rule Engine.
Configuration description	
Description of functionality to be tested	This test-case tests if the system correctly create a SMIL document using the given template and style-sheet.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the script language and the SMIL language.
Data set used	
Steps	1 The test-case begins when the Format Tool receives a format creation request (from a Rule or through the AXEditor). 2 The system receives resource references. 3 The system receives template and style-sheet descriptors. 4 The system may receive optimized value for parameters defined in the style-sheet. 5 The system produces a format descriptor.
Expected results	The system has to produce a valid SMIL document.
Variations	• The style-sheet may be parameterized or not; in the first case optimized values are provided.
Issues	
Test case Scope/Type	Backend/Black box

5.2.2 Interactive Formatting Tools (WP4.3.2: DSI, WP5.4.2: DSI)

5.2.2.1 Interactive formatting process (DSI: Vaccari)

TCId	TC 5.2.2.1
Test case	Interactive formatting process
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available.
Configuration description	
Description of functionality to be tested	This test-case tests the correct execution of all the steps involved in the interactive formatting process.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the GUI and with the SMIL language.

Data set used	
Steps	1 The test-case begins when the Actor starts the AXEditor for formatting. 2 The Actor performs an interactive template selection (see TC 7.2.2.2). 3 The Actor performs an interactive style-sheet selection (see TC 7.2.2.3). 4 The Actor performs an interactive style-sheet optimization (see TC 7.2.2.4). 5 The Actor performs a format creation (see TC 7.2.1.5). 6 The system allows previewing of the resulting document.
Expected results	The system has to produce (and preview) a valid SMIL document, which correctly includes all (or most of) the resources contained in the AXMEDIS Object. The SMIL document has to be suitable for the given profile descriptors.
Variations	• Profiles may be available or not
Issues	
Test case Scope/Type	GUI/White box

5.2.2.2 Interactive template selection (DSI: Vaccari)

TCId	TC 5.2.2.2
Test case	Interactive template selection
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available.
Configuration description	
Description of functionality to be tested	This test-case tests if the the AXEditor allows the selection of a ready-made template suitable for the given resources.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the GUI and with the SMIL language.
Data set used	
Steps	1 The test-case begins when the Actor sends a request for template selection to the AXEditor. 2 The Actor provides resource references. 3 The Actor may provide Profile references (User, Device and Context Profiles) 4 The Actor may give information about the type of document to be created. 5 The Actor may give information about the preferred output format. 6 The Actor may give information about the target device(s). 7 The Actor may give indications about the category of each resource. 8 The AXEditor proposes the ordered list of best templates.
Expected results	The AXEditor has to select one or more templates suitable for the given resources and profiles: all (or most of) the resources are used by a template that meets profiles and preferences.
Variations	• A template ID may be directly provided to the system; • Profiles may be provided or not; • Information about the type of document to be created may be provided or not; • Information about the output format may be provided or not; • Information about the target device may be provided or not; • Information about resources category may be provided or not
Issues	
Test case Scope/Type	GUI/Black box

5.2.2.3 Interactive style-sheet selection (DSI: Vaccari)

TCId	TC 5.2.2.3
Test case	Interactive style-sheet selection
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available; • A template has been selected.
Configuration description	
Description of functionality to be tested	This test-case tests if the AXEditor allows selection of style-sheets suitable for the given template, resources and profiles.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the GUI and with the SMIL language; a good knowledge of XSLT.
Data set used	
Steps	1 The test-case begins when the Actor selects a template in the AXEditor. 2 The Actor may provide Profile references (User, Device and Context Profiles). 3 The Actor may give information about the target device(s). 4 The AXEditor proposes the ordered list of best style-sheets suitable for the given template: the Actor can read their description and metadata and choose the best one.
Expected results	The AXEditor has to select one or more style-sheets suitable for the given template, resources and profiles.
Variations	• A style-sheet ID may be directly provided to the system; • Profiles may be provided or not; • Information about the target device may be provided or not.
Issues	
Test case Scope/Type	GUI/Black box

5.2.2.4 Interactive style-sheet optimization (DSI: Vaccari)

TCId	TC 5.2.2.4
Test case	Interactive style-sheet optimization
Initial conditions	• An AXMEDIS Object has been selected and descriptors for its digital resources are accessible to the Format Tool; profile descriptors for user preferences, device capabilities and delivery context should also be available; • A template has been selected; • A style-sheet has been selected.
Configuration description	
Description of functionality to be tested	This test-case tests if the AXEditor provide optimized values for parameters defined in a style-sheet.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the GUI and with the SMIL language; a good knowledge of XSLT.
Data set used	
Steps	1 The test-case begins when the AXEditor receives a style-sheet optimization

	request. 2 The system receives resource references. 3 The system may receive Profile references (User, Device and Context Profiles). 4 The system may receive information about the target device(s). 5 The system produces optimized values for parameters defined in the style-sheet.
Expected results	Values produced have to be suitable for the style-sheet optimization according to the given profiles.
Variations	• Information about profiles may be provided or not. • Information about the target device may be provided or not.
Issues	
Test case Scope/Type	GUI/Black box

5.2.2.5 Template creation (DSI: Vaccari)

TCId	TC 5.2.2.5
Test case	Template creation
Initial conditions	• An AXMEDIS Object has been selected and its resources are physically accessible to the AXMEDIS SMIL Editor; otherwise, a set of resource descriptors has to be accessible. In the latter case, “fake” media files are used to preview the format in the Editor.
Configuration description	
Description of functionality to be tested	This test-case tests if the AXEditor creates a valid template.
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the GUI and with the SMIL language.
Data set used	
Steps	1 The test-case begins when the Actor starts the AXMEDIS SMIL Editor. 2 The Actor creates a SMIL document which includes the resources contained within the AXMEDIS Object. 3 The Actor exports the document as “AXMEDIS Template”.
Expected results	The AXMEDIS SMIL Editor has to create a valid template using a SMIL document.
Variations	
Issues	
Test case Scope/Type	GUI/Black box

5.2.2.6 Style-sheet creation (DSI: Vaccari)

TCId	TC 5.2.2.6
Test case	Style-sheet creation
Initial conditions	• An AXMEDIS Object has been selected and its resources are physically accessible to the AXMEDIS SMIL Editor; otherwise, a set of resource descriptors has to be accessible. In the latter case, “fake” media files are used to preview the format in the Editor. • At least one template for the current document has already been created.
Configuration description	This test-case tests if the AXEditor creates a valid style-sheet.

Description of functionality to be tested	
Partners, people involved	Content Distributor.
Validator(s) skill	An appropriate familiarity with the GUI and with the SMIL language; a good knowledge of XSLT.
Data set used	
Steps	1. The test-case begins when the Actor starts the AXMEDIS SMIL Editor. 2. The Actor creates a SMIL document which includes the resources contained within the AXMEDIS Object. 3. The Actor marks some attributes as input for the optimization logic of the Format Tool. 4. The Actor choices a reference template. 5. The Actor exports the document as “AXMEDIS Style-sheet”.
Expected results	The AXMEDIS SMIL Editor has to create a valid XSLT style-sheet, suitable for the given template, using a SMIL document.
Variations	
Issues	
Test case Scope/Type	GUI/Black box

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

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 Component to NPD

TCId	TC6.1.1.2
Test case	Add Component to 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 “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 Information in NPD

TCId	TC6.1.1.3
Test case	Edit Information in NPD 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 Delete Information in NPD

TCId	TC6.1.1.4
Test case	Delete Information in NPD
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 functionality to be tested	This is a generic use case responsible for removing anything from the NPD. e.g. 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 Show Information regarding component of NPD

TCId	TC6.1.1.5
Test case	Show information regarding component of 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 "show"
Description of functionality to be tested	This use case is responsible for showing information related to various components, their copyrights, DRM, History (metadata, timestamp, version), 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

6.1.1.6 Delete a NPD

TCId	TC6.1.1.6
Test case	Delete a 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

6.1.1.7 Search a NPD

TCId	TC6.1.1.7
Test case	Search a NPD
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

6.1.1.8 Track Component

TCId	TC6.1.1.8
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

6.1.1.9 Timestamp Generator

TCId	TC6.1.1.9
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

6.1.1.10 List Work

TCId	TC6.1.1.10
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

6.1.1.11 Select a work item from the list of work

TCId	TC6.1.1.11
Test case	Select a work item from the list of work
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

6.1.1.12 Complete a Task of a WorkItem

TCId	TC6.1.1.12
Test case	Complete a task of Work Item
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

6.1.1.13 Change State/Phase of Task of WorkItem

TCId	TC6.1.1.13
Test case	Change State/Phase of Task of WorkItem
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

6.1.1.14 Notification of Information to a personnel for a task of Work Item

TCId	TC6.1.1.14
Test case	Notification of Information to a personnel for a task of Work Item
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

6.1.1.15 Global Viewer of all Information of NPD

TCId	TC6.1.1.15
Test case	Global viewer of all Information of 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 “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

6.1.1.16 Check-in Task Performed by Manual Operator

TCId	TC6.1.1.16
Test case	Check-in Task Performed by Manual Operator 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

6.1.1.17 Check-out Task Performed by Manual Operator

TCId	TC6.1.1.17
Test case	Check-out Task Performed by Manual Operator 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 Setup for metadata mapping (UNIVLEEDS)

TCId	TC7.1.1
Test case	Generating a mapping file which can be used for metadata adaptation
Initial conditions	Two XML files with the metadata. One as the source metadata and one as the targeted metadata
Configuration description	No configuration necessary
Description of functionality to be tested	This test case checks that the XSLT document is generated according to the mappings created in the GUI
Partners, people involved	General user including content producers, integrators, particularly user of the automated content gathering and adaptation
Validator skill	Basic window application usage
Data set used	Example files are available in the samples directory
Steps	1 Load the source XML metadata 2 Load the target XML metadata 3 Connect elements from the source to the target 4 Select ‘Save XSL’ from the file menu
Expected results	An XSLT document will be created which can be used to transform metadata in the source language according to the mappings defined by the user
Variations	None
Issues	None
Test case Scope/Type	GUI

7.1.2 Setup for content crawling

TCId	TC7.1.2
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 (ODBC, XML

	files, Lobster, Web Services)
Issues	None
Test case Scope/Type	Backend, black box

7.1.3 Creating objects from the Collected Content

TCId	TC7.1.3
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

7.2 Fingerprint extractor as a collection of collector engine plug-ins for extracting features

7.2.1 Calculating content descriptors/fingerprint (during crawling)

TCId	TC7.2.1
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 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 Querying for AXMEDIS from clients

Since the user interface can be web based (as hypothesized in some TC) but can also be a C++ user interface embedded in a client application, also in this case the TC is valid apart from the fact that it has to be tested with a Unit Test instead of a BlackBox approach.

8.2.3 Bookmark a query

TCId	TC8.2.3
Test case	Bookmark a query
Initial conditions	None
Configuration description	AXMEDIS Query 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

8.2.4 Retrieve a bookmarked query

TCId	TC8.2.4
Test case	Retrieve a bookmarked query
Initial conditions	None
Configuration description	AXMEDIS Query 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

8.2.5 Organize bookmarked queries

TCId	TC8.2.5
Test case	Organize bookmarked query
Initial conditions	None
Configuration description	AXMEDIS Query User Interface for Application
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

8.2.6 Save an incomplete query

TCId	TC8.2.6
Test case	Save an incomplete query
Initial conditions	None
Configuration description	AXMEDIS Query User Interface for application
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

8.2.7 Retrieve an incomplete query

TCId	TC8.2.7
Test case	Retrieve an incomplete query
Initial conditions	None
Configuration description	AXMEDIS Query User Interface for application
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 (DSI)

9.1 AXEPTool for P2P on B2B (DSI)

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 computer machine. The tool is registered and certificates have been already obtained. This Test is applied only if a discovery technology is used for the implementation of the P2P network.
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 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”, percentage of download and the guessed time of completion are provided. 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 (DSI)

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	The Publication Tool Engine is based on the AXCP and allows the user to execute publication rules in two ways: by scheduler and from the Rule Editor User Interface
Configuration description	AXMEDIS Data Base, AXEPTool, AXCP as Publication Tool Engine (and Publication/Loading Rules/Selections Interface), AXEPTool OUT AXMEDIS Data Base via web service.
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, define the rule and its metadata 2 The user may activate the rule sending it to the scheduler, the rule can be active on the basis of changes or periodic
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 AXCP Scheduler and engine as 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 AXCP detects that the object has been Updated and public it again on the P2P
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 Manual publication of AXMEDIS Objects with the AXEPTool

TCId	TC9.2.4
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

9.2.5 Producing a query to search on the AXEPTool network

TCId	TC9.2.5
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

9.2.6 View/Manage query results coming from the AXEPTool

TCId	TC9.2.6
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

9.2.7 Active query pool management for the AXEPTool

TCId	TC9.2.7
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.

9.2.8 Downloading an AXMEDIS object

TCId	TC9.2.8
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

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

TCId	TC9.2.9
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

9.2.10 Selecting objects for the AXDB from those downloaded

9.2.11 Automatic loading new versions of AXMEDIS Objects for the AXEPTool

TCId	TC9.2.11
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	Developers, Testers

involved	
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

9.2.12 Automatic loading new AXMEDIS Objects with the AXEPTool

TCId	TC9.2.12
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

9.2.13 Manual loading of AXMEDIS Objects with AXEPTool

TCId	TC9.2.13
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

9.2.14 Creation of a loading rule for the AXEPTool

TCId	TC9.2.14
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

9.2.15 Preview an AXMEDIS object content coming from AXEPTool

TCId	TC9.2.15
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	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

tested	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

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

10.1 Programme and publication programme 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. P&P Repository is accessible for saving created P&P Programmes 2. Query support can provide list of objects available for distribution or formatting and distribution
Configuration description	1. P&P Repository directory is specified in the configuration 2. AXMEDIS SELECTION is specified in the configuration for the editor to access query support a. MAIN QUERY SUPPORT WSDL b. SELECTION ARCHIVE WSDL 3. AXEMDIS DATABASE is defined in the configuration for the P&P Editor to access AXOIDS AXMEDIS P&P DISTRIBUTORS is specified in the configuration for the programme manager to select predefined distribution channels
Description of functionality to be tested	New P&P 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	None, Create a new programme
Steps	1 Programme producer uses GUI to submit query for objects and is returned a list of results 2 Programme manager to select part/all/none of results using the GUI 3 Programme manager schedules as a new rule or updates an existing rule in the P&P Programme 4 Programme manager specify the distribution channel of this programme 5 Schedule is saved as a P&P Programme
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 programme editing

TCId	TC10.2
Test case	Editing and collection of P&P Programmes in the P&P Repository
Initial conditions	Set of P&P Programmes available for editing and loading in the P&P Repository, see TC10.1 for creating P&P Programmes
Configuration description	1. P&P Repository directory is specified in the configuration 2. AXMEDIS SELECTION is specified in the configuration a. MAIN QUERY SUPPORT WSDL b. SELECTION ARCHIVE WSDL 3. AXEMDIS DATABASE is defined in the configuration 4. AXMEDIS P&P DISTRIBUTORS is specified in the configuration

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 P&P Programme in the collection 2. The user selects and loads certain P&P Programme 3. The user edits the P&P Programme 4. The programme is saved back to the repository either by overwriting the existing programme or by creating a new programme
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 programme and publication programmes

TCId	TC10.3
Test case	Actor decides to “activate”, i.e. to publish, the programme
Initial conditions	1. Set of complete P&P Programmes defined and set as inactive (by default) 2. P&P Editor is connected to the P&P Engine
Configuration description	1. P&P Repository directory is specified in the configuration in AXMEDIS_P&P_EDITOR to access and save P&P Programmes 2. AXMEDIS P&P Engine is specified in the configuration for the P&P Editor to communicate with the P&P Engine
Description of functionality to be tested	Loading a P&P Programme with a complete set of distribution 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	P&P Programmes in the repository created in TC10.1 Creating P&P Programmes and TC10.2 Editing P&P Programmes
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 distribution 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 programme and publication programme from workflow

TCId	TC10.4
Test case	A Workflow user activates a P&P Programme on a P&P Engine by sending the

	P&P Engine an Activate request with a P&P Programme.
Initial conditions	1. The test P&P Programme is in the current directory. 2. Active P&P Engine. 3. P&P Engine connected to WF. Available WF (e.g. a dummy client) on the same machine for simplicity. If WF is on a different machine, validator skill will include basic XML editing to configure the P&P Engine to communicate with WF.
Configuration description	1. Engine is active and accessing correct system time. 2. AXMEDIS Workflow can communicate with the P&P Engine. AXMEDIS Workflow Plugin is loaded on start-up to set communication parameters, this requires the WF Plugin's directory to be defined in the configuration file. For simplicity of this TC, a dummy client for WF can be used.
Description of functionality to be tested	P&P Programme is launched by AXMEDIS WF dummy client. This tests the WF Plugin and SOAP communication functions.
Partners, people involved	Partners involved with distribution, object formatting and database management
Validator(s) skill	XML Editing if WF is running on a different machine
Data set used	\Applications\axpnpeng\doc\test\AXPnPProgramme4.pp
Steps	1 Run the P&P Engine 2 Run the Dummy WF application using the Command Prompt with the P&P programme as its option parameter 3 Check the P&P Engine command line feedback to confirm the launch of the P&P programme
Expected results	Programme loaded in the P&P Engine can be seen by viewing the P&P Engine console window
Variations	None
Issues	None
Test case Scope/Type	Backend, Blackbox

