



Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE12.2.3.1

AXMEDIS-ELTEO specification

Version: 2.0

Date: 13-06-2007

Responsible: TEO (revised and approved by coordinator)

Project Number: IST-2-511299 Project Title: AXMEDIS Deliverable Type: report Visible to User Groups: yes Visible to Affiliated: yes Visible to the Public: YES contractually public
Deliverable Number: DE12.2.3.1 Contractual Date of Delivery: M30 Actual Date of Delivery: 13-06-2007 Title of Deliverable: DE12.2.3.1 AXMEDIS-ELTEO specification Work-Package contributing to the Deliverable: WP12.2.3 Task contributing to the Deliverable: WP12.2.3 Nature of the Deliverable: report Author(s): KTU, TEO, Elion
Abstract: this document specifies the detailed architecture of TEO AXMEDIS demonstrator for IPTV and provides use cases and scenarios. It also defines the specification of missing components to be developed by project partners. It involves AXMEDIS player for STB, components for middleware server and MPEGTSAadaptationPlugin. Keyword List: STB, AXMEDIS player for STB, P2P client, VOD server, DRM, Middleware, IPTV, RTSP, streaming.

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-fulfillment of any of his obligations in respect of this License.

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	7
1.1	LIST OF MODULES OR EXECUTABLE TOOLS SPECIFIED IN THIS DOCUMENT	9
1.2	LIST OF FORMATS SPECIFIED IN THIS DOCUMENT	9
1.3	LIST OF DATABASES SPECIFIED IN THIS DOCUMENT	9
1.4	LIST OF PROTOCOLS SPECIFIED IN THIS DOCUMENT	10
1.5	LIST OF TERMS USED IN THIS DOCUMENT	10
2	GENERAL USE CASES AND SCENARIOS	12
2.1	USE CASE MODEL DESCRIBING END USER ACTIONS	12
2.1.1	End User connects to TEO AXMEDIS network	13
2.1.2	End User browses TEO AXMEDIS content	13
2.1.3	End User views trailers	14
2.1.4	End User buys AXMEDIS content	15
2.1.5	End User stops playing AXMEDIS object video	16
2.1.6	End User pauses playing AXMEDIS object video	16
2.1.7	End User continues playing AXMEDIS object video	17
2.1.8	End User seeks AXMEDIS object video	18
2.1.9	End User selects to view his information	18
2.1.10	End User selects to view his usage information	19
2.1.11	End User selects to view his billing information	19
2.1.12	End User enters his PIN	20
2.1.13	End User chooses payment method	20
2.1.14	License generation	21
2.2	USE CASE MODEL DESCRIBING TEO VOD MANAGER ACTIONS	22
2.2.1	TEO VOD Manager connects to TEO Content & User Management Portal	22
2.2.2	TEO VOD Manager selects content management	23
2.2.3	TEO VOD Manager downloads content from global AXMEDIS network to local DB	23
2.2.4	TEO VOD Manager deletes content from local DB	24
2.2.5	TEO VOD Manager makes content report	25
2.3	USE CASE MODEL DESCRIBING TEO IPTV ADMINISTRATOR ACTIONS	26
2.3.1	TEO IPTV Administrator connects to TEO Content & User Management Portal	26
2.3.2	TEO IPTV Administrator selects user management	27
2.3.3	TEO IPTV Administrator adds new user	27
2.3.4	TEO IPTV Administrator deletes user	28
2.3.5	TEO IPTV Administrator modifies user rights	28
2.3.6	TEO IPTV Administrator makes user report	29
2.4	USE CASE MODEL DESCRIBING CONTENT PRODUCTION IN VRS FACTORY	30
2.4.1	Content adaptation for streaming to IPTV STB	30
2.4.2	Producer creates AXMEDIS object for streaming to IPTV STB	31
2.4.3	VRS Producer creates AXMEDIS object content for PC	31
2.5	GENERAL USE CASES AND SCENARIOS OF ELION PC DEMONSTRATOR (ELION)	33
2.5.1	End user use cases	33
2.5.2	Manager use cases	36
3	GENERAL ARCHITECTURE AND RELATIONSHIPS AMONG THE SPECIFIED MODULES	38
3.1	GENERAL ARCHITECTURE OF VIDEO ON DEMAND DISTRIBUTION VIA IPTV STB (TEO)	38
3.2	GENERAL ARCHITECTURE OF CONTENT PRODUCTION & PROCESSING DEMONSTRATOR (VRS)	43
3.3	END USER SCENARIOS (SEQUENCE DIAGRAMS)	46
3.3.1	End User connection to TEO network scenario (Sequence diagram)	47
3.3.2	End User browsing in TEO AXMEDIS DB scenario (Sequence diagram)	49
3.3.3	End User viewing TEO AXMEDIS content trailers scenario (Sequence diagram)	50
3.3.4	End User buys TEO AXMEDIS content scenario (Sequence diagram)	50
3.3.5	End User stopping TEO AXMEDIS content scenario (Sequence diagram)	52
3.3.6	End User pausing TEO AXMEDIS content scenario (Sequence diagram)	53
3.3.7	End User continue playing TEO AXMEDIS content scenario (Sequence diagram)	53
3.3.8	End User seeking TEO AXMEDIS content scenario (Sequence diagram)	54
3.3.9	End User viewing his information scenario (Sequence diagram)	55