10.5 Processing of P&P Programme

TCId	TC10.5
Test case	A programme manager activates a P&P Programme using the P&P Editor to distribute on localhost
Initial conditions	1. A set of completed P&P Programmes loaded into the P&P Editor 2. AXMEDIS Player is available in the P&P Engine directory
Configuration description	1. P&P Repository directory is specified in the configuration in AXMEDIS_P&P_EDITOR to access and save P&P Programmes 2. P&P Engine is active and accessing correct system time. 3. AXMEDIS P&P Engine is specified in the configuration for the P&P Editor to communicate with the P&P Engine 4. Available distribution-channel profile including bandwidth to allow estimate of time for the actual delivery
Description of functionality to be tested	Delivery according to schedule of published programme (immediate delivery for On-Demand)
Partners, people involved	Partners involved with distribution, object formatting and database management
Validator(s) skill	Programme editing and partners involved with data queries.
Data set used	\Applications\axpnpeng\doc\test\AXPnPProgramme4.pp
Steps	1 Edit a P&P Programme fro activation using the P&P Editor 2 Activate the P&P Programme using "Activate" – the result can be

	view/validate by viewing the P&P Engine console window and also the P&P Monitor 3 Wait for the scheduled time 4 AXMEDIS player automatically load the specified object for playback
Expected results	AXMEDIS object is played on the local machine
Variations	• immediate delivery for On-Demand
Issues	AXMEDIS Player needs to be available and ActiveX running
Test case Scope/Type	Backend, Blackbox

10.6 Trial pre-activation of programme and publication programme

TCId	TC10.6
Test case	Actor decides to publish ("quick trial") or ("full trial") the P&P Programme
Initial conditions	Set of complete P&P Programmes defined and set as inactive
Configuration description	1. P&P Repository directory is specified in the configuration in AXMEDIS PNP EDITOR to access P&P Programmes saved in the P&P Repository 2. AXMEDIS P&P Engine is specified in the configuration for the P&P Editor to communicate with the P&P Engine.
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	P&P Programmes in the P&P repository on the AXMEDIS CVS associated with the P&P Editor Directory Applications/bin/pnpeditor/pnprepository/
Steps	1 Select and load a programme from the P&P Repository, for final checking 2 In the P&P Editor select Command→Programme Test→(Full Trial | Quick Trial) to activate/publish the trial programme as a quick or full trial 3 The component checks the status and required information of the programme. Request for editing if the programme is incomplete (e.g. unknown publication date or channel) 4 User receives 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

10.7 Monitoring of programme and publication engine

TCId	TC10.7
Test case	The P&P Engine Monitor monitoring the P&P Engine to allow a user to view active programme information.
Initial conditions	A P&P Engine is running on a machine that is accessible by the Monitor
Configuration description	Engine and Monitor must be configured to use the same port for communication
Description of functionality to be tested	Connecting to a P&P Engine; viewing active programmes; disconnecting from Engine
Partners, people involved	None
Validator(s) skill	1. Application setup 2. Programme editing

Data set used	Example programmes in axpnpeng test directory
Steps	1 Start P&P Engine 2 Start the P&P Editor 5 Activate the example programmes (taking care that the programme start times are valid) 6 Start the P&P Engine Monitor 7 Connect to the Engine 8 View the active programmes 9 Remove a programme from the Engine by setting it inactive in the P&P Editor 10 Update view (click 'refresh tree now') 11 Disconnect 12 Close all applications
Expected results	Once the monitor is connected, and refreshed it will display a tree containing active programmes and their next AXOBJECT start time. After a programme has completed on the Engine and the Monitor is refreshed, an updated list of programmes is displayed
Variations	• No programmes are active on the Engine: the Monitor pops up a message box alerting the user that there are no active programmes running on the Engine
Issues	None
Test case Scope/Type	GUI/Blackbox

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

11.1 AXMEDIS B2B Client Application

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).
Expected results	The professional User has successfully finished the B2B Client Registration; he has completely received all related Authorizations. The B2B Client Application can completely access the AXMEDIS Selection.
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

11.4 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 AXMEDIS Certifier and Supervisor and networks of AXCS (WP5.6.1: DSI)

12.1.1 AXMEDIS Registration of AXCSs (DSI)

TCId	TC12.1.1
Test case	Registration of an AXCS on AXCSs network
Initial conditions	None
Configuration description	An AXCS Manager wants to register his AXCS to AXCSs network. Data should be transferred from AXCS to other AXCSs on the network.
Description of functionality to be tested	Registration of an AXCS on AXCSs network
Partners, people involved	AXCS Managers
Validator(s) skill	None
Data set used	AXDS-AXCS1
Steps	1 Actor submits the test registration 2 The system returns results • The results are checked against the list of expected results
Expected results	AXCSs over the network register 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 AXCS 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 AXCS Manager register the received tool/device in the system 5 The results are checked against the list of expected results
Expected results	AXCS 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

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

12.1.3 AXMEDIS Object ID Generator

12.1.3.1 Generation of unique object ID (DSI)

TCId	TC12.1.3.1
Test case	Generation of unique object ID
Initial conditions	Temporary 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, Creator
Validator(s) skill	None
Data set used	AXDS-OIDGen1
Steps	1 Object ID is saved in the 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 record with information about the AXMEDIS Object
Variations	None
Issues	None
Test case Scope/Type	Backend / UnitTest

12.1.3.2 Registration of metadata about a new object

TCId	TC12.1.3.2
Test case	Registration of metadata about a new object
Initial conditions	Object ID created, AXMEDIS Editor opened and database connection available.
Configuration description	An actor wants to associate with the AXMEDIS Editor some metadata (such as Dublin Core metadata) to the newly created object.
Description of functionality to be tested	Metadata associated with the object are correctly saved in the database.
Partners, people involved	Integrator, Designer, Creator
Validator(s) skill	None
Data set used	AXDS-AXCS4
Steps	1. Object related metadata are saved into the database. The provided information must be compliant respect with related database field types, otherwise an error occours. 2. The hash of the objects is signed and returned
Expected results	A new database record 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 (DSI)

TCId	TC12.1.4.1
Test case	Search of AXMEDIS Objects (via web interface)
Initial conditions	AXCS database 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	AXCS database 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, AXCS 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 AXCS Collector

12.1.5.1 On-line transfer among AXCSs over AXCSs network (DSI)

TCId	TC12.1.5.1
Test case	On-line transfer among AXCSs over AXCSs network (DSI)
Initial conditions	AXCS database is filled with a predefined set of data (AXDS-AXCS10), network connection is available
Configuration description	Some information managed by AXCS has to be transferred to other AXCSs. This transfer involves AXCS Synchronizer and AXCS Collector
Description of functionality to be tested	Transfer between AXCS Synchronizer and AXCS Collector, AXCS Synchronizer queue capabilities, AXCS Collector capabilities
Partners, people involved	AXCS Managers
Validator(s) skill	Medium/High AXMEDS system knowledge
Data set used	AXDS-AXCS10 as initial condition and AXDS-AXCS11 to perform the test
Steps	1 A meaningful set of data stored in AXCS database is prepared to be transferred to AXCS Collector 2 AXCS Synchronizer transfer functionality is started with the correct parameters to transfer data to AXCS Collector
Expected results	The transfer occurs in the correct way, data is transferred completely with no error and inserted in 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 among AXCSs over AXCSs network (DSI)

TCId	TC12.1.5.2
Test case	Off-line transfer among AXCSs over AXCSs network (DSI)
Initial conditions	AXCS database is filled with a predefined set of data (AXDS-AXCS10), network connection is available

Configuration description	Some information collected by AXCS has to be transferred to other AXCS even if the connection among AXCSs is interrupted. In this case the transfer doesn't occur on-line, but off-line in a second time
Description of functionality to be tested	Transfer between AXCS Synchronizer and AXCS Collector, AXCS Synchronizer queue capabilities, AXCS Collector capabilities
Partners, people involved	AXCS Managers
Validator(s) skill	Medium/High AXMEDS system knowledge
Data set used	AXDS-AXCS10 as initial condition and AXDS-AXCS11 to perform the test
Steps	1 A meaningful set of data stored in AXCS database is prepared to be transferred to AXCS Collector 2 Step 1 repeated for a meaningful number of AXCSs 3 AXCS 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 destination 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 o The service has to return an error code o The pertinent AXCS Synchronizer queue is as the beginning of the test
Issues	None
Test case Scope/Type	Backend / WhiteBox

12.1.6 AXMEDIS Registration Service (DSI)

12.1.6.1 End User registration in a distribution channel (DSI)

TCId	TC12.1.6.1
Test case	Registration of an End User in a distribution channel
Initial conditions	AXCS database is filled with a predefined set of user registrations (AXDS-AXCS8)
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-AXCS8 as initial condition and 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 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

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

TCId	TC12.1.6.3
Test case	Registration of an End User in a different distribution channel
Initial conditions	AXCS database 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 functionality to be tested	Registration of an End user in a distribution channel different from the one the 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 ○ The service has to return an error code • 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

12.1.6.3 Registration of a new structured group of people (DSI)

TCId	TC12.1.6.4
Test case	Registration of a new Teacher/School or Student
Initial conditions	AXCS database 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	Registration of a new Teacher/School and of the pertinent students.

tested	
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

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

TCId	TC12.1.6.5
Test case	Registration of an Old User of the Channel on AXMEDIS
Initial conditions	AXCS database 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

12.1.6.5 User password modification

TCId	TC12.1.6.6

Test case	User password modification
Initial conditions	AXCS database 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 o The service has to return an error code • An actor tries to change a password to a new invalid password o The service has to return an error code
Issues	Password requirements
Test case Scope/Type	GUI / BlackBox

12.1.7 AXMEDIS Certification and Verification (FUPF)

12.1.7.1 Certification of AXMEDIS Tool by a User on a device

TCId	TC12.1.7.1a
Test case	Certification of AXMEDIS Tool by a User on a device
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	Certification of an AXMEDIS tool by an AXMEDIS User on a device
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 and 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 needed information for the certification: AXUID, AXRTID, tool fingerprint, registration deadline. 4 The mentioned PMS contacts the pertinent AXMEDIS Certification and Verification sending him all the received information plus AXDOM 5 AXCV checks that the tool was not previously certified.

	6 AXCV checks that the user and tool are registered (check that the user data and status are correct, that the received Tool FP matches the original one and that the registration deadline does not exceed the maximum tool deadline). 7 AXMEDIS Certification and Verification generates a TID (tool ID), an enabling code and a tool private key and x509 certificate and inserts it together with all the received information in the AXCS Database (except the private key), using the proper interface. At the end, the tool has been certified in AXMEDIS by the corresponding user. 8 AXMEDIS Certification and Verification sends to PMS the generated TID, enabling code and a PKCS12 structure that includes the tool private key and certificate. 9 The PMS sends to AXOM (as a part of the tool) a confirmation message, including previous information. 10 AXOM stores the received TID, imports the certificate and private key and enables the tool.
Expected results	The tool is certified in the AXMEDIS system A new tool ID, enabling code and certificate are generated and bounded to the tool The requester receives notification about the certification
Variations	None
Issues	None
Test case Scope/Type	Backend

TCId	TC12.1.7.1b
Test case	Certification of an AXMEDIS tool by an AXMEDIS User on a device
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 an AXMEDIS tool performed by an AXMEDIS User on a device
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 and 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 certification: AXUID, AXRTID, tool fingerprint, registration deadline. 4 The mentioned PMS contacts the pertinent AXMEDIS Certification and Verification sending him all the received information plus AXDOM 5 AXCV determines that the tool was previously certified, or that the user and tool are not registered, or that the user data and status are not correct 6 The PMS and AXOM are sent a message notifying the unsuccessful certification
Expected results	• The tool is not certified • The PMS is sent a message notifying the unsuccessful certification
Variations	• If the tool is registered (the received Tool FP matches the original FP) and the user is not, the user might be asked to register

	• If the tool was previously certified, a recovery mechanism should be provided
Issues	None
Test case Scope/Type	Backend

TCId	TC12.1.7.1c
Test case	Certification of an AXMEDIS tool by an AXMEDIS User on a device
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 an AXMEDIS tool performed by an AXMEDIS User on a device
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 and 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 certification: AXUID, AXRTID, tool fingerprint, registration deadline. 4 The mentioned PMS contacts the pertinent AXMEDIS Certification and Verification sending him all the received information plus AXDOM 5 AXCV determines that the tool was not previously certified, that the user and tool are registered, and that the user data and status are correct 6 AXCV determines one of the following: 6.1 the received Tool FP does not match the original one 6.2 that the registration deadline exceeds the maximum tool deadline. 7 The PMS and AXOM are sent a message notifying the unsuccessful certification
Expected results	• The tool is not certified • If the Tool FP did not match (case 6.1), the registered tool is quarantined in order to check if it has to be blocked • In case 6.2 the user is blocked and a SupervisorInputData is stored in AXCS accounting database to explain the reason why.
Variations	The registered tool is blocked instead of quarantined
Issues	None
Test case Scope/Type	Backend

12.1.7.2 Verification of AXMEDIS users using AXMEDIS tools on a Device before content consumption

TCId	TC12.1.7.2a
Test case	Verification of AXMEDIS users using AXMEDIS tools on a Device before content consumption
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 hash/fingerprint consistency
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: AXUID, AXTID and the tool fingerprint digest (hardware and software). 2 PMS of reference contacts the Domain Manager to retrieve the domain the user is working in (AXDOM). 3 PMS of reference contacts AXCS-AXCV sending the received information plus the AXDOM (domain) and the list of operations performed during offline operation (list of ActionLogs) if any 4 AXCV verifies the received user and tool information. 5 Certification and verification (AXCV, inside AXCS) checks that the tool is certified. 6 AXCV contacts AXMEDIS Supervisor (AXS) to verify and store the list of offline operations. AXS recalculates for each received Action Log the tool history fingerprint and compares the last one against the one that is stored in the AXCS database (using AXCS Database interface). 7 The PMS is notified that the user data, tool data and operation history have been correctly verified.
Expected results	The AXMEDIS Certification and Verification returns a notification message targeted to the PMS where it notifies that the user data, tool data and operation history have been correctly verified.
Variations	None
Issues	None
Test case Scope/Type	Blackbox

TCId	TC12.1.7.2b
Test case	Verification of AXMEDIS users using AXMEDIS tools on a Device before content consumption
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 user data, tool data and tool operation history hash/fingerprint consistency due to tool fingerprint digest mismatch
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: AXUID, AXTID and the tool fingerprint digest (hardware and software). 2 PMS of reference contacts the Domain Manager to retrieve the domain the user is working in (AXDOM). 3 PMS of reference contacts AXCS-AXCV sending the received information plus the AXDOM (domain) and the list of operations performed during offline

	operation (list of ActionLogs) if any 4 AXCV verifies the received user and tool information and determines that the received tool fingerprint digest does not match the one stored in the database 5 The PMS is notified the tool fingerprint digest mismatch.
Expected results	A reverification should be the next step performed by AXOM after this test case. See test case "Reverification of AXMEDIS users using AXMEDIS tools on a Device during content consumption inside a domain"
Variations	None
Issues	None
Test case Scope/Type	Blackbox

TCId	TC12.1.7.2c
Test case	Verification of AXMEDIS users using AXMEDIS tools on a Device before content consumption
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 user data, tool data and tool operation history hash/fingerprint consistency
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: AXUID, AXTID and the tool fingerprint digest (hardware and software). 2 PMS of reference contacts the Domain Manager to retrieve the domain the user is working in (AXDOM). 3 PMS of reference contacts AXCS-AXCV sending the received information plus the AXDOM (domain) and the list of operations performed during offline operation (list of ActionLogs) if any 4 AXCV verifies the received user and tool information and determines one of the following: 4.1 the tool is not certified, its status is not correct or the deadline has expired 4.2 the received operation list is incorrect 4.3 the tool should have been already blocked 4.4 the operation history verification is not correct 5 The PMS is notified the error
Expected results	AXMEDIS Certification and Verification returns a notification message targeted to the PMS where it notifies the problem that occurred Additional expected results for the different cases are the following: 4.1 none 4.2 the tool is blocked in AXCS 4.3 user and tool are blocked in AXCS 4.4 the tool is blocked in AXCS
Variations	None
Issues	None
Test case Scope/Type	Blackbox

12.1.7.3 Reverification of AXMEDIS users using AXMEDIS tools on a Device during content consumption inside a domain

TCId	TC12.1.7.3a
Test case	Reverification of AXMEDIS users using AXMEDIS tools on a Device during content consumption inside a domain
Initial conditions	The verification has failed due to tool fingerprint digest mismatch
Configuration description	None
Description of functionality to be tested	Positive reverification of user data, tool data and tool operation history hash/fingerprint consistency
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: AXUID, AXTID and whole tool fingerprint (hardware and software). 2 PMS of reference contacts the AXCS sending him the received information plus the list of operations performed during offline operation (list of ActionLogs) 3 AXCV verifies the received user and tool information. 4 Certification and verification (AXCV, inside AXCS) checks that the tool is certified. 5 AXCV contacts AXMEDIS Supervisor (AXS) to verify and store the list of offline operations. AXS recalculates for each received Action Log the tool history fingerprint and compares the last one against the one that is stored in the AXCS database (using AXCS Database interface). 6 The PMS is notified that the user data, tool data and operation history have been correctly reverified.
Expected results	The AXMEDIS Certification and Verification returns a notification message targeted to the PMS where it notifies that the user data, tool data and operation history have been correctly verified.
Variations	None
Issues	None
Test case Scope/Type	Blackbox

TCId	TC12.1.7.3b
Test case	Reverification of AXMEDIS users using AXMEDIS tools on a Device during content consumption inside a domain
Initial conditions	The verification has failed due to tool fingerprint digest mismatch
Configuration description	None
Description of functionality to be tested	Negative reverification of user data, tool data and tool operation history hash/fingerprint consistency due to tool fingerprint mismatch
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: AXUID, AXTID and the whole tool fingerprint (hardware and software) 2 PMS of reference contacts the Domain Manager to retrieve the domain the

	user is working in (AXDOM) 3 PMS of reference contacts AXCS-AXCV sending the received information plus the AXDOM (domain) and the list of operations performed during offline operation (list of ActionLogs) if any 4 AXCV verifies the received user and tool information and determines that the received tool fingerprint does not match the one stored in the database 5 AXCV blocks the certified tool and user and stores a SupervisorInputData in AXCS accounting database to explain the reason why 6 The PMS is notified the tool fingerprint mismatch
Expected results	AXOM should immediately deactivate it when it receives the "unsuccessful reverification" notification message.
Variations	None
Issues	None
Test case Scope/Type	Blackbox

TCId	TC12.1.7.3c
Test case	Reverification of AXMEDIS users using AXMEDIS tools on a Device during content consumption inside a domain
Initial conditions	The verification has failed due to tool fingerprint digest mismatch
Configuration description	None
Description of functionality to be tested	Negative reverification of user data, tool data and tool operation history hash/fingerprint consistency
Partners, people involved	None
Validator(s) skill	High, technical
Data set used	AXDS-CertVer1
Steps	1 AXMEDIS Tool (AXOM) sends some needful information to PMS, such as: AXUID, AXTID and the whole tool fingerprint (hardware and software) 2 PMS of reference contacts the Domain Manager to retrieve the domain the user is working in (AXDOM) 3 PMS of reference contacts AXCS-AXCV sending the received information plus the AXDOM (domain) and the list of operations performed during offline operation (list of ActionLogs) if any 4 AXCV verifies the received user and tool information and determines one of the following: 4.1 the tool is not certified, its status is not correct or the deadline has expired 4.2 the received operation list is incorrect 4.3 the tool should have been already blocked 4.4 the operation history verification is not correct The PMS is notified the error 5 AXCV blocks the certified tool and user and stores a SupervisorInputData in AXCS accounting database to explain the reason why 6 The PMS is notified the tool fingerprint mismatch
Expected results	AXMEDIS Certification and Verification returns a notification message targeted to the PMS where it notifies the problem that occurred AXOM should immediately deactivate it when it receives the "unsuccessful reverification" notification message. Additional expected results in AXCS for the different cases are the following: • none • the tool is blocked in AXCS

	• user and tool are blocked in AXCS • the tool is blocked in AXCS
Variations	None
Issues	None
Test case Scope/Type	Blackbox

12.1.7.4 Storage of PMS Action Log in AXCS Accounting Database

TCId	TC12.1.7.4a
Test case	Storage of PMS Action Log in AXCS Accounting database
Initial conditions	PMS Server has successfully authorised a user to perform an action over a resource
Configuration description	
Description of functionality to be tested	Successful verification and storage of Action Log through AXMEDIS Certification and Verification by PMS Server after authorising
Partners, people involved	PMS Server
Validator(s) skill	Advanced user
Data set used	AXDS-CertVer1
Steps	1. PMS Server fills the required data of the Action Log after authorising a user 2. PMS Server sends AXCS-AXCV the generated single Action Log to be verified and stored 3. AXCS-AXCV correctly verifies the information in the received Action Log (certified tool status, tool deadline and tool fingerprint) 4. AXCS-AXCV contacts AXCS-AXS to store the ActionLog 5. AXCS-AXS checks that all required ActionLog fields are present, and determines that tool operation history was not previously set to “invalid” 6. AXCS-AXS sets the registration timestamp to the appropriate value, recalculates and stores in AXCS RegCert database the operation history hash and stores the ActionLog in AXCS accounting database.
Expected results	1. The ActionLog sent by PMS is correctly stored in AXCS accounting database 2. The successful result is notified to PMS Server 3. The user is authorised to perform the requested action
Variations	None
Issues	AXCS-AXCV does NOT set the HistVerSuccess field, as it cannot verify the operation history in this case
Test case Scope/Type	GUI / BlackBox

TCId	TC12.1.7.4b
Test case	Storage of PMS Action Log in AXCS Accounting database
Initial conditions	PMS Server has successfully authorised a user to perform an action over a resource
Configuration description	
Description of functionality to be tested	Unsuccessful verification and storage of Action Logs through AXMEDIS Certification and Verification by PMS Server after authorising
Partners, people involved	PMS Server
Validator(s) skill	Advanced user

Data set used	AXDS-CertVer1
Steps	1. PMS Server fills the required data of the Action Log after authorising a user 2. PMS Server sends AXCS-AXCV the generated single Action Log to be verified and stored 3. AXCS-AXCV does not correctly verify the information in the received Action Log due to: 3.1 incorrect user status 3.2 incorrect certified tool status 3.3 expired tool deadline 3.4 tool fingerprint mismatch 4. AXCS-AXCV sends PMS Server a message notifying the unsuccessful verification and storage of the Action Log
Expected results	• PMS Server does not authorise the user to perform the action • In case 3.2 and 3.3 the tool must be deactivated by AXOM • In case 3.4 the user and certified tool are be blocked by AXCS-AXCV
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC12.1.7.4c
Test case	Storage of PMS Action Log in AXCS Accounting database
Initial conditions	PMS Server has successfully authorised a user to perform an action over a resource
Configuration description	
Description of functionality to be tested	Unsuccessful verification and storage of Action Logs through AXMEDIS Certification and Verification by PMS Server after authorising
Partners, people involved	PMS Server
Validator(s) skill	Advanced user
Data set used	AXDS-CertVer1
Steps	1 PMS Server fills the required data of the Action Log after authorising a user 2 PMS Server sends AXCS-AXCV the generated single Action Log to be verified and stored 3 AXCS-AXCV correctly verifies the information in the received Action Log (certified tool status, tool deadline and tool fingerprint) 4 AXCS-AXCV contacts AXCS-AXS to store the ActionLog 5 AXCS-AXS checks that all required ActionLog fields are present, and determines that tool operation history was previously set to “invalid” (i.e. the tool should have been already blocked) 6 AXCS-AXCV blocks the certified tool and the user 7 AXCS-AXCV sends PMS Server a message notifying that the tool should have been already blocked
Expected results	• PMS Server does not authorise the user to perform the action • The user and certified tool are blocked by AXCS-AXCV
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

12.1.8 Manual blocking

12.1.8.1 Manual user blocking / unblocking (DSI)

TCId	TC12.1.8.1
Test case	User blocking/unblocking
Initial conditions	AXCS database 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	Manual blocking/unblocking of a user
Partners, people involved	Publishers, Distributors, Final Users, AXCS manager
Validator(s) skill	High, Technical
Data set used	AXDS-Supervisor1
Steps	1 The block/unblock requester sends a block/unblock request to an authority entitled to decide for (un)blocking 2 The authority checks and validates the received request 3 If the check is passed, the authority sends the request to the proper AXCS administrator 4 The AXCS administrator blocks/unblocks the user using the AXCS Management Console 5 The AXCS administrator sends to the requester the confirmation of the (un)blocking of the user
Expected results	• The user is marked as (un)blocked in the AXCS Registration and Certification Database (its own status results changed) • The requester receives notification about the (un)blocking
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	Backend / BlackBox

12.1.8.2 Certified Tool blocking/ unblocking

The "registration" term refers to Tool Off-line Registration scenario. A registered tool is a software product. An instance of a Registered Tool running on a terminal becomes a Certified Tool.

Blocking a tool can have different "rules":

- Blocking a specific certified tool belonging to one user (e.g. due to manipulations).
- Blocking a specific version of the tool (named registered tool) in a mandatory manner (e.g. that specific version has a security hole and an exploit has been released over the internet). It is a way to force downloading a new version to preserve system integrity.

TCId	TC12.1.8.2
Test case	Certified tool blocking/unblocking
Initial conditions	AXCS database system is filled with a predefined set of registered and certified tools
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be	Manual blocking/unblocking of a certified tool

tested	
Partners, people involved	Publishers, Distributors, Final Users, AXCS manager
Validator(s) skill	High, Technical
Data set used	AXDS-Supervisor2
Steps	1 The block/unblock requester sends a block/unblock request to an authority entitled to decide for (un)blocking 2 The authority checks and validates the received request 3 If the check is passed, the authority sends the request to the proper AXCS administrator 4 The AXCS administrator blocks/unblocks the certified tool using the AXCS Management Console 5 The AXCS administrator sends to the requester the confirmation of the (un)blocking of the certified tool
Expected results	• The certified tool is marked as (un)blocked in the AXCS Registration and Certification Database (its own status results changed) • The specific instance of a tool (named certified tool) results (un)blocked for all users. Other tool instances belonging to the same tool family (named registered tool) are not affected by the status change • The requester receives notification about the (un)blocking
Variations	• An actor tries to block a tool not certified 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

12.1.8.3 Registered Tool blocking/ unblocking

TCId	TC12.1.8.3
Test case	Registered tool blocking/unblocking
Initial conditions	AXCS database is filled with a predefined set of registered tools
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	Manual blocking/unblocking of a registered tool
Partners, people involved	Publishers, Distributors, Final Users, AXCS manager
Validator(s) skill	High, Technical
Data set used	AXDS-Supervisor3
Steps	1 The block/unblock requester sends a block/unblock request to an authority entitled to decide for (un)blocking 2 The authority checks and validates the received request 3 If the check is passed, the authority sends the request to the proper AXCS administrator 4 The AXCS administrator blocks/unblocks the registered tool using the AXCS Management Console 5 The AXCS administrator sends to the requester the confirmation of the (un)blocking of the registered tool
Expected results	• The registered tool is marked as (un)blocked in the AXCS Registration and Certification Database • All the certified tool instances belonging to that tool family (named registered

	tool) result (un)blocked for all users • The requester receives notification about the (un)blocking
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

12.1.9 AXMEDIS Supervisor (FUPF)

12.1.9.1 AXMEDIS Protection information delivery

TCId	TC12.1.9.1a
Test case	AXMEDIS Protection information delivery
Initial conditions	An AXMEDIS User has a protected AXMEDIS object and needs the corresponding key to unprotect and consume it. PMS Client has already asked PMS Server for authorisation t, who has successfully authorised the user
Configuration description	None
Description of functionality to be tested	PMS Client successfully receives information for unprotecting an AXMEDIS object
Partners, people involved	PMS Client
Validator(s) skill	Advanced user
Data set used	AXDS-PMS1
Steps	1 PMS Server requests AXCS-AXS the protection information associated to an AXMEDIS object 2 AXMEDIS Supervisor verifies the request validity 3 AXS retrieves the information through AXCS database Interface and returns it to PMS.
Expected results	PMS Client stores the protection information in the secure local cache, if possible, for content fruition
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC12.1.9.1b
Test case	AXMEDIS Protection information delivery
Initial conditions	An AXMEDIS User has a protected AXMEDIS object and needs the corresponding key to unprotect and consume it. PMS Client has already asked PMS Server for authorisation t, who has successfully authorised the user
Configuration description	None
Description of functionality to be tested	PMS Client does not receive information for unprotecting an AXMEDIS object
Partners, people involved	PMS Client
Validator(s) skill	Advanced user
Data set used	AXDS-PMS1

Steps	1 PMS Server requests AXCS-AXS the protection information associated to an AXMEDIS object 2 AXMEDIS Supervisor verifies the request validity 3 AXMEDIS Supervisor determines that there is not any protection information for the received parameters
Expected results	AXMEDIS Supervisor sends PMS Server, who sends PMS Client the negative result
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

12.1.9.2 Storage/update of protection information of an AXMEDIS object to the AXCS

TCId	TC12.1.9.2a
Test case	Storage/update of protection information of an AXMEDIS Object in AXCS ObjectsId Database
Initial conditions	An AXMEDIS User has protected an AXMEDIS object and generated the related protection information
Configuration description	None
Description of functionality to be tested	The protection information is correctly stored/updated in AXCS ObjectsId Database
Partners, people involved	Creators, Distributors, Tool producers
Validator(s) skill	Advanced user
Data set used	AXDS-PMS1
Steps	1 AXMEDIS Supervisor receives a store/update request containing the AXOID of the protected object, the version, the protection timestamp and the related protection information 2 AXMEDIS Supervisor verifies the request validity 3 The protection information is stored/updated in AXCS database and linked to the given AXOID.
Expected results	4. The protection information is stored/updated in AXCS database and linked to the given AXOID 5. The successful result is notified to the user
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC12.1.9.2b
Test case	Storage/update of protection information of an AXMEDIS Object in AXCS ObjectsId Database
Initial conditions	An AXMEDIS User has protected an AXMEDIS object and generated the related protection information
Configuration description	None
Description of functionality to be tested	The protection information is not correctly stored/updated in AXCS ObjectsId Database
Partners, people involved	Creators, Distributors, Tool producers
Validator(s) skill	Advanced user

Data set used	AXDS-PMS1
Steps	1 AXMEDIS Supervisor receives a store/update request containing the AXOID of the protected object, the version, the protection timestamp and the related protection information 2 AXMEDIS Supervisor verifies the request validity 3 AXMEDIS Supervisor determines that there is not any entry in AXCS Objects database that matches the input information
Expected results	6. The protection information is not stored/updated in AXCS database nor linked to the given AXOID 7. The unsuccessful result is notified to the user
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

12.1.9.3 Storage of SupervisorInputData in the AXCS Accounting Database

TCId	TC12.1.9.3a
Test case	Storage of SupervisorInputData in AXCS Accounting database
Initial conditions	Someone has requested a license to PMS Server or the result of an authorisation in PMS Server is negative. A SupervisorInputData is generated.
Configuration description	None
Description of functionality to be tested	Successful storage of a SupervisorInputData coming from PMS Server
Partners, people involved	PMS, AXCS-AXS
Validator(s) skill	Advanced user
Data set used	AXDS-Supervisor4
Steps	1. AXCS-AXS receives a SupervisorInputData coming from PMS Server 2. AXCS-AXS verifies that the SupervisorInputData has the necessary information and stores it in AXCS Accounting database.
Expected results	• The SupervisorInputData is susccessfully stored in AXCS Accounting database • The successful storage is notified to the PMS Server
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

TCId	TC12.1.9.3b
Test case	Storage of SupervisorInputData in AXCS Accounting database
Initial conditions	Someone has requested a license to PMS Server or the result of an authorisation in PMS Server is negative. A SupervisorInputData is generated.
Configuration description	None
Description of functionality to be tested	Unsuccessful storage of a SupervisorInputData coming from PMS Server
Partners, people involved	PMS, AXCS-AXS
Validator(s) skill	Advanced user
Data set used	AXDS-Supervisor4

Steps	1 AXCS-AXS receives a SupervisorInputData coming from PMS Server 2 AXCS-AXS determines that the SupervisorInputData does not have all the necessary information to be stored
Expected results	• The SupervisorInputData is not stored in AXCS Accounting database • The error is notified to PMS Server
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

12.1.10 AXMEDIS Reporting Service and Statistics Web Service (EXITECH)

12.1.10.1 Object usage reporting for accounting purposes

TCId	TC12.1.10.1
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 involved	Distributors, Providers
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

12.1.10.2 Object usage reporting for statistic purposes

TCId	TC12.1.10.2
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 involved	Distributors, Providers
Validator(s) skill	None
Data set used	None
Steps	1 Actor submits the object usage reporting request with the request for high level statistics 2 The system returns results

	3 The results are checked against the list of expected results
Expected results	AXCS correctly elaborates reports and the high level statistics are correctly provided
Variations	• Empty report • Report with only one item • Report with a very huge number of items
Issues	None
Test case Scope/Type	BlackBox

12.1.11 Accounting Manager and Reporting Tool (EXITECH)

12.1.11.1 List of all operations performed on an object

TCId	TC12.1.11.1
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

12.1.11.2 List of all operations performed by a user

TCId	TC12.1.11.2
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 involved	Distributors
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

12.1.11.3 Usage report about an object

TCId	TC12.1.11.3
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

12.1.11.4 Usage report about a distributor

See Variation 1 of test case 12.1.11.3.

12.1.11.5 Usage report about a provider

See Variation 2 of test case 12.1.11.3.

12.1.12 AXCS Synchroniser (DSI)

TCId	TC12.1.12
Test case	AXCS Synchroniser
Initial conditions	AXCS on one channel is filled with predefined data
Configuration description	AXMEDIS Registration and Certification database, AXMEDIS Accounting database as needed
Description of functionality to be tested	AXCS Synchroniser is capable of giving data to other AXCSs via AXCS Collector
Partners, people involved	Actors that interact with AXCS, AXCS
Validator(s) skill	None

Data set used	AXDS-AXCS10
Steps	1 AXCS send to other AXCSs a request for data 2 AXCS Synchroniser send the data collected in the AXCS 1 Data returned are checked against those filled in the Initial Conditions
Expected results	AXCS correctly returns Data
Variations	None
Issues	None
Test case Scope/Type	BlackBox

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

Protection tool engine test cases regarding DRM support are defined in section 12.4.2, DRM support.