3.3.10	End User viewing his usage information scenario (Sequence diagram).....	55
3.3.11	End User viewing his billing information scenario (Sequence diagram)	56
3.3.12	End User entering his PIN scenario (Sequence diagram)	57
3.3.13	End User choosing his payment method scenario (Sequence diagram)	58
3.4	TEO VOD MANAGER SCENARIOS (SEQUENCE DIAGRAMS)	59
3.4.1	TEO VOD Manager content management scenario (Sequence diagram).....	59
3.4.2	TEO VOD Manager downloading AXMEDIS content and placing into VOD Server and AXDB (Sequence diagram)	60
3.4.3	TEO VOD Manager content deleting scenario (Sequence diagram)	61
3.4.4	TEO VOD Manager content reporting scenario (Sequence diagram).....	62
3.4.5	TEO VOD Manager connecting scenario (Sequence diagram)	63
3.5	TEO IPTV ADMINISTRATOR SCENARIOS (SEQUENCE DIAGRAMS)	64
3.5.1	TEO IPTV Administrator user management scenario (Sequence diagram)	64
3.5.2	TEO IPTV Administrator user adding scenario (Sequence diagram)	64
3.5.3	TEO IPTV Administrator user deleting scenario (Sequence diagram)	65
3.5.4	TEO IPTV Administrator user rights modification scenario (Sequence diagram)	66
3.5.5	TEO IPTV Administrator user reporting scenario (Sequence diagram)	67
3.5.6	TEO IPTV Administrator connecting scenario (Sequence diagram)	67
3.6	IMPLEMENTATION DIAGRAMS	68
3.7	GENERAL ARCHITECTURE AND RELATIONSHIPS AMONG THE SPECIFIED MODULES (ELION)	70
3.7.1	General architecture of video on demand distribution for PC (ELION)	70
3.7.2	End user scenarios	71
3.7.3	Manager scenario	75
4	SPECIFICATION OF THE MODULES TO BE PRODUCED	76
4.1	MODULE AXSTB PLAYER (TEO).....	76
4.1.1	General Description of the Module.....	77
4.1.2	Module Design in terms of Classes.....	79
4.1.3	User interface description	80
4.1.4	Technical and Installation information	80
4.1.5	Draft User Manual	81
4.1.6	Examples of usage	81
4.1.7	Integration and compilation issues.....	81
4.1.8	Configuration Parameters	81
4.1.9	Errors reported and that may occur.....	81
4.1.10	Formal description of algorithm <.....>.....	82
4.2	MODULE CONTENT DOWNLOADER P2P CLIENT (TEO)	83
4.2.1	General Description of the Module.....	84
4.2.2	Module Design in terms of Classes.....	86
4.2.3	User interface description	87
4.2.4	Technical and Installation information	87
4.2.5	Draft User Manual	87
4.2.6	Examples of usage	87
4.2.7	Integration and compilation issues.....	87
4.2.8	Configuration Parameters	87
4.2.9	Errors reported and that may occur.....	87
4.2.10	Formal description of algorithm <.....>.....	87
4.3	MODULE VOD ACCESS SERVICE (TEO).....	88
4.3.1	General Description of the Module.....	89
4.3.2	Module Design in terms of Classes.....	89
4.3.3	User interface description	101
4.3.4	Technical and Installation information	101
4.3.5	Draft User Manual	101
4.3.6	Examples of usage	101
4.3.7	Integration and compilation issues.....	101
4.3.8	Configuration Parameters	101
4.3.9	Errors reported and that may occur.....	102
4.3.10	Formal description of algorithm <.....>.....	102
4.4	MODULE TEO CONTENT & USER MANAGEMENT PORTAL (TEO)	103
4.4.1	General Description of the Module.....	104