12.2.1 Content protection

TCId	TC12.2.1
Test case	Content protection
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, edit, debug, print, and activitate protection rules • User can deactivate and delete protection rules • User can create licences automatically and assign them to objects • User can automatically verify and edit licenses and PARs
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	See details below
Expected results	
Variations	
Issues	None
Test case Scope/Type	GUI/Whitebox

12.2.2 Create a new protection rule

TCId	TC12.2.2
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 new 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

12.2.3 Search for and select a protection rule

TCId	TC12.2.3
Test case	Search for and select of a 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 load 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 search for a protection rule 2 The User selects a protection rule for opening it in the GUI
Expected results	Rule is opened in the editor and can be processed or executed
Variations	• Rules can be active/inactive
Issues	None
Test case Scope/Type	GUI/Whitebox

12.2.4 Activating a protection rule

TCId	TC12.2.4
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	• Rule already selected and loaded • User can modify/cancel this action before the activation
Issues	None
Test case Scope/Type	GUI/Whitebox

12.2.5 Removing a protection rule

TCId	TC12.2.5
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

12.2.6 Debugging a protection rule

TCId	TC12.2.6
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

12.2.7 Editing protection rules

TCId	TC12.2.7
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

12.2.8 Printing protection rules

TCId	TC12.2.8
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

12.2.9 Automatic creation and association of licenses

TCId	TC12.2.9
Test case	Automatic creation and association of licenses
Initial conditions	The AXMEDIS Protection Rules Editor is running
Configuration description	Protection Rules Editor is running
Description of functionality to be tested	• User creates licenses and associates them with existing content
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 creates a script for the creation of one or more licences 2 The User creates a rule that specifies the AXMEDIS objects that are to be associate with this/these licenses 3 The User writes or updates the schedule associated with the rule 4 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

12.2.10 Automatic verification of licenses or PARs

TCId	TC12.2.10
Test case	Automatic verification of licences or PARs
Initial conditions	The AXMEDIS Protection Rules Editor is running
Configuration description	Protection Rules Editor is running
Description of functionality to be tested	• User verifies licenses or PARs
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 creates a script for the automatic verification of licences or PARs 2 The User creates a rule that specifies the AXMEDIS objects that are to verified 3 The User writes or updates the schedule associated with the rule 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

12.2.11 Automatic editing of PARs

TCId	TC12.2.11
Test case	Automatic editing of PARs
Initial conditions	The AXMEDIS Protection Rules Editor is running
Configuration description	Protection Rules Editor is running
Description of functionality to be tested	• User can automatic edit PARs
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 creates a script for the editing PARs 2 The User creates a rule that specifies the AXMEDIS objects that are to be associate with this/these licenses 3 The User writes or updates the schedule associated with the rule 4 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

12.2.12 Automatic editing of licenses

TCId	TC12.2.12
Test case	Automatic editing of licenses
Initial conditions	The AXMEDIS Protection Rules Editor is running
Configuration description	Protection Rules Editor is running
Description of functionality to be tested	• User can automatic edit licenses
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 creates a script for the editing of licenses 2 The User creates a rule that specifies the AXMEDIS objects that are to be associate with this/these licenses

	3 The User writes or updates the schedule associated with the rule 4 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 Administrative Information Integrator (WP9.1: EXITECH)

12.3.1 Integrating Distributor administrative information on the basis of end user actions

TCId	TC12.3.1
Test case	Integrating Distributor administrative information on the basis of end user actions
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	Low for manual check of results
Data set used	AXDS-DB3
Steps	1 Distributor submits a manual request for administrative information to be displayed 2 The system returns results in an XML format 3 The XML format is mapped to the that chosen for the CMS 4 The result provided are checked against those expected
Expected results	The result set of statistic has
Variations	• Actor is the collecting society
Issues	None
Test case Scope/Type	BlackBox

12.3.2 Integrating Collecting Society administrative information on the basis of end user actions

Variation of TC 12.3.1.

12.3.3 Automatic Administrative information retrieval for distributors

TCId	TC12.3.3
Test case	Automatic Administrative information retrieval for distributors
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 a request 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

12.3.4 Automatic Administrative information retrieval for collecting societies

This is a variation of the TC 12.3.3.

12.4 Protection Manager Support / Server General

12.4.1 Protection Manager Support / Server

12.4.1.1 Consumption of a protected and governed AXMEDIS object in a connected environment

TCId	TC12.4.1.1a
Test case	Consumption 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. • Consumption of the object requires protection information
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 ProtectionProcessor through PMS requests the authorisation for the needed action. It sends object identifier, user identifier, tool identifier, operation identifier along with all other information needed to retrive license appropriated to grant the requested action 2 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 3 As the end-user has the appropriate license, the authorisation is positive. 4 PMS checks with AXCS if end-user has got the keys for decrypting object. 5 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 (protection information) needed to unprotect the object from the AXCS. This information is delivered to the PMSClient over a secure channel. 6 The object is consumed at this time or at a delayed time. Protection Information remains in PMSClient Secure Cache until the object is consumed by the user.
Expected results	The end-user is allowed to consume the protected AXMEDIS object
Variations	None
Issues	None

TCId	TC12.4.1.1b
Test case	Consumption 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 but has not 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 ProtectionProcessor through PMS requests the authorisation for the needed action. It sends object identifier, user identifier, tool identifier, operation identifier along with all other information needed to retrive license appropriated to grant the requested action 2 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 cannot find an appropriate license. 3 PMS report ProtectionProcessor authorization failure
Expected results	The end-user cannot consume the protected AXMEDIS object
Variations	None
Issues	None

TCId	TC12.4.1.1c
Test case	Consumption 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. • Consumption of the object does not require protection information
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 ProtectionProcessor through PMS requests the authorisation for the needed action. It sends object identifier, user identifier, tool identifier, operation identifier along with all other information needed to retrive license appropriated to grant the requested action 2 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 3 As the end-user has the appropriate license, the authorisation is positive. 4 The object is consumed at this time or at a delayed time
Expected results	The end-user is allowed to consume the protected AXMEDIS object
Variations	None
Issues	None

12.4.1.2 Consumption of a protected and governed AXMEDIS object in an unconnected environment

TCId	TC12.4.1.2a
Test case	Consumption of a protected and governed AXMEDIS object in an unconnected 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 in the local PMS cache. • Authorization support is running. • Consumption of the object requires protection information
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 ProtectionProcessor through PMS requests the authorisation for the needed action. It sends object identifier, user identifier, tool identifier, operation identifier along with all other information needed to retrive license appropriated to grant the requested action 2 PMS requests the authorisation to the authorisation support. 3 Action is granted by a license present in the local cache 4 Protection Information is not available thus in unconnected environment PMSClient returns a failure in authorisation
Expected results	The end-user cannot consume the protected AXMEDIS object
Variations	None
Issues	None

TCId	TC12.4.1.2b
Test case	Consumption of a protected and governed AXMEDIS object in an unconnected 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, but not in the local PMS cache. • Authorization support is running.
Configuration description	None
Description of functionality to be	An end-user wants to consume a protected and governed AXMEDIS object for first time. End-user is not entitled of the appropriate licenses to consume the

tested	object
Partners, people involved	End-user
Validator(s) skill	None
Data set used	AXDS-PMS1
Steps	1 ProtectionProcessor through PMS requests the authorisation for the needed action. It sends object identifier, user identifier, tool identifier, operation identifier along with all other information needed to retrive license appropriated to grant the requested action 2 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 3 Authorization fails
Expected results	The end-user cannot consume the protected AXMEDIS object
Variations	None
Issues	None

TCId	TC12.4.1.2c
Test case	Consumption of a protected and governed AXMEDIS object in an unconnected 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 in the local cache. • Authorization support and AXCS are running. • Consumption of the object does not require protection information
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 ProtectionProcessor through PMS requests the authorisation for the needed action. It sends object identifier, user identifier, tool identifier, operation identifier along with all other information needed to retrive license appropriated to grant the requested action 2 PMS requests the authorisation to the authorisation support. 3 Action is granted by a license present in the local cache 4 PMSClient authorise object consumption
Expected results	The end-user consume the protected AXMEDIS object
Variations	None
Issues	None

12.4.1.3 Protection of an AXMEDIS object (DSI)

TCId	TC14.4.1.3
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.
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 The user opens the Protection Editor and Viewer on the desired object 3 The user sets the wanted protection algorithms and related parameters 4 The user save the object and the AXOM, by means of Protection Processor, save the object protecting it.
Expected results	AXMEDIS object has been protected and the protection information has been stored within it.
Variations	None
Issues	None

12.4.1.4 Registration of a protected object (DSI)

TCId	TC12.4.1.4
Test case	Registration of a protected object
Initial conditions	AXMEDIS editor is opened with a protected AXMEDIS object loaded The user has logged in and his identity has been validated.
Configuration description	None
Description of functionality to be tested	A user wants to register on AXCS a protected 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 register a protected AXMEDIS object. 2 The user clicks on the “Register object” voice in the menu 3 The AXOM save the object on a file cutting off the protection information contained in it. The cut protection information is sent to the AXCS together with the other registration data.
Expected results	AXMEDIS object has been protected and the protection information has been stored on the AXCS.
Variations	None
Issues	None

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

TCId	TC12.4.1.5
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 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 and registered another time (see TC "Registration of a protected object").
Expected results	AXMEDIS object has been properly stored together with its new protection information
Variations	None
Issues	None

12.4.2 DRM Support (WP4.5.1: FUPF)

12.4.2.1 License creation for new content

TCId	TC12.4.2.1a
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 PMS Server through the PMS Client 2 User enters the required data to create the license 3 User clicks the "Submit" button of the license creation tool 4 License generator (inside PMS Server) creates the license based on the received information 5 License 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 (by means of PMS Client) 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	TC12.4.2.1b
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 PMS Server through the PMS Client 2 User enters the required data to create the license 3 User clicks the "Submit" button of the license creation tool 4 License generator (inside PMS Server) creates the license based on the received information 5 License verifier 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

12.4.2.2 License creation for cross-media content

TCId	TC12.4.2.2a
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	TC12.4.2.2b
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

12.4.2.3 License verification against parent licenses

TCId	TC12.4.2.3
Test case	License verification against parent licenses
Initial conditions	At least, one user license should exist, and a License Distributor license.
Configuration description	None
Description of functionality to be tested	An actor requests a license to verify if it is really created by the issuer with an acceptable LD.
Partners, people involved	License creator
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 A user creates a new license. 2 User asks for verification of the newly created license. The license is checked against the chain of parent licenses (at least the direct parent). To do so, the license is checked using a simplified version of the MPEG-21 authorisation algorithm. First, we look for the direct parent license and check that conditions are exactly the same. If not, we iterate until there are no more parent licenses for that license or the condition is satisfied. 3 If the license is verified, then it is stored. 4 If not, the license is not verified and the user is informed.
Expected results	A Boolean value is returned. It indicates if license is valid and created from

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

12.4.2.4 License verification against PAR

TCId	TC12.4.2.4
Test case	License verification against Possible Available Rights
Initial conditions	At least, one user license should exist, a License Distributor license and a special License containing possible available rights for the distributor.
Configuration description	None
Description of functionality to be tested	An actor requests a license to verify if it is really created by the issuer with an acceptable LD and permits one of the Possible Available Rights, specified in the database for this issuer.
Partners, people involved	License creator
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 A user creates a new license. 2 User asks for verification of the newly created license. The license is checked against object PAR. To do so, the license is checked using a simplified version of the MPEG-21 authorisation algorithm. 3 If the license is verified, then it is stored. 4 If not, the license is not verified and the user is informed.
Expected results	A Boolean value is returned. It indicates if license is valid and created according to PAR.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

12.4.2.5 User authorisation on unconnected environment

TCId	TC12.4.2.5a
Test case	User authorisation on unconnected environment. The object is unprotected, so can be used without accessing keys.
Initial conditions	At least one license in the secure cache authorises the user to perform the action is trying to do.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is offline and the user wants to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s).

	3 Locally the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 4 The authorisation support performs the authorisation checking if conditions are satisfied with one of the obtained licenses. 5 The result of the authorisation is positive the user is authorised. 6 A complete ActionLog is stored in the secure cache.
Expected results	A bitwise containing zero.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.5b
Test case	User authorisation on unconnected environment. The object is unprotected, so can be used without accessing keys.
Initial conditions	There are no licenses in the secure cache that authorises the user to perform the action is trying to do.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is offline and the user wants to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 Locally the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 4 The authorisation support performs the authorisation checking if conditions are satisfied with one of the obtained licenses. 5 The result of the authorisation is negative and the user is not authorised. 6 A complete ActionLog is stored in the secure cache.
Expected results	A bitwise containing the corresponding error number if authorisation is rejected.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.5c
Test case	User authorisation on unconnected environment. The object is protected, so can not be used without accessing keys.
Initial conditions	At least one license in the secure cache authorises the user to perform the action is trying to do. Exists the protection info of the object in the Secure Cache
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people	Distributors, end-users.

involved	
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is offline and the user wants to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The object is protected, the PMS Client requests to the SecureCache the Protection Information and is present. 4 Locally the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 5 The authorisation support performs the authorisation checking if conditions are satisfied with one of the obtained licenses. 6 The authorisation is positive the user is authorised. 7 A complete ActionLog is stored in the secure cache.
Expected results	A bitwise containing zero.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.5d
Test case	User authorisation on unconnected environment. The object is protected, so can not be used without accessing keys.
Initial conditions	At least one license in the secure cache authorises the user to perform the action is trying to do. There is no protection info of the object in the Secure Cache
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is offline and the user wants to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The object is protected, the PMS Client requests to the SecureCache the Protection Information, but it is not present. 4 The authorisation is negative because can not unprotect the object and the user is not authorised.
Expected results	A bitwise containing the corresponding error number if authorisation is rejected.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

12.4.2.6 User authorisation on semiconnected environment (PMS server online, AXCS offline)

TCId	TC12.4.2.6a

Test case	User authorisation on semi-connected environment
Initial conditions	At least one license in PMS server authorises the user to perform the action is trying to do. The object is unprotected, so can be used without accessing keys.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is online, but AXCS is offline and you want to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The PMS Client sends the request to the PMS Server. 4 In the PMS Server, the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 5 The authorisation support performs the authorisation. 6 The result of the authorisation is positive the user is authorised. 7 A complete ActionLog is stored in the secure cache (locally).
Expected results	A bitwise containing zero.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.6b
Test case	User authorisation on semi-connected environment
Initial conditions	No licenses in PMS server authorises the user to perform the action is trying to do. The object is unprotected, so can be used without accessing keys.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is online, but AXCS is offline and you want to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The PMS Client sends the request to the PMS Server. 4 In the PMS Server, the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 5 The authorisation support performs the authorisation. 6 The result of the authorisation is negative and the user is not authorised.
Expected results	A bitwise containing the corresponding error number if authorisation is rejected.
Variations	None

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

TCId	TC12.4.2.6c
Test case	User authorisation on semi-connected environment
Initial conditions	At least one license in PMS server authorises the user to perform the action is trying to do. The object is protected, so can not be used without accessing keys.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is offline and the user wants to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The object is protected, the PMS Client requests to the SecureCache the Protection Information and is not present. 4 The authorisation is negative because can not unprotect the object and the user is not authorised
Expected results	A bitwise containing the corresponding error number if authorisation is rejected.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

12.4.2.7 User authorisation on fully connected environment (PMS server online, AXCS online)

TCId	TC12.4.2.7a
Test case	User authorisation on connected environment
Initial conditions	At least one license in PMS server authorises the user to perform the action is trying to do. But there are Action Logs pending in the Secure Cache
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is online, and AXCS is online and you want to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The PMS Client realises that there are pending Actions Logs in Secure Cache. 4 The user is not authorised and returns an error exposing that a Verification

	must be done.
Expected results	A bitwise containing the corresponding error number if authorisation is rejected.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.7b
Test case	User authorisation on connected environment
Initial conditions	At least one license in PMS server authorises the user to perform the action is trying to do. The object is unprotected, so can be used without accessing keys.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is online, and AXCS is online and you want to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The PMS Client sends the request to the PMS Server. 4 In the PMS Server, the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 5 The authorisation support performs the authorisation. 6 The result of the authorisation is positive and the user is authorised 7 A complete ActionLog is stored in the AXCS. 8 The ActionLog is returned, with the authorisation result, to the PMS Client. 9 The history hash is calculated locally, using the received Action Log and the previous history hash, and the history hash is stored in the SecureCache. Notice that this Action Log is not stored in the Secure Cache.
Expected results	A bitwise containing zero.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.7c
Test case	User authorisation on connected environment
Initial conditions	There is no license in PMS server that authorises the user to perform the action is trying to do. The object is unprotected, so can be used without accessing keys.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3

Steps	1 The PMS server is online, and AXCS is online and you want to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The PMS Client sends the request to the PMS Server. 4 In the PMS Server, the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 5 The authorisation support performs the authorisation. 6 The result of the authorisation is negative and the user is not authorised. 7 The ActionLog is returned, with the authorisation result, to the PMS Client.
Expected results	A bitwise containing the corresponding error number if authorisation is rejected.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.7d
Test case	User authorisation on connected environment
Initial conditions	At least one license in PMS server authorises the user to perform the action is trying to do. The object is protected, so can not be used without accessing keys.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is online, and AXCS is online and you want to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The PMS Client sends the request to the PMS Server. 4 In the PMS Server, the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 5 The object is protected, the PMS Server requests to the AXCS Server the Protection Information and is found. 6 The authorisation support performs the authorisation. 7 The result of the authorisation is positive the user is authorised. 8 A complete ActionLog is stored in the AXCS. 9 The ActionLog is returned, with the authorisation result, and the Protection Information to the PMS Client. 10 The history hash is calculated locally, using the received Action Log and the previous history hash, and the history hash is stored in the SecureCache. Notice that this Action Log is not stored in the Secure Cache
Expected results	A bitwise containing zero.
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

TCId	TC12.4.2.7e

Test case	User authorisation on connected environment
Initial conditions	At least one license in PMS server authorises the user to perform the action is trying to do. The object is protected, so can not be used without accessing keys.
Configuration description	None
Description of functionality to be tested	When a user wants to perform an action over a resource, the authorisation server checks if the user has the appropriate permissions according to the license terms.
Partners, people involved	Distributors, end-users.
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport3
Steps	1 The PMS server is online, and AXCS is online and you want to perform an action. 2 The PMS Client receives an authorisation request that includes an ActionLog with the user identification, the right, the resource and optionally the license(s) or its (their) identifier(s). 3 The PMS Client sends the request to the PMS Server. 4 In the PMS Server, the authorisation support obtains the licenses associated to the user from the secure cache of DRM licenses. 5 The object is protected, the PMS Server requests to the AXCS Server the Protection Information and is not found. 6 The result of the authorisation is negative so the user is not authorised. 7 The authorisation result is returned to the PMS Client.
Expected results	A bitwise containing the corresponding error number if authorisation is rejected
Variations	None
Issues	None
Test case Scope/Type	GUI and Backend (license verification) / BlackBox

12.4.2.8 Navigation on licensing information

TCId	TC12.4.2.8
Test case	DRM information needs to be browsed
Initial conditions	There is at least one license
Configuration description	None
Description of functionality to be tested	DRM E&V Interface.
Partners, people involved	End user, Creator
Validator(s) skill	User skill
Data set used	AXDS-DRMSupport1
Steps	1 Open DRM Editor and Viewer 2 Open license file (menu file) 3 Browse its properties
Expected results	A clear display with the DRM information
Variations	• DRM Editor and Viewer is visualized from the axeditor program
Issues	None
Test case Scope/Type	GUI

12.4.2.9 Rights Expression Translator

TCId	TC12.4.2.9
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	4 Select a source license 5 Execute the Rights Expression Translator 6 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

12.4.2.10 License migration

TCId	TC12.4.2.10a
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	TC12.4.2.10b
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	None

description	
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

12.4.2.11 Cooperative Authorisation Check

TCId	TC12.4.2.11
Test case	Cooperative Authorisation Check
Initial conditions	At least two license are needed, and they complete a chain of authorization e.g. (end-user license + licensing license)
Configuration description	PMS Server is aware of other PMS Servers reachable through the network
Description of functionality to be tested	The PMS Network it is able to process a license chain.
Partners, people involved	An actors that needs to obtain an authorization
Validator(s) skill	DRM expert
Data set used	AXDS-DRMSupport4
Steps	1 The user has obtained an end-user license. 2 The user wants to consume content, being related to this license, so the AXMEDIS Player (or another AXMEDIS compliant player) contacts the related PMS Server in order to obtain authorization 3 The PMS Server searches in the licence DB for a proper license. 4 The PMS finds a valid license issued by a distributor. 5 The PMS Server searches in the licence DB for a proper license stating that the distributor is authorized issuing licenses of that content. 6 The PMS does not find such a license 7 The PMS contacts the nearby PMS Servers asking to check for the issuing authorization 8 One of the PMS, which holds a suitable license, responds to the calling PMS with an affirmative answer. 9 The PMS responds to the AXMEDIS Player with affirmative result.
Expected results	The end-user is authorized
Variations	Multi-level chain of licence (>2)
Issues	None
Test case Scope/Type	GUI and Backend (license storage) / BlackBox

12.5 Encryption/Decryption Support (FUPF)

12.5.1.1 Encryption

TCId	TC12.5.1.1a
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	TC12.5.1.1b
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

12.5.1.2 Decryption

TCId	TC12.5.1.2a
Test case	Decryption of AXMEDIS object
Initial conditions	None
Configuration description	None
Description of	An actor wants to open an AXMEDIS protected object

functionality to be tested	
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	TC12.5.1.2b
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

12.5.1.3 Encryption of symmetric key

TCId	TC12.5.1.3
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

12.5.1.4 Decryption of symmetric key

TCId	TC12.5.1.4
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

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

13.1.7 AXMEDIS Plug-in for Mozilla (SEJER)

These test cases apply to both Mozilla and Firefox, herein defined as “the browser”.

13.1.7.1 Loading an object from the src attribute of the OBJECT tag

TCId	TC13.1.7.1
Test case	Loading an object from the src attribute of the EMBED tag
Initial conditions	The browser is open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed

Description of functionality to be tested	The user opens an HTML page containing an EMBED tag, with the type attribute defined on application/axmedis-object" and the src set to the URL to an AXMEDIS object.
Partners, people involved	Actor that can use the AXMEDIS Player
Validator(s) skill	Low, End User
Data set used	AXDS-DB2, AXDS-DB3
Steps	1 The Actor loads an HTML with the plug-in
Expected results	The plug-in display the requested resource of the AXMEDIS Object.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

13.1.7.2 Loading an object from the address bar of the Browser

TCId	TC13.1.7.2
Test case	Loading an object from the address bar of the Browser
Initial conditions	The browser is open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed
Description of functionality to be tested	The user type in the address bar of the browser the URL to an AXMEDIS Object
Partners, people involved	Actor that can use the AXMEDIS Player
Validator(s) skill	Low, End User
Data set used	AXDS-DB2, AXDS-DB3
Steps	1 The Actor type in the address bar the URL to an AXMEDIS Object
Expected results	The plug-in display the requested resource of the AXMEDIS Object.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

13.1.7.3 Checking that the plug-in properties are correctly exposed through attributes on the EMBED tag

TCId	TC13.1.7.3
Test case	Checking that the plug-in properties are correctly exposed through attributes on the EMBED tag
Initial conditions	The browser is open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed
Description of functionality to be tested	The user opens an HTML page containing an EMBED tag, with the type attribute defined on application/axmedis-object" and the src set to the URL to an AXMEDIS object. The properties of the plug-in are defined in attributes of the EMBED tag.
Partners, people involved	Actor that can use the AXMEDIS Player

Validator(s) skill	Low, End User
Data set used	AXDS-DB2, AXDS-DB3, AXDS-MozillaPlugin
Steps	1 The Actor loads an HTML with the plug-in and attributes set
Expected results	The plug-in display the requested resource of the AXMEDIS Object. And the player is in the defined configuration (hierarchy view opened or closed, background properly set, etc.)
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

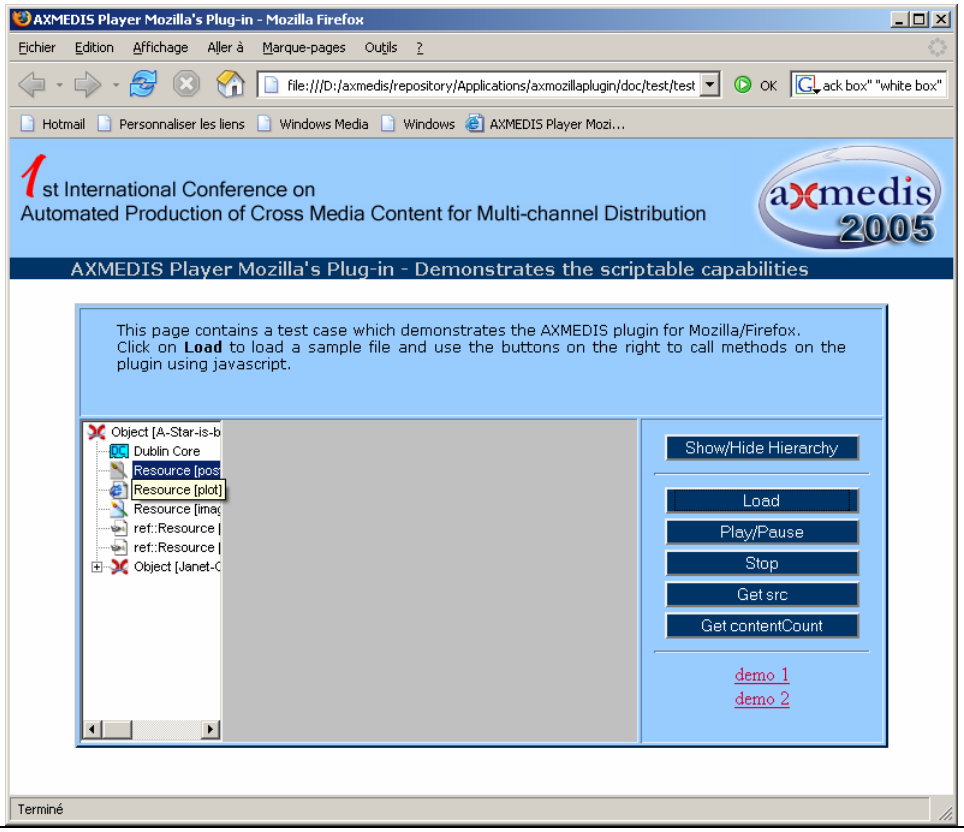
13.1.7.4 MPEG21 conformance of the URL

TCId	TC13.1.7.4a
Test case	Loading an object on a specific resource using the fragment notation on the URL, like myobject.axm #item_name='resource.ext'
Initial conditions	The browser is open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed
Description of functionality to be tested	The user type in the address bar of the browser the URL to an AXMEDIS Object on a specific resource using its name.
Partners, people involved	Actor that can use the AXMEDIS Player
Validator(s) skill	Low, End User
Data set used	AXDS-DB3
Steps	1 The Actor type in the address bar the URL to an AXMEDIS Object, appending the fragment to open a specific resource by name
Expected results	The plug-in display the requested resource of the AXMEDIS Object.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

TCId	TC13.1.7.4b
Test case	Loading an object on a specific resource using the fragment notation on the URL, like myobject.axm#item_id=0
Initial conditions	The browser is open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed
Description of functionality to be tested	The user type in the address bar of the browser the URL to an AXMEDIS Object on a specific resource using its index in the hierarchy.
Partners, people involved	Actor that can use the AXMEDIS Player
Validator(s) skill	Low, End User
Data set used	AXDS-DB3
Steps	1 The Actor type in the address bar the URL to an AXMEDIS Object, appending the fragment to open a specific resource by index
Expected results	The plug-in display the requested resource of the AXMEDIS Object.
Variations	None

Issues	None
Test case Scope/Type	GUI / BlackBox

13.1.7.5 JavaScript compatibility with the ActiveX

TCId	TC13.1.7.5
Test case	Test all exported functions in both IE and Mozilla
Initial conditions	The browser is open, Internet Explorer is open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed. The AXMEDIS ActiveX is properly installed A test HTML page is available, compatible on both Internet Explorer and Mozilla Firefox. Every JavaScript function is attached to a button to allow the Actor to test them. 
Description of functionality to be tested	The user creates an HTML that can be displayed in both Internet Explorer and Mozilla/Firefox and writes JavaScript functions which, once the plug-in/object is retrieved, work on both browsers.
Partners, people involved	Actor that can use the AXMEDIS Player
Validator(s) skill	Medium, Experienced User
Data set used	AXDS-DB2, AXDS-MozillaPlugin
Steps	1 The Actor clicks on each button to check every function.
Expected results	There is no JavaScript error.
Variations	None
Issues	None

Test case Scope/Type	GUI / BlackBox

13.1.8 AXMEDIS Player based on Mozilla (SEJER)

The AXMEDIS player based on Mozilla is basically a XULRunner application build around the AXMEDIS Mozilla's plug-in.

13.1.8.1 The functions exposed by the plug-in are wired in the player

TCId	TC13.1.8.1
Test case	The functions exposed by the plug-in are wired in the player
Initial conditions	The AXMEDIS Mozilla based player is open
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed
Description of functionality to be tested	The functions exposed by the plug-in are wired in the player, i.e. there is a button in the toolbar or a menu to launch it.
Partners, people involved	Actor that can use the AXMEDIS Mozilla based player
Validator(s) skill	Medium, Experienced User
Data set used	AXDS-DB2, AXDS-MozillaPlayer
Steps	1 Click on every menu and button to ensure a response from the user interface.
Expected results	The command is launched.
Variations	None
Issues	None
Test case Scope/Type	GUI / BlackBox

13.1.8.2 The locale of the player can be changed

TCId	TC13.1.8.2
Test case	The locale of the player can be changed
Initial conditions	The AXMEDIS Mozilla based player is closed
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed
Description of functionality to be tested	The user interface should be displayed in different languages
Partners, people involved	Actor that can use the AXMEDIS Mozilla based player
Validator(s) skill	Medium, Experienced User
Data set used	AXDS-DB2, AXDS-MozillaPlayer
Steps	1 Launch the player, check the locale 2 Close the player 3 Change the locale line in the chrome\chrome.manifest file 4 Launch the player again.
Expected results	The user interface reflects the new locale
Variations	None
Issues	None

Test case Scope/Type	GUI / WhiteBox

13.1.8.3 The skin of the player can be changed

TCId	TC13.1.8.3
Test case	The skin of the player can be changed
Initial conditions	The AXMEDIS Mozilla based player is closed
Configuration description	An AXMEDIS Player must be available. The Player must be properly installed and configured. The AXMEDIS Mozilla plug-in is properly installed
Description of functionality to be tested	The user interface should be displayed using different skins
Partners, people involved	Actor that can use the AXMEDIS Mozilla based player
Validator(s) skill	Medium, Experienced User
Data set used	AXDS-DB2, AXDS-MozillaPlayer
Steps	1 Launch the player, check the skin 2 Close the player 3 Change the skin line in the chrome\chrome.manifest file 4 Launch the player again.
Expected results	The user interface reflects the new skin
Variations	None
Issues	None
Test case Scope/Type	GUI / WhiteBox

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	TC14.1.2
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 refund a transaction

TCId	TC14.1.8
Test case	Mediaclub refund 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 User login

14.3.1 Authentication through an external SSO system

TCId	TC14.3.2
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

14.4 Catalogue Browsing

14.4.1 Catalogue listing

TCId	TC14.4.1
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

14.4.2 Available resources listing

TCId	TC14.4.3
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

14.4.3 User Page

TCId	TC14.4.5
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

14.5 Catalogue Content Purchase

14.5.1 Content Fetching

TCId	TC14.5.1
Test case	Content Fetching
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

14.5.2 User Authentication Form

TCId	TC14.5.2
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

14.5.3 Catalogue Content Transaction

TCId	TC14.5.3
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

14.5.4 Content Access

TCId	TC14.5.4
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

14.5.5 Content Preview

TCId	TC14.5.5
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

14.5.6 License Acquisition

TCId	TC14.5.6
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

14.5.7 Multi-device license activation and back-up

TCId	TC14.5.7
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

14.5.8 Pre-ordering and registration for a group of students

TCId	TC14.5.8
Test case	
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

14.6 Business Models

14.6.1 Rental

TCId	TC14.6.1
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

14.6.2 Pay per download

TCId	TC14.6.2

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

14.6.3 Sell through

TCId	TC14.6.3
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

14.6.4 Subscription

TCId	TC14.6.4
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

14.6.5 Pay per minute

TCId	TC14.6.5
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

14.6.6 Pay per day

TCId	TC14.6.7
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

14.6.7 Pay per credits

TCId	TC14.6.8
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	2 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

14.7 Advanced payment methods

14.7.1 Wallet

TCId	TC14.7.1
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

14.7.2 Gift certificates

TCId	TC14.7.2
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

15 AXMEDIS for Distribution towards Mobiles

15.1 General Assumptions and notes to Architecture

1) The AXMEDIS enabled ILABS, IRC distribution system includes:
   a) An AXMEDIS network node, which:
      i) Automatically fetches all AXMEDIS objects matching pre-set criteria; licensing attributes, content type, time-span, etc.
      ii) Makes all fetched content and assets available for immediate use, providing online availability of ready-to-use files in specific formats (WMA, MIDI, etc).
      iii) Maintains a list of all files available for use from local storage.
      iv) Automatically synchronizes object and content expiration, and license changes with the AXMEDIS network.
   b) The ILABS, IRC APS (Application Server), with integrated Personalization (PE) and Handset Management engines (HME).
   c) A plug-in that interacts with the AXMEDIS platform, encapsulating and simplifying the platform functionality for the ILABS, IRC servers and components.
2) The AXMEDIS enabled ILABS, IRC Transcoding Server includes:
   a) A Transcoding Server, which manages the transcoding logic and routines.
   b) A plug-in that interacts with the AXMEDIS platform, encapsulating and simplifying the platform functionality for the ILABS, IRC servers and components.
   c) A Transcoding platform including Codecs, configuration and Interface.
3) Categories:
   a) Category: this object is defined within Mobile Application. It is meant as an "area of interest" and it consists of:
      i) a category key, that has to unique inside the Mobile Application
      ii) a sequence of category names, one for each language supported by the Mobile Application; these names are the ones that will be seen by Mobile Application users
      iii) an associated query to retrieve contents for category

15.2 Use cases

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 User (performing the operation)
2. The Mobile Admin (performing the execution monitoring or any other specific administrative operation)

In the test description when “User” is used with no other specification it is always the “end user” operating. When the “Mobile Admin” is directly involved then the word user is not mentioned. It is given for granted that when the “User” is operating and the ”Mobile Admin” 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 Mobile Admin) 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 Mobile Admin 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.

15.2.1 Domain registration

TCId	TC15.2.1
Test case	Domain registration
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • Domain Management module • AXMEDIS Communication module (PMS related module only) • PMS Domain
Description of functionality to be tested	The Mobile Admin creates and registers a new domain within the AXMEDIS system
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile2, AXDS-Mobile5
Steps	1 The Mobile Admin accesses to a reserved section of the Portal and generates a new domain identifier 2 The Mobile Front End forwards the identifier to the Mobile Back End 3 The Mobile Back End forwards received data to the Domain Management module 4 The Domain Management module prepares a “Domain object” that will be passed to the PMS Domain through the PMS Communication module in order to register the newly created domain within the AXMEDIS system 5 The PMS Domain registers the newly created domain within the AXMEDIS system
Expected results	The new domain is registered and operational
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.2 Content Preparation/ingestion

TCId	TC15.2.2
Test case	Content Preparation/ingestion
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End

	• Content Management module • Content Retriever • AXMEDIS Communication module • Query Support • AXCP • DB Management module
Description of functionality to be tested	AXMEDIS/other objects are ingested and packed for delivery
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 The mobile admin activates the ingestion procedure 2 Content is adapted according to specified rules and available rights (and eventually transformed into AXMEDIS objects if not already so) 3 AXMEDIS object are packed into IMS Content Packages, where each item refers to the resource that is represented by AXMEDIS object.
Expected results	Content is ingested and post-processed
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.3 Content Retrieving Criteria Management

TCId	TC15.2.3
Test case	Content Retrieving Criteria Management
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • Catalogue Management module • DB Management module
Description of functionality to be tested	The Mobile Admin defines catalogue categories, which represent possible areas of interest, such as art, science, history and so on.
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.

Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-DB1, AXDS-DB2
Steps	1 The Mobile Admin defines catalogue categories 2 For each category: 2.1 Assigns a "category name" 2.2 Defines related retrieving query. 4 Defines active categories (that means visible by users).
Expected results	Updated content retrieving criteria are operational
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.4 Content Retrieving Criteria Definition

TCId	TC15.2.4
Test case	Content Retrieving Criteria Definition
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • Catalogue Management module • DB Management module
Description of functionality to be tested	The Mobile Admin creates and defines content retrieving definitions and related queries
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-DB1, AXDS-DB2
Steps	1 The Mobile Admin interacts with the Front-End which to start the content retrieving criteria definition activity 2 The request is forwarded to the Back End that replies with a Criteria definition query form 3 The mobile manager fills requested data and confirms 4 Inserted data are checked by the Mobile Back-End and forwarded to the Catalogue Management module 5 The Catalogue management module interacts with the Database Management module in order to store queries.
Expected results	Created content criteria definition are available for update
Variations	• Catalogue Management module could cache categories queries in order to avoid communication operations with database to retrieve them later
Issues	A category requires a unique key within the whole categories set since it is used to retrieve corresponding query The Mobile Admin is also requested to give translation of category name for each language supported by the system since users won't see category key but a category name properly translated
Test case Scope/Type	GUI, Backend / BlackBox

15.2.5 Content Retrieving Criteria Selection

TCId	TC15.2.5
Test case	Content Retrieving Criteria Selection
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • Catalogue Management module • DB Management module
Description of functionality to be tested	The Mobile Administrator decides which of the categories previously defined and stored into the system are actually seen by users
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-DB1, AXDS-DB2
Steps	1 The Mobile Administrator asks to the Front End the list of previously defined categories 2 The request is dispatched to the Back End and Catalogue Management modules 3 The latter retrieves from database the list of categories and their current status (active/ not active) 4 The generated List is passed back and shown to the Mobile Admin 5 The mobile admin chooses whether updating or not categories status and then sends confirmation (whenever needed). 6 The Categories status (active/not active) is then updated in the database.
Expected results	Selected content retrieval criteria are operational
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.6 Content Retrieving Criteria Removing

TCId	TC15.2.6
Test case	Content Retrieving Criteria Removing
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • Catalogue Management module • DB Management module

Description of functionality to be tested	The Mobile Administrator removes content selection criteria previously stored into the system.
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-DB1, AXDS-DB2
Steps	1 The Mobile Admin through the Front End requests to access to the list of inserted criteria 2 The request is handed over to the Back End module that recovers data from the database 3 The Mobile Admin selects the criteria to delete and confirms 4 The request is processed and selected criteria are removed.
Expected results	Selected content retrieval criteria are no more operational
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.7 Supported device profile adding

TCId	TC15.2.7
Test case	Supported device profile adding
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • Device Profile Management module • Content Management module • AXMEDIS Communication module • Query Support • AXCP • DB Management module
Description of functionality to be tested	The Mobile Admin adds a profile for a newly supported device
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile7
Steps	1 The Mobile Admin requests insertion of a newly supported device profile to the Mobile Front End 2 The Mobile Front End forwards the request to the Mobile Back End which in

	turns returns the data structure to be filled in. 3 The Mobile Admin fills all required form fields and submits insertion to the Front End. 4 The Request is again forwarded to the Back End, which performs basic data checking including one with the User Management. 5 The User Management module communicates the request to the Device Profile Management 6 The Device Profile Management creates a Device Profile object from inserted data and asks for inserting it into local database. 6 The storage is performed and a batch procedure is launched to adapt available objects for new device profile.
Expected results	Newly supported profile device is operational
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.8 Supported device profile removing

TCId	TC15.2.8
Test case	Supported device profile removing
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • Device Profile Management module • Content Management module
Description of functionality to be tested	The Mobile Admin decides/needs to remove some of the supported device profiles previously defined.
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile7
Steps	1 The Mobile Admin asks the front end the list of supported device profiles 2 The front end retrieves, thanks to backend, the list of supported device profiles 3 The Mobile Admin chooses among them the one to delete and requires deletion. 4 The request is forwarded to the backend that processes the request and removes that device profile (already adapted content is not deleted but simply marked as not useful)
Expected results	Selected profile devices are no more operational
Variations	• It is foreseen as possible that the Mobile Admin requires a combination of device profile and related adapted content deletion. In this case the procedure would be more complex as to be positively completed would also require content deletion completion
Issues	Some kind of marking is required for contents associated to the removed device

	profiles. This is specifically needed to avoid loss of content and yet prevent useless computational effort during search and retrieval phases related to content offering preparation.
Test case Scope/Type	GUI, Backend / BlackBox

15.2.9 User registration by administrator

TCId	TC15.2.9
Test case	User registration by administrator
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • User Management module • AXMEDIS Communication module • AXCS • DB Management module
Description of functionality to be tested	The Mobile Admin registers a new user (upon specific request maybe due to a failure in connection by user during registration and consequent incomplete pending operational status of the procedure)
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile5, AXDS-Mobile7
Steps	1 The Mobile Admin requests the front end to retrieve the registration form 2 Once retrieved the Mobile Admin fills-in all mandatory data and confirms 3 The front end module forwards imputed data to the user management module 4 The user management module forwards needed data to the AXCS. 5 The AXCS registers the new user and provides a positive ACK 6 The positive confirmation is sent back to the Mobile Admin 7 The Mobile Admin sends a formal registration notice to the requesting new user
Expected results	Specified user is now registered
Variations	• The user self-registers.
Issues	The present case is foreseen not just for the sake of completeness, but also because in some context (business contest mainly) it is usually the practice to have a service subscription validation prior to the subscription. User registration by Mobile Admin could be activated, for example, via e-mail (request coming from the user willing to be registered).
Test case Scope/Type	GUI, Backend / BlackBox

15.2.10 User update by administrator

TCId	TC15.2.10
Test case	User update by administrator

Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • User Management module • AXMEDIS Communication module • AXCS • DB Management module
Description of functionality to be tested	The Mobile Administrator updates users' data
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile5, AXDS-Mobile7
Steps	1 The Mobile Admin requests the user management module the list of registered users 2 The Mobile Admin selects the user whose data needs update 3 The Mobile Admin modifies and confirms user's data 4 The new data are updated into the system
Expected results	Specified user data have been updated
Variations	• The user self-changes provided data.
Issues	The present case is foreseen not just for the sake of completeness, but also because in some context (business contest mainly) it is usually the practice to have a service subscription validation prior to the subscription. User data update by Mobile Admin could be activated, for example, via e-mail (request coming from the user being registered).
Test case Scope/Type	GUI, Backend / BlackBox

15.2.11 User remove by administrator

TCId	TC15.2.11
Test case	User remove by administrator
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • User Management module • AXMEDIS Communication module • AXCS • DB Management module

Description of functionality to be tested	The Mobile Admin proceeds to remove a user from the system
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile2
Steps	1 The Mobile Admin requests the User management module to remove a user from the system, 2 The user management module requests the AXCS to perform an update of the "finalRegDeadline" parameter into user's registration data 3 The AXCS confirms the performed update process 4 User's data are deleted from local database 5 The Mobile Admin sends an email to the user to notify deletion.
Expected results	Selected user is no more registered
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.12 User roles management

TCId	TC15.2.12
Test case	User roles management
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Mobile Front End • Mobile Back End • User Management module • DB Management module
Description of functionality to be tested	The Mobile Administrator can change a user's role, for example assigning also an administrator role (users can have more than one role at the time)
Partners, people involved	Mobile Admin, is a registered AXMEDIS user with a specific UID and has all the right and tools to perform the operation
Validator(s) skill	The Mobile Admin 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, is a registered AXMEDIS user with a specific UID, the operation is performed in the factory.
Data set used	AXDS-MCTestUser, AXDS-Mobile2
Steps	1 The Mobile Admin requests the user management module the list of registered users 2 The Mobile Admin selects the user whose role data needs update 3 The Mobile Admin modifies and confirms user's data 4 The new data are updated into the system
Expected results	Selected users' role has been positively updated/enhanced

Variations	• Instead of adding a role to a user it could be removed
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.13 User registration

TCId	TC15.2.13
Test case	User registration
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • User Management module • AXMEDIS Communication module • AXCS • DB Management module
Description of functionality to be tested	The User has to register before being able to access and use AXMEDIS Mobile Application
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile5, AXDS-Mobile7
Steps	1 The user interacts with the Application Front End in order to retrieve a registration form. There are two types of data to insert: the elements that should be filled by every user wishing to register (mandatory elements) and the elements that would be very useful whenever the user decides to fill in (recommended elements). 2 The user inputs the data and confirms; the registration request is forwarded to the backend 3 The backend performs some consistency and validity checks. If provided data are formally corrected they are passed to User Management module 4 The User management module makes a first check to verify if the user is already present into the local DB to avoid useless duplications. 5 The User management module sends a registration request to the AXCS via the AXCS Communication module, which function is to properly format and forward requests to the AXCS 6 If the user registration is successful then user data are stored locally into local database. 7 The user management module, at the end of the registration process, sends an email to the user as feedback to communicate the attributed AXMEDIS user identifier.
Expected results	The user is successfully registered both locally and on AXMEDIS
Variations	• The user has already registered himself into AXMEDIS through another distributor so s/he has to register only for the Mobile Application now. Therefore the Registration form should allow inserting also an AXUID if the user already owns one. In this case communication with AXCS is limited to the notification of the new subscribed distributor if not needed at all.

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

15.2.14 Certification of users

TCId	TC15.2.14
Test case	Certification of users
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • User Management module • AXMEDIS Communication module • PMS
Description of functionality to be tested	A user wishes to be certified (with own device) in order to access to AXMEDIS objects
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile5, AXDS-Mobile7
Steps	1 A user requires to be certified (with own device) in order to access to AXMEDIS objects 2 The front end fetches the request form from the backend and presents it to the user 3 The user has to provide (directly or indirectly) the needed information to the application that forwards user and tool information to the Domain PMS 4 The Domain PMS performs all necessary checks and operations 5 In case of positive result, the Domain PMS returns acknowledge to the requesting application 7 The front end receives the positive feedback and provides a proper feedback to the user.
Expected results	The user (and related device) is properly certified
Variations	None
Issues	Tool-user-device become linked once certified, as an AXMEDIS tool is certified for an AXMEDIS user on an AXMEDIS device.
Test case Scope/Type	GUI, Backend / BlackBox

15.2.15 Client application download

TCId	TC15.2.15
Test case	Client application download
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End
Description of	User downloads AXMEDIS Player on own devices in order to be able to use

functionality to be tested	AXMEDIS contents
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile2, AXDS-Mobile7
Steps	1 The Users logs onto the Mobile Application Portal 2 The user requires download instructions to the front end 3 The front end requires instructions provision to the backend 4 Instructions are returned to the front end that takes care to present them to the user 5 The user prepares the device for connection and download 6 The user connects to "the network" (here this has to be regarded in the wider sense of the term) using his personal device and download the AXMEDIS client application
Expected results	The AXMEDIS player is downloaded on the user device
Variations	• A more complete case foresee the download, install and usage
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.16 User login

TCId	TC15.2.16
Test case	User login
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • User Management module • DB Management module
Description of functionality to be tested	A user logs to the Mobile Application Portal
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 A user wishes to use AXMEDIS Mobile functionalities 2 The user performs a login to the mobile portal by accessing the login form and providing login and password chosen at registration time 3 The Mobile front end receives the imputed data and passes it to the Mobile Back End 4 The Mobile backend performs some basic checks and then further forwards

	received data to the User Management module 5 The user management module requests the User Database Management module to verify user credentials 6 Upon confirmation of successful data match, access to the Mobile Portal is granted
Expected results	The user is properly logged onto the mobile portal
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.17 User interface language selection

TCId	TC15.2.17
Test case	User interface language selection
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • User Management module • DB Management module
Description of functionality to be tested	A user wants to change the current language used by the application
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 The user interacts with the Front End requesting a change in GUI language 2 The front end forwards the requested change to the Back End 3 The list of supported languages is retrieved and passed back to the front end 4 The front end displays the list of supported languages 5 The user chooses a language and from this point onwards all pages will be visualized using the selected language.
Expected results	The GUI is in the selected language
Variations	• If a user is already logged then the currently selected language is updated also into user's profile, elsewhere language changes only for visualization.
Issues	At any time a user can change the current language used by the application, yet this may imply some change in the usage scenario which may imply (but not necessarily) a potential thread as a change in interface during operation is far less usual (as a behaviour) than at initial (login) time and this could be due to a change in person performing the operation.
Test case Scope/Type	GUI, Backend / BlackBox

15.2.18 Catalogue loading and browsing

TCId	TC15.2.18
Test case	Catalogue loading and browsing

Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • Catalogue Management module • Content Management module • Centent Retrieve • AXMEDIS Communication module • Query support • DB Management module
Description of functionality to be tested	A user wants to access contents browsing the Mobile Catalogue and related categories
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 A user wishes to use AXMEDIS Mobile functionalities to access to content 2 The user performs a login to the mobile portal 3 Once granted access the front end requires the back end to provide the list of available content categories 4 Requests for content categories received at the back end are dispatched to the Catalogue Management module and categories are retrieved from the local database. 5 Retrieved list of foreseen content categories is sent back to the front end that takes care to display it to the user 6 The user chooses a category, and such information (received by the front end) is passed to the back end 7 Related contents are retrieved both in the Lobster and via Query Support (and related communication modules). 8 A list of the available content is generated and sent back to the front end that presents it to the user 9 The user is now able to browse the list of available content (in the selected category)
Expected results	The user is able to browse the catalogue categories, related content and select content for purchase
Variations	• The Catalogue Management module could cache locally queries associated to categories in order to avoid retrieving them from database every time they are needed • In order to achieve better performances, the Catalogue Management could cache contents data in order to be able to answer directly to further requests and to avoid calling the Database Management and the Content Retriever modules functionalities again later. This certainly applies to non AXMEDIS data, on the other hand for AXMEDIS data references used for retrieval in local AXDB will be cashed in order to speed up the retrieval process as described previously.

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

15.2.19 Contents Search

TCId	TC15.2.19
Test case	Contents Search
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • Content Management module • Centent Retrieve • AXMEDIS Communication module • Query on demand • DB Management module
Description of functionality to be tested	A user wants to get more contents than the ones available and listed on the Mobile Portal by categories selection, therefore performs a search
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 A user may wish to get more contents than the ones available and listed on the Mobile Portal by categories selection 2 To retrieve other contents the user can exploit the Search Contents page and fill-in query form parameters. 3 The query is dispatched locally and remotely through the Query on Demand module. 4 Received results are turned into an object list by the backend 5 The retrieved content list is presented to the user by the front end
Expected results	The user is able to browse retrieved contents, related info and select content for purchase
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.20 Getting content information

TCId	TC15.2.20
Test case	Getting content information
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • Catalogue Management module

	• Content Management module • Centent Retrieve • AXMEDIS Communication module • Query support • DB Management module
Description of functionality to be tested	The user asks for more information about a specific content
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 A user is interested in a specific content found on any Portal pages 2 The user asks for more information about that content 3 The request is received by the front end module that dispatches to the Catalogue Management via the backend 4 The catalogue management retrieves content metadata from the local database or remotely through the Query Support 5 Retrieved data are sent back to the front end for display to user 6 The front end presents the user received data
Expected results	The user is able to access to requested content info
Variations	• Catalogue Management module already cached contents on catalogue loading, therefore no forwarding of requests is needed towards the Content Retriever and Query Support modules.
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.21 Content Preview

TCId	TC15.2.21
Test case	Content Preview
Initial conditions	• All components are active and properly functioning, • Connection with remote systems necessary to operation fulfillment are in place and operational • A content preview is available (either as a specific option or as a capability of the player)
Configuration description	• Mobile Front End • AXMEDIS Player • PMS
Description of functionality to be tested	User wants to have a preview of a specific content before purchase
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks

	on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 The User is logged in and currently browsing the catalogue 2 The user wants to have an idea on a specific content before purchasing a fruition license. 3 The user asks for a "preview" of the content 4 The front end receives the request 5 The front end asks the AXMEDIS Player on the user device to display content preview
Expected results	The user has been able to preview content (whenever possible)
Variations	• The player is not able to perform a preview and therefore specific "preview" objects have been prepared at ingestion time
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.22 Content Delivery

TCId	TC15.2.22
Test case	Content Delivery
Initial conditions	• All components are active and properly functioning, • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • Content Management module • Centent Retrieve • User Management module • AXMEDIS Communication module • Query support • DB Management module
Description of functionality to be tested	Content is delivered to the user according to acquired license rights
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 The Front End interacts with the Back End to activate the delivery 2 The back end forwards requests to the Content Management and Retriever modules. 3 Content is retrieved either from the AXDB 4 Also User Profile and Device Profile are retrieved as they are needed to have a format suitable for user device and preferences 5 Content is adapted via AXCP 6 Adapted content is delivered

Expected results	The user has been able to have the selected content delivered and ready for use according to acquired rights
Variations	• Content is retrieved from the Lobster as it not an AXMEDIS object
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.23 Content Acquire

TCId	TC15.2.23
Test case	Content Acquire
Initial conditions	• All components are active and properly functioning, • Connection with remote systems necessary to operation fulfillment are in place and operational • The payment related modules are properly interconnected and operations • It is also assumed that all operation related to payments are positively terminated throughout the process.
Configuration description	• Mobile Front End • Mobile Back End • CAtalogue Management module • E-Commerce • AXMEDIS Communication module • PMS
Description of functionality to be tested	A user wants to acquire rights to execute some actions over an object (install, uninstall, play, modify, etc…).
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 The user requests the front end to purchase a content 2 The front end requests the back end the PARs list for the selected content 3 The back ends retrieves the requested data from the Catalogue Management module 4 Retrieved data are presented back to the user 5 The User chooses the set of rights to acquire from the provided list 6 The user is requested to confirm performed choice 7 Upon conformation the user is requested to provide needed payment data. As some of these data have already been provided at registration time, the user is asked either to confirm the purchase form 8 The user confirms 9 The order confirmation is forwarded to the e-commerce module 10 The E-Commerce module interfaces with the Bank payment module. 11 If payment is successful then the Catalogue Management module asks to the PMS Communication module to forward to the PMS Web Service a license generation request. 12 The PMS Communication module has to call three PMS methods in order to create a license: an initialization method, a method to add grants to license (one method call for each grant in license) and a finalization method, which

	returns new license identifier.
Expected results	The user has been able to purchase the desired content (and related fruition)
Variations	• As some of data related to content acquisition payments have not been provided at registration time, the user is asked to complete the purchase form
Issues	The set of rights that will be available to the user for purchase is a subset of the original Potential Available Rights (PAR) that depends on the following factors: o Original PAR; o Rights acquired by the distributor (eventually already restricted either by the rights owner at selling time or by any other distributor that is preceding the present one in the sale value chain); o Rights that the distributor decides to present the user as available (and that have to be less or equal to the acquired ones. Potential Available Rights (PAR) are defined by right owner who can then grant some rights combination to right purchaser. In case among acquired PAR there is the right to issue licenses or grant some rights (always a subset of the acquired ones), the new issuer can define the subset of PAR that will be available to the next actor in the chain. This brings to a nested set of restrictions that are supported by a chain of licenses each enforcing the correct set of rights and ensuring the possibility for the actor to properly use the object respecting the DRM chain. In the following example is sketched a simple sequence with a main owner, two chained distributors (for example EU and National level) and the user.
Test case Scope/Type	GUI, Backend / BlackBox