4.4.2	Module Design in terms of Classes.....	104
4.4.3	User interface description	105
4.4.4	Technical and Installation information	105
4.4.5	Draft User Manual	105
4.4.6	Examples of usage	105
4.4.7	Integration and compilation issues.....	106
4.4.8	Configuration Parameters	106
4.4.9	Errors reported and that may occur.....	106
4.4.10	Formal description of algorithm <.....>.....	106
4.5	PLUG-IN MPEGTSADAPTATIONPLUGIN (TEO)	107
4.5.1	General Description of the Module.....	108
4.5.2	Module Design in terms of Classes.....	109
4.5.3	User interface description	111
4.5.4	Technical and Installation information	111
4.5.5	Draft User Manual	111
4.5.6	Examples of usage	111
4.5.7	Integration and compilation issues.....	111
4.5.8	Configuration Parameters	111
4.5.9	Errors reported and that may occur.....	111
4.5.10	Formal description of algorithm <.....>.....	112
4.6	ELION PORTAL INTEGRATION MODULE ELAXAPI_WS	112
4.6.1	General description of the module	113
4.6.2	Module design in terms of classes	114
4.6.3	Module database design.....	114
4.6.4	Module WDSL specification	114
4.6.5	User interface description	117
4.6.6	Technical and installation information	117
4.6.7	Draft user manual.....	117
4.6.8	Example of usage.....	117
4.6.9	Integration and compilation issues.....	117
4.6.10	Configuration Parameters	117
4.6.11	Errors reported and may occur	117
4.6.12	Formal description of algorithm <.....>.....	118
5	BIBLIOGRAPHY	119
6	GLOSSARY	119

1 Executive Summary and Report Scope

This document represents the general architecture of ELTEO demonstrators (content production (preparation) for streaming (VRS), distribution for VOD in IPTV (via STB) (TEO) and distribution via internet to PC (Elion)), specification of components to be developed for the TEO AXMEDIS demonstrator in IPTV environment and use case in which these components will be utilised. These components are needed in order to a) enable the demonstration (replay) of AXMEDIS video objects on a TV set via STB (AXSTBPlayer) and b) integrate the AXMEDIS Framework components and tools into IPTV and VOD architecture. The document is a pre-requisite for further work to be carried out in TEO AXMEDIS IPTV demonstrator implementation – development, integration and demonstrator setup. Specifically, specifications for the following main components are provided:

- AXMEDIS Player for STB – AXSTB Player;
- Content downloader P2P client and VOD Access Service;
- TEO Content & User Management Portal and
- AXMEDIS DRM support tool for streaming media (MPEGTSAdaptationPlugin).

The full AXMEDIS specification document has been decomposed in the following parts:

DE number	Deliverable title	responsible
DE3.1.2.2.1	Specification of General Aspects of AXMEDIS framework, first update of DE3.1.2 part A AXMEDIS-DE3-1-2-2-1-Spec-of-AX-Gen-Asp-of-AXMEDIS-framework-upA-v1-0.doc	DSI
DE3.1.2.2.2	Specification of AXMEDIS Command Manager, first update of DE3.1.2 part B AXMEDIS- DE3-1-2-2-2-Spec-of-AX-Cmd-Man-upB-v1-0.doc	DSI
DE3.1.2.2.3	Specification of AXMEDIS Object Manager and Protection Processor, first update of DE3.1.2 part B AXMEDIS-DE3-1-2-2-3-Spec-of-AXOM-and-ProtProc-upB-v1-0.doc	DSI
DE3.1.2.2.4	Specification of AXMEDIS Editors and Viewers, first update of DE3.1.2 part B AXMEDIS-DE3-1-2-2-4-Spec-of-AX-Editors-and-Viewers-upB-v1-0.doc	DSI
DE3.1.2.2.5	Specification of External AXMEDIS Editors/Viewers and Players, first update of DE3.1.2 part B AXMEDIS-DE3-1-2-2-5-Spec-of-External-Editors-Viewers-Players-upB-v1-0.doc	EPFL
DE3.1.2.2.6	Specification of AXMEDIS Content Processing, first update of DE3.1.2 part C AXMEDIS-DE3-1-2-2-6-Spec-of-AX-Content-Processing-upC-v1-0.doc	DSI
DE3.1.2.2.7	Specification of AXMEDIS External Processing Algorithms AXMEDIS-DE3-1-2-2-7-Spec-of-AX-External-Processing-Algorithms-v1-0.doc	FHGIGD
DE3.1.2.2.8	Specification of AXMEDIS CMS Crawling Capabilities, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-8-Spec-of-AX-CMS-Crawling-Capab-v1-0.doc	DSI
DE3.1.2.2.9	Specification of AXMEDIS database and query support, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-9-Spec-of-AX-database-and-query-support-v1-0.doc	EXITECH
DE3.1.2.2.10	Specification of AXMEDIS P2P tools, AXEPTTool and AXMEDIS, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-10-Spec-of-AXEPTTool-and-AXMEDIS-tools-v1-0.doc	CRS4
DE3.1.2.2.11	Specification of AXMEDIS Programme and Publication tools, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-11-Spec-of-AX-Progr-and-Pub-tool-v1-0.doc	UNIVLEDS
DE3.1.2.2.12	Specification of AXMEDIS Workflow Tools, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-12-Spec-of-AX-Workflow-Tools-v1-0.doc	IRC