15.2.24 Content fruition

TCId	TC15.2.24
Test case	Content fruition
Initial conditions	• All components are active and properly functioning, • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • AXMEDIS Player • PMS
Description of functionality to be tested	The user accesses to selected content for fruition
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile3, AXDS-Mobile4, AXDS-Mobile5, AXDS-Mobile6, AXDS-Mobile7, AXDS-DB1, AXDS-DB2
Steps	1 The user should log onto the Mobile Portal 2 The user enters the special section where is stored the list of contents for which a license with some rights has been acquired. 3 The user selects a content title to play it 4 The AXMEDIS Player(s) present on the user device is activated 5 The AXMEDIS player(s) performs all needed rights checking before playing
Expected results	The user has been able to use the acquired content (and related fruition)
Variations	• The same operation could be performed for a different business model (pay

	per view, pay per access, pay per download…
Issues	None
Test case Scope/Type	GUI, Backend / BlackBox

15.2.25 User Data Update

TCId	TC15.2.25
Test case	User Data Update
Initial conditions	• All components are active and properly functioning, • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Mobile Front End • Mobile Back End • User Management module • AXMEDIS Communication module • AXCS • DB Management module
Description of functionality to be tested	A registered user modifies data previously inserted at registration time
Partners, people involved	• The User (performing the operation) • The Mobile Admin (performing the execution monitoring)
Validator(s) skill	• The User (no special requirement – qualitative evaluation of results is expected) • The Mobile Admin (ICT skills sufficient to examine logs and perform checks on local applications and AXMEDIS involved components)
Data set used	AXDS-MCTestUser, AXDS-Mobile1, AXDS-Mobile2, AXDS-Mobile5, AXDS-Mobile7
Steps	1 A registered user logs onto the system 2 The user requests to modify data previously inserted at registration time 3 The Mobile Portal offers a specific section where the user is prompted with inserted data. 4 The User can revise and update 5 Once ended the user either confirm changes (or aborts) 6 User new data are handled by the front end, which forwards request to the backend 7 The backend retrieves user data from the database through User Management and User Profile modules 8 Updated information is then forwarded and stored in the database.
Expected results	User data have been successfully updated
Variations	None
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 PC+DVB Card Terminal

TCId	TC16.1.1
Test case	PC+DVB Card Terminal
Initial conditions	The user has 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 (power, SNR, BER) 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 (if any external led is present); 3 The User launches the DVB Tuner Application, configures the satellite parameters 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 STB Terminal (MBI)

TCId	TC16.1.2
Test case	STB Terminal
Initial conditions	The user has a STB and 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 (power, SNR, BER) of the satellite from which the Satellite Data Broadcast will be received.
Configuration description	The satellite cable is properly connected to the DVB Adapter.
Description of functionality to be tested	The STB is properly configured 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 STB; 2 The User verifies the box blinking for the activity (if any external led is present); 3 The User sets the satellite configuration parameters for the reception and verifies the lock of the satellite transponder where the AXMEDIS Service is transmitting.
Expected results	The STB is correctly connected to the satellite cable; The STB light is blinking to indicate an activity (if any); The STB is able to lock the satellite transponder.

Variations	None
Issues	None
Test case Scope/Type	Backend / WhiteBox.

16.1.3 User Software Installation

TCId	TC16.1.3
Test case	User Software Installation
Initial conditions	The hardware installation procedure is completed. The spectrum analyser (used by the installer of the satellite dish) indicates a good quality of signal (power, SNR, BER) of the satellite from which the Satellite Data Broadcast will be received. The backend shall be running and sending some basic data.
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 AXMEDIS Service is transmitting.
Description of functionality to be tested	Well functioning of the AXMEDIS Client after installation: • The DVB Adapter or STB is able to lock the signal coming from the appropriate transponder; • The DVB Adapter or STB 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	The AXMEDIS professional user, the AXMEDIS Satellite Data Broadcast 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-ITV5
Steps	1 The User launches the 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 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 Client Application.
Expected results	The AXMEDIS Client (limited to basic functionalities) works fine: • The AXMEDIS Client starts/stops and behaves correctly; • All previously installed application still works 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 Satellite Data Broadcast Distributor for troubleshooting.
Test case Scope/Type	GUI, Backend / WhiteBox

16.1.4 User Registration

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

TCId	TC16.1.4
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.
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. The User can entirely access to the complete AXMEDIS Offer reserved to him.
Variations	None.
Issues	None.
Test case Scope/Type	GUI, Backend / WhiteBox

16.2 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 Selection

16.3.1 Manual Content Selection

TCId	TC16.3.1
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.

16.3.2 Automatic Content Selection

TCId	TC16.3.2
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

16.4 Content Reception

TCId	TC16.4
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

16.5 Content Reparation

TCId	TC16.5
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

16.6 Content Access

TCId	TC16.6
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.

16.7 Content Preview

TCId	TC16.7
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

16.8 License Acquisition

TCId	TC16.8
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

16.8.1 User Identification

TCId	TC16.8.1
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

16.8.2 Billing

TCId	TC16.8.2
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

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 or any other specific administrative operation)

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	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Kiosk Factory Application Front End • Catalog Management Module • Query support • Local DB
Description of functionality to be tested	The kiosk Manager creates a catalogue at the Kiosk Factory starting from available sources (internal and external) fixing availability dates, available rights (depending on object origin, acquired rights and desired usable by target consumers) and prices
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, is a registered AXMEDIS user with a specific UID, the operation is performed in the kiosk factory.
Data set used	AXDS-MCTestUser, AXDS-Kiosk1, AXDS-Kiosk2
Steps	1 The kiosk manager initiates catalogue preparation procedure (@kiosk factory) 2 The kiosk manager inserts the following data in the kiosk factory catalogue creation user interface: 2.1 Catalogue file name (mandatory)

	2.2 Catalogue identifier (mandatory) 2.3 Catalogue description 2.4 Catalogue template 3 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 4 The query support system returns a list of AXOID and related metadata 5 The Kiosk manager browse the list and identifies the needed objects accessing to public metadata and preview samples stored in AXINFO for each AXOID of the received list. 6 The kiosk manager selects the desired objects to be included into the catalogue 7 The kiosk manager adds additional information on a per item basis, for the time being it is expected to operate by adding allowed grants and ranking orders: 7.1 Selecting ranking for the Top 10 category 7.2 Selecting ranking for the Best pick category 7.3 Selecting ranking for the Offer category These operation are repeated up to catalogue completion, it is expected to perform it automatically in a second phase exploiting analysis of statistical reports on content/object usage 8 The kiosk manager adds required information on allowed grants specifying: 8.1 Rights available for purchase by the end user 8.2 Validity of rights (from – to) 8.3 Additional specification on allows use 8.4 Country and Region of validity for the grant 8.5 Fee type, amount and currency 8.6 Back account for the fee reception 9 These operation are presently manually repeated up to catalogue completion, it is expected to perform it automatically in a second phase
Expected results	The kiosk manager is able to select & retrieve needed objects, generate and store a kiosk catalogue
Variations	None
Issues	None
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	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Kiosk Factory Application Front End • Catalog Management Module • Query support • Local DB
Description of functionality to be tested	The kiosk manager is able to distribute a specified kiosk catalogue to all kiosks or to a selected subset
Partners, people	The kiosk manager, is a registered AXMEDIS user with a specific UID and has all

involved	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-MCTestUser, AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk9
Steps	1 The kiosk manager initiates catalogue sending procedure (@kiosk factory) 2 The kiosk manager inserts the following data in the kiosk factory catalogue transmission user interface: 2.a Catalogue file name (mandatory) 2.b Catalogue target domain (mandatory) 3 The kiosk manager activates the sending procedure
Expected results	The specified catalogue is successfully distributed to selected kiosks
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.3 Content Catalogue Loading

TCId	TC17.3
Test case	Content Catalogue Loading Update
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The user has administrative rights and is able to operate with the specified tools and supporting components
Configuration description	• Kiosk Application Front End • Catalog Management Module • Local DB
Description of functionality to be tested	The kiosk manager is able to receive and load a specified kiosk catalogue in a selected kiosk
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-MCTestUser, AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk9
Steps	1 The kiosk catalogue management module checks current catalogue validity and finds that a change is needed (either for reception of a newer one or for validity expiration) 2 The kiosk catalogue management module performs a “switch to maintenance mode” 3 The kiosk system exits normal operational and enters in maintenance mode. 4 The kiosk catalogue management module Loads the new catalogue 5 The kiosk catalogue management module performs a “switch to normal mode”
Expected results	The specified kiosk catalogue is successfully loaded and put in operation in a selected kiosk
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend /BlackBox

17.4 User registration to kiosk

TCId	TC17.4
Test case	User registration to kiosk
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • User Management module • Data Management module • Local DB • AXCS Connection module • AXCS
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-MCTestUser, AXDS-Kiosk5, AXDS-Kiosk13
Steps	1 The system is fully functional; the user is at a POP that is in idle mode and currently displays the login page. 2 Being not registered the user presses the “register” button on the login page. 3 The application front-end presents the user a registration form with the following data: o Login ID (mandatory) o Password (mandatory) o First Name (mandatory) o Last Name (mandatory) o e-mail (mandatory) o Birth date o Telephone o Mobile phone o VAT o Address (base on the following fields): ▪ ▪ state, town, street, number and post-code o Preferred payment method: ▪ pre-paid-cards, credit card… o Preferred device o Notes 4 The user provides the required data and confirms. 5 The kiosk local application server properly formats the data and send a request to the AXMEDIS Registration Service 6 In case of success the AXMEDIS Registration Service sends back to the kiosk user final UID 7 The kiosks retrieves the registration clearance, stores provided UID and sends the confirmation e-mail to the user specified account and grants user access to the kiosk application and services.
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	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

17.5 User Login

TCId	TC17.5
Test case	User Login
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • User Management module • Data Management module • Local DB
Description of functionality to be tested	• The kiosk authentication application
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-MCTestUser, AXDS-Kiosk5, AXDS-Kiosk13
Steps	1 The user interacts with the application front-end inserting the requested login data (user ID and password) and confirming 2 Filled in data structure is sent back to the kiosk user management 3 The kiosk user management checks user information locally 4 The kiosk user management sends user data to the AXCS for verification (via AXCS web service interface) 5 The AXCS checks received info 6 The AXCS logs the registration event 7 The AXCS sends back to the kiosk user management a ACK 8 The kiosk user management confirms the login to the application front end 9 The application front end grants access to available services: application front-end presents the user a page allowing to: 9.a Browse the catalogue 9.b Modify own data 9.c View or revise the current chart 9.d Browse acquired content 9.e Change GUI language 9.f Catalogue management (available only for administrative users) 9.g User management (available only for administrative users) 9.h Logout
Expected results	The user should be logged into the system
Variations	None

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

17.6 Content Browsing & previewing

TCId	TC17.6
Test case	Content Browsing & Previewing
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • AXMEDIS Client Player
Description of functionality to be tested	The user is intending to use the catalogue, browse its content, select and preview selected items
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-MCTestUser, 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.a Brief description for text 5.b Thumbnail for images 5.c X sec sample for Audio (X will depend on IPR rules) 5.d X sec sample for Video (X will depend on IPR rules) 5.e X sec sample for Animations (X will depend on IPR rules) 5.f X sec sample for Multimedia (X will depend on IPR rules) 6 The end users decides next step between: 6.a Activate acquiring procedure 6.b Returning to browsing
Expected results	The user is able to use the catalogue, browse its content, select and preview selected items
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.7 Content Selection and Chart Management

TCId	TC17.7
Test case	Content Selection And Chart Management
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational

Configuration description	• Kiosk Application Front End • Catalogue Management module • Data Management module • Local DB
Description of functionality to be tested	The user is intending to add to own chart selected content for a specific usage
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-MCTestUser, AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The end user selects a specific content for addition to the chart 2 The user requests to proceed either to check out or to continue browsing 3 Depending on previous step results the system enters one of the following to states: 3.a Check out procedure activation 3.b Browsing & previewing mode
Expected results	The user is able to add to own chart selected content fro the specified usage
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

17.8 Check out procedure initiation

TCId	TC17.8
Test case	Check Out Procedure Initiation
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • User Management module • Data Management module • Kiosk ECommerce module
Description of functionality to be tested	Based on user request the system should start the check-out procedure to grant content acquisition
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-MCTestUser, 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

17.9 Purchasing / Acquiring / Renting

TCId	TC17.9
Test case	Purchasing / Acquiring / Renting
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • User Management module • Data Management module • Kiosk ECommerce module • Local DB • Domain PMS
Description of functionality to be tested	Based on previously performed steps the system should start the finalisation steps of the check-out procedure to grant content acquisition
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-MCTestUser, 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

17.10 Repository Selection

TCId	TC17.10
Test case	Repository Selection
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Data Management module • Kiosk ECommerce module • Local DB • Query support • Domain PMS
Description of functionality to be tested	The system should identify the location of the content to be delivered to the end user and get it to ensure availability during delivery
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-MCTestUser, 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 and data cashed locally
Expected results	The selected object is located, retrieved and locally cashed for subsequent delivery (in case of remotely available content, local cashed content will have to be removed)
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

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

TCId	TC17.11
Test case	Destination Target Identification (Unique Id For Target – WiFi)
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Data Management module • Kiosk Delivery module

	• Local DB
Description of functionality to be tested	Target device should be identified with no ambiguity so to ensure respect of possible constraints in content fruition connected to device unique ID
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	None
Issues	None
Test case Scope/Type	Backend / BlackBox

17.12 Delivery Template Selection (Depending on Device)

TCId	TC17.12
Test case	Delivery Template Selection (Depending On Device)
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Data Management module • Kiosk Delivery module • Local DB
Description of functionality to be tested	Depending on user device identification the proper content template should be identified and selected
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-MCTestUser, 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 system selects the template to be used for delivery
Expected results	Based on user device identification the proper content template is identified and selected
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

17.13 Delivery Format Selection (Depending on content)

TCId	TC17.13
Test case	Delivery Format Selection (Depending On Content)

Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Data Management module • Kiosk Delivery module • Local DB
Description of functionality to be tested	Depending on user device identification the proper content format should be identified and selected
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-MCTestUser, 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	Based on user device identification the proper content format is identified and selected
Variations	None
Issues	None
Test case Scope/Type	Backend / BlackBox

17.14 Billing

TCId	TC17.14
Test case	Billing
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Data Management module • Kiosk ECommerce module • Local DB • AXCS
Description of functionality to be tested	The application should prepare and send the billing info to the end user for documenting the positively carried out acquisition process
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-MCTestUser, 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 application has properly handled the economic transaction, corresponding billing info is produced and given both to the end user and to the AXMEDIS Certifier & Supervisor
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.15 Data Delivery

TCId	TC17.15
Test case	Data Delivery
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational • The billing phase is closed positively. The user device is wireless connected and the front-end application is properly running on it.
Configuration description	• Data Management module • Kiosk Delivery module • Local DB
Description of functionality to be tested	Selected content (along with related info, license…) should be successfully delivered to the end-user device
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-MCTestUser, 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	Selected content (along with related info, license…)is successfully delivered to the end-user device
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend /BlackBox

17.16 Check out procedure closure

TCId	TC17.16
Test case	Check Out Procedure Closure
Initial conditions	• All components are active and properly functioning

	• Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • Kiosk ECommerce module
Description of functionality to be tested	The system should complete the check-out & delivery stage and should return to normal browse mode
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-MCTestUser, 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
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

17.17 Successful Delivery Check (Recovery in Case of Failure)

TCId	TC17.17
Test case	Successful Delivery Check (Recovery In Case Of Failure)
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • Kiosk Delivery module • AXCS
Description of functionality to be tested	The delivery phase should end positively and selected content should be available onto end-user device
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-MCTestUser, AXDS-Kiosk1, AXDS-Kiosk2, AXDS-Kiosk3, AXDS-Kiosk12
Steps	1 The local system should monitor the download process to ensure a smooth delivery 2 The kiosk delivery module identifies the target device 3 Depending on target device the Kiosk delivery module acts as follows:

	3.a The target device is a terminal (POP) 3.a.1 The Kiosk delivery module adapts the content to the fruition device (if necessary) 3.a.2 The Kiosk delivery module returns to the application front end the info needed to retrieve the locally cached AXMEDIS object(s) 3.a.3 The application front end loads a page to confirm delivery and grant access to the AXMEDIS object(s) 3.b The target device is a user PDA 3.b.1 Adapts the content to the fruition device (if necessary) 3.b.2 The Kiosk delivery module retrieves device data (kind, storage, certificate...) 3.b.3 The Kiosk delivery module performs required check on received device data 3.b.4 If checks are positive the Kiosk delivery module loads a page to ask downlad activation 3.b.5 The user activates the download (a positive result to previous step is assumed here and the user should be free to decide the local storage position on the PDA) 3.b.6 The kiosk delivery module takes the cached content from the local storage 3.b.7 The kiosk delivery module retrieves from the local storage the kind of operation requested on the AXMEDIS object. If the requested operation is a purchase acts as follows: 3.b.7.1 If the AXMEDIS object is a NOT governed one the kiosk delivery module requires the Local License generator to generate a “device based” license 3.b.7.2 The Local License generator generates a “device based” license 3.b.7.3 The Local License generator returns the kiosk delivery module the generated “device based” license 3.b.7.4 Kiosk delivery module requires the AXCS to generate the due keys 3.b.7.5 The AXCS retrieves the due keys 3.b.7.6 The AXCS returns the kiosk delivery module the retrieved due keys 3.b.8 The kiosk delivery module loads data onto the PDA (AXMEDIS object, keys and license for not governed objects) 3.b.9 The kiosk delivery module monitors the download 4 The kiosk delivery module notifies the Kiosk Application front end of successful closure of the check out procedure 5 The user can now use the content according to acquired rights via the AXMEDIS viewer, while in case of problems the system should p erform at lest 3 retries 6 Inform the customer of the incurred problem 7 Ask the customer which choice is preferred among: 7.a New set of delivery retry 7.b Deferred delivery 7.c Delivery cancel 8 The system should take note of customer decision and consequently proceed to: 8.a Activate a new set of delivery retry (maximum 3) 8.b Deferred delivery

	8.b.1 Ask the customer the time of next delivery 8.b.2 Schedule next delivery 8.b.3 Flag the process for possible cancellation & refund 8.c Delivery cancel 8.c.1 Enter secure mode 8.c.2 Establish a secure connection with the AXMEDIS certification authority 8.c.3 Performs a roll back request (including billing cancelling and money refund) 8.c.4 The system notifies the customer that the delivery and related transaction has been annulated 8.c.5 The system notifies the customer that refund procedure has been activated 8.c.6 The secure connection with the certification authority is released 8.c.7 The system exits protected mode 9 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

17.18 Content Fruition after Download on PDA or mobile

TCId	TC17.18
Test case	Content fruition after download on PDA
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • Kiosk Delivery module • AXCS • Domain PMS
Description of functionality to be tested	The acquired and delivered content should be used on the end-user device respecting DRM rules imposed by the set of acquired grants
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-MCTestUser, 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 PMS domain (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.a Sends back to the requesting PMS domain needed user keys (in case of positive result) 10.b Sends back to the requesting PMS domain a NACK 11 The PMS domain receives the reply and forwards it to the requesting viewer (via AXOM) 12 Depending on check results the viewer proceeds as follows: 12.a Allows content fruition 12.b Blocks content fruition
Expected results	The end user can use the content only according to the agreed license model and DRM fruition rules, no errors are reported
Variations	None
Issues	None
Test case Scope/Type	GUI, Backend /BlackBox

17.19 User Profile Change

TCId	TC17.19
Test case	User Profile Change
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • User Management module • Data Management module • Local DB • AXCS Connection module • AXCS
Description of functionality to be tested	The user wants to change/update data provided at registration time
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-MCTestUser, AXDS-Kiosk5, AXDS-Kiosk13
Steps	1 The application front end has granted access to available services including: 1.a Browse the catalogue 1.b Modify own data 1.c View support information 1.d 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: o Login ID (mandatory) o Password (mandatory) o First Name (mandatory)

	o Last Name (mandatory) o e-mail (mandatory) o Birth date o Telephone o Mobile phone o VAT o Address (base on the following fields): ▪ ▪ state, town, street, number and post-code o Preferred payment method: ▪ pre-paid-cards, credit card… o Preferred device o Notes 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 new data is now stored in the local system and updated in all other relevant places of the AXMEDIS infrastructure
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend, Backend / BlackBox

17.20 User device configuration & application front-end installation

TCId	TC17.20
Test case	User device configuration & application front-end installation
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Kiosk Application Front End • User Management module • Data Management module • Local DB
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-MCTestUser, AXDS-Kiosk10
Steps	1 The user has access to a page with the following info: 1.a How to connect the PDA / Tablet to the kiosk via WiFi (including how to test the connection) 1.b 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
Expected results	The end user device is properly configured and operational
Variations	None
Issues	None
Test case Scope/Type	GUI, Frontend / BlackBox

17.21 Content Update (Via Satellite)

TCId	TC17.21
Test case	Content Update (via Satellite)
Initial conditions	• All components are active and properly functioning • Connection with remote systems necessary to operation fulfillment are in place and operational
Configuration description	• Data Management module • Local DB • AXMEDIS Action Manager • AXMEDIS Loader & Saver
Description of functionality to be tested	The kiosk content (regardless of its type) should be updated thanks to remotely sent data using the satellite down-link
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-MCTestUser, 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.a Data is not available yet so a further check is scheduled and the application enters wait mode (cycling back to point 1) 2.b Data is available therefore is downloaded (2.b.1) and progressively cached locally (2.b.2) 2.c 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.a Performs consistency check on received data 5.b If result is positive returns ACK and control to the AXMEDIS Action Manager 5.c If result is negative requires the distribution server to resend the damaged packages via Up-link as detailed here after: 5.c.1 Satellite Reception Content Checker requires missing or damaged packages via Up-Link 5.c.2 Satellite Reception Content Checker receives missing or damaged packages via Up-Link 5.c.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.a Received data are AXMEDIS Object: data is stored in the AXDB 8.b Received data are system / application updates: invoke the kiosk data manager to store data locally according to needs 8.b.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 Composite Test Case: Automatic content production

This test case describes main steps to test the production and put in execution of content processing rules. The fist tool to be used it the AXCP Rule Editor. It is used to edit a new or an existing AXCP Rule by defining properties in terms of firing conditions when it is active in the AXCP Rule Engine, parameters involved in the script, dependencies related to AXCP Tools to be used, the source code of script written using the AXCP Rule Language. If the production is dependent by AXMEDIS Worklow request, a notification is produced and sent back to communicate the end of the activity. When a rule is activated, it is transferred into the AXCP Rule Engine. The AXCP Rule Scheduler periodically tests the firing condition and when it is verified the rule is put in execution on a AXCP Rule Executor.

Who (producer, integrator, distributor, author, etc.)	**Action Performed (create, assign id, certify, authenticate, register, make query, open, access, load, save, extract, copy, play, move, send, etc.) REFER TO TEST CASES DEFINED IN TH EPREVIOUS PAGES**	**What is produced (resource, object, metadata, license, protection information, etc..)**	**Description of Content**	**AXMEDIS Tool Name or NAXT (not with AXMEDIS tool)**
Producer	Create a new AXCP Rule	A new empty AXCP Rule		AXCP Rule Editor, AXMEDIS Workflow manager
Producer	Load an AXCP Rule	An AXCP rule		AXCP Rule Editor
Producer	Edit the rule	Schedule, the script of rule, rule parameters and dependencies		AXCP Rule Editor
Producer	Debug the script	Corrections, simulation, analysis		AXCP Rule Editor
Producer	Activate the AXCP Rule	A feedback notification from the AXCP Rule Engine		AXCP Rule Editor, AXCP Rule Engine
Producer	Send completion of activity	A workflow notification		AXCP Rule Editor, AXMEDIS Workflow manager
AXCP Rule Engine	Run an AXCP Rule	Axmedis Objects, licnese, protection, adaptation, formatting		AXCP Rule Scheduler and Executor, AXMEDIS AXCP Tools (plugins), Main Query Support, AXDB
AXCP Rule	Send completion of activity	A workflow		AXCP Rule

Engine		notification		Scheduler and Executor, AXMEDIS Workflow manager

19 Composite Test Case: Content Protection and Governance

Who (producer, integrator, distributor, author, etc.)	**Action Performed (create, assign id, certify, authenticate, register, make query, open, access, load, save, extract, copy, play, move, send, etc.) REFER TO TEST CASES DEFINED IN THE PREVIOUS PAGES**	**What is produced (resource, object, metadata, license, protection information, etc..)**	**Description of Content**	**AXMEDIS Tool Name or NAXT (not with AXMEDIS tool)**
Creator or Distributor	Content Protection (see TC "Protection of an AXMEDIS object" or "Registration of a protected object")	Protection Information		Protection Editor
Creator or Distributor	Generate the license (see TC "License creation for new content")	AXMEDIS license	Domain license	PMS Server

20 Composite Test Case: Content Acquisition for the domain and usage in the domain

Who (producer, integrator, distributor, author, etc.)	**Action Performed (create, assign id, certify, authenticate, register, make query, open, access, load, save, extract, copy, play, move, send, etc.) REFER TO TEST CASES DEFINED IN THE PREVIOUS PAGES**	**What is produced (resource, object, metadata, license, protection information, etc..)**	**Description of Content**	**AXMEDIS Tool Name or NAXT (not with AXMEDIS tool)**
User	Purchase the license			
Distributor	Generate the license (see TC "License creation for new content")	AXMEDIS license	Domain license	PMS Server
User	Download license and/or content			
User	Register in a domain (test cases to be defined)	Info updated in domain manager	User belongs to a domain	AXMEDIS Domain Manager
User	Certify an AXMEDIS tool (first use of the tool) (see TC "Certification of AXMEDIS tool by a user on a Device")	Tool certificate, private key, enabling code, AXTID, AXCS database entry	After certifying AXMEDIS tool is ready to be used in the system.	AXMEDIS Tool
User	Perform an action over the resource (see TC "User authorisation based on licenses")	Authorisation result. Storage of protection info in secure cache if positive authorisation		PMS Client PMS Server PMS Domain Home / Factory

21 Composite Test Case: Content Production and Usage (Kiosk & Mobile)

Who (producer, integrator, distributor, author, etc.)	Action Performed (create, assign id, certify, authenticate, register, make query, open, access, load, save, extract, copy, play, move, send, etc.) REFER TO USE CASES DEFINED IN TH EPREVIOUS PAGES	What is produced (resource, object, metadata, license, protection information, etc..)	Description of Content	AXMEDIS Tool Name or NAXT (not with AXMEDIS tool)
Producer	Creates an object	AXMEDIS Object	HTML page (image & text)	AXMEDIS Editor
Producer	Protects an object and issues a license (TC14.3.1 and TC13.5.2.x)	License	License Protection info	AXMEDIS License editor Protection tool
Producer	Publishes an object for distribution	AXMEDIS protected Object	HTML page License Protection info	AXMEDIS P&P / AXEXPTool
Distributor	Retrieves an object	AXMEDIS protected Object	HTML page License Protection info	AXMEDIS Query Support
Distributor	Finalises distribution - adding pricing, use restrictions… (TC17.2-8, TC19.1)	AXMEDIS protected Object	HTML page License Protection info	AXMEDIS License editor Protection tool AXCP
Distributor	Distributes an object via kiosk (TC19.1-3/19.21)	AXMEDIS protected Object	HTML page License Protection info	Kiosk application AXMEDIS Framework
Distributor	Distributes an object via mobile (TC17.2.1-12)	AXMEDIS protected Object	HTML page License Protection info	Mobile application AXMEDIS Framework
User	Selects and acquires an object via kiosk (TC19.4-17)	AXMEDIS protected Object	HTML page License Protection info	Kiosk application AXMEDIS Framework
User	Uses the acquired object via kiosk (TC19.18)	AXMEDIS protected Object	HTML page License Protection info	Kiosk application AXMEDIS Framework
User	Selects and acquires an object via mobile (TC17.2.13-23)	AXMEDIS protected Object	HTML page License Protection info	Mobile application AXMEDIS Framework
User	Uses the acquired object via mobile (TC17.2.24)	AXMEDIS protected Object	HTML page License Protection info	Mobile application AXMEDIS Framework

22 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.

22.1 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.

22.2 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.

22.3 AXDS-DB3

A set of at AXMEDIS objects with at least one HTML resource referencing other resources (images, stylesheets) within or without the object.

22.4 AXDS-Editor1

An AXMEDIS Object.

22.5 AXDS-Editor2

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

22.6 AXDS-Editor3

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

22.7 AXDS-Editor4

An AXMEDIS object containing at least one element.

22.8 AXDS-Editor5

Composite AXMEDIS object.

22.9 AXDS-Editor6

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

22.10 AXDS-IVE1

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

22.11 AXDS-IVE2

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

22.12 AXDS-Composition1

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

22.13 AXDS-Composition2

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

22.14 AXDS-Composition3

Collection of Composition rules.

22.15 AXDS-Composition4

Collection of active Composition rules.

22.16 AXDS-Formatting1

Formatting rules set to active.

22.17 AXDS-Formatting2

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

22.18 AXDS-Formatting3

Collection of formatting rules

22.19 AXDS-Formatting4

Collection of active formatting rules.

22.20 AXDS-Workflow1

An NPD process containing at least one component

22.21 AXDS-Workflow2

An NPD process containing at least two components

22.22 AXDS-Workflow3

An NPD process containing at least one component plus CPA data

22.23 AXDS-Workflow4

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

22.24 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.

22.25 AXDS-AXEPPR

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

22.26 AXDS-AXEPLR

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

22.27 AXDS-AXEPAS

An active selection of AXMEDIS objects.

22.28 AXDS-AXEPP2Pheaders

The headers involved in the P2P handshaking.

22.29 AXDS-AXEPQH

A set of query Hits.

22.30 AXDS-P&P1

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

22.31 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.

22.32 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.

22.33 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.

22.34 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.

22.35 AXDS-ITV1

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

22.36 AXDS-ITV2

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

22.37 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.

22.38 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.

22.39 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.

22.40 AXDS-ITVlogin

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

22.41 AXDS-ITVloginB

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

22.42 AXDS-ITVpreferences

Set of Authorizations and Filters associated to a test login.

22.43 AXDS-ITVobjects

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

22.44 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.

22.45 AXDS-ITVstations

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

22.46 AXDS-ITVschedule

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

22.47 AXDS-ITVcredentials

List of credentials.

22.48 AXDS-ITVlicences

Set of licencesID.

22.49 AXDS-ITVpayments

Accepted form of payment to acquire licences.

22.50 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).

22.51 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).

22.52 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.

22.53 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.

22.54 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.

22.55 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.

22.56 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).

22.57 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.

22.58 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).

22.59 AXDS-AXCS10

It includes a group of at least 30 records per table, for ALL tables of ALL databases managed by AXCS. It can be considered a global set of data for AXCS. All included data must be consistent.

22.60 AXDS-AXCS11

It includes a group of at least 30 records per table, for ALL tables of ALL databases managed by AXCS. It can be considered a global set of data for AXCS. 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-AXCS10.

22.61 AXDS-OIDGen1

A set of 20 ObjectID’s and associated information.

22.62 AXDS-Supervisor1

A group of 20 user identifiers (AXUID, AXCID, AXDID, AXCSID and AXTPID).

22.63 AXDS-Supervisor2

A group of 20 certified tool identifiers (AXTID).

22.64 AXDS-Supervisor3

A group of 20 registered tool identifiers (AXRTID).

22.65 AXDS-Supervisor4

A group of 20 SupervisorInputData (AXRTID).

22.66 AXDS-CertVer1 (Full data set)

A group of at least 20 sets that include: user identifier (AXUID, AXCID, AXDID, AXCSID and AXTPID), Tool Identifier (AXTID), Tool fingerprint (hardware+software), Tool fingerprint digest, Tool operation history hash (lastFPPA) and operation history (set of Action Logs)..

22.67 AXDS-CertVer2 (Subset 1)

A group of at least 20 sets that include: user identifier (AXUID, AXCID, AXDID, AXCSID and AXTPID) and Tool fingerprint (hardware+software).

22.68 AXDS-CertVer3 (Subset 2)

A group of at least 20 sets that include: user identifier (AXUID, AXCID, AXDID, AXCSID and AXTPID), Tool Identifier (AXTID) and Tool fingerprint (hardware+software).

22.69 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.

22.70 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

22.71 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

22.72 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

22.73 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.

22.74 AXDS-DRMSupport5

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

22.75 AXDS-DRMSupport6

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

22.76 AXDS-DRMSupport7

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

* Optional fields

22.77 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

22.78 AXDS-PMS1

An AXMEDIS object, object version, protection stamp and protection information (including keys).

22.79 AXDS-PMS2

Protection information related to an AXMEDIS Object.

22.80 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.

22.81 AXDS-ENCDEC1

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

22.82 AXDS-PlMulti

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

22.83 AXDS-PlVid

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)

22.84 AXDS-PlAu

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).

22.85 AXDS-MCProject

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

22.86 AXDS-MCProducer

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

22.87 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.

22.88 AXDS-MCShop

The Media Club data definition for the shop.

22.89 AXDS-MCPayMethod

The Media Club data specifying all available payment methods.

22.90 AXDS-MCTransaction

The Media Club transaction data.

22.91 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

22.92 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.

22.93 AXDS-PCDist1

An AXMEDIS Protected Object.

22.94 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

22.95 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

22.96 AXDS-Kiosk3

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

22.97 AXDS-Kiosk4

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

22.98 AXDS-Kiosk5

AXMEDIS Certifier & Supervisor user management data

22.99 AXDS-Kiosk6

Licenses of the AXMEDIS governed object

22.100 AXDS-Kiosk7

Composition & formatting rules

22.101 AXDS-Kiosk8

DRM rules

22.102 AXDS-Kiosk9
Selected Content List for Kiosk

22.103 AXDS-Kiosk10
The kiosk applications

22.104 AXDS-Kiosk11
The kiosk procedures

22.105 AXDS-Kiosk12
The kiosk catalogue

22.106 AXDS-Kiosk13
The kiosk local user management data

22.107 AXDS-Mobile1
Categories List for Mobile Portal

22.108 AXDS-Mobile2
The Mobile Portal local user management data

22.109 AXDS-Mobile3
Composition & formatting rules

22.110 AXDS-Mobile4
DRM rules

22.111 AXDS-Mobile5
AXMEDIS Certifier & Supervisor user management data

22.112 AXDS-Mobile6
Licenses of the AXMEDIS object

22.113 AXDS-Mobile7
Device Profile data

22.114 AXDS-MozillaPlugin
HML pages for testing the JavaScript capabilities of the plugin.

22.115 AXDS-MozillaPlayer
Localization and skin files for the player