AXMEDIS ELTEO Specification

DE3.1.2.2.13	Specification of AXMEDIS Certifier and Supervisor and networks of AXCS, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-13-Spec-of-AXCS-and-networks-v1-0.doc	DSI
DE3.1.2.2.14	Specification of AXMEDIS Protection Support, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-14-Spec-of-AX-Protection-Support-v1-0.doc	FUPF
DE3.1.2.2.15	Specification of AXMEDIS accounting and reporting, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-15-Spec-of-AX-Accounting-and-Reporting-v1-0.doc	EXITEC H

1.1 List of Modules or Executable Tools Specified in this document

A module is a component that can be or is reused in other cases or points of the AXMEDIS Framework or of other AXMEDIS based solutions. The modules/tools have to include effective components and/or tools and also testing components and tools. Below is a list of key modules specified in this document:

Module/tool Name	Module/Tool Description and purpose, state also in which other AXMEDIS area is used	Standards exploited if any
AXSTB Player	The module is responsible for responding to End User actions. The components of this package will be installed into STB.	MPEG-2 TS, DVB-CSA
Content Downloader P2P Client	Component for placing downloaded AXMEDIS object to VOD server and updating AXMEDIS object metadata and placing metadata and stream information into AXDB.	
VOD Access Service	VOD Access Service module is responsible for proper End User AXSTB Player connection with AXMEDIS content. This service will act in MW server.	
TEO Content & User Management Portal	This package is responsible for TEO AXMEDIS local content and user management and is part of MW server. TEO Content & User Management Portal will be used by TEO VOD Manager.	
DRM support tool for streaming: MPEGTSAdaptationPlugin	Takes from AXMEDIS object mpeg stream content through AXMPEGTSRESOURCE_IN parameter and: <ol style="list-style-type: none"> 1. encodes mpeg stream and returns by AXMPEGTSRESOURCE_OUT parameter; 2. generates MPEG-TS streaming information, where is descrambler control words, and returns like resource; by AXMPEGSTREAMINFO_OUT parameter. MPEGTSAdaptationPlugin will be implemented in VRS Factory or used directly by AXEDITOR.	MPEG-2 TS, DVB-CSA

1.2 List of Formats Specified in this document

A format can be (i) an XML content file for modeling some information, (ii) a file format for storing information, (iii) a format that is manipulated by the tools described in this document, etc...

Format Name	Format Description and purpose, state also in which other modules is used	Standards exploited if any
AxMPEGStreamInfo	Information about scrambled MPEG-2 TS elementary streams and DVB-CSA CWs	MPEG-2 TS, DVB-CSA

1.3 List of Databases Specified in this document

This section is not applicable

Database Name	Database description and purpose, state also in which other AXMEDIS area is using	Standards exploited if any

1.4 List of Protocols Specified in this document

A protocol is a communication modality among distinct processes that can be located or not on different computers.

Protocol Name	protocol Description and purpose, state also in which other modules is used	Who is the master and who is the slave	Standards exploited if any
AXSTB streaming	To stream video content from VOD server to STB client	Master : VOD Server (Head End) Slave : TEO AXSTB Player (client)	MPEG-2 TS, DVB-CSA

1.5 List of Terms used in this document

Term	Description
AXDB	Database to be developed specifically for the demonstrator, for storing AXMEDIS object metadata and streaming information (AxStreamResource). The MPEG-2 TS resources are excluded due to storage efficiency. This DB is based on AXMEDIS database, specified in DE3.1.2.2.9 “Specification of AXMEDIS database and query support, first update of part of DE3.1.2”
AXEPTool B2B	Tool for connecting Middleware server with AXMEDIS network and downloading AXMEDIS objects. This tool will be used according to specifications described in DE3.1.2.2.10 “Specification of AXMEDIS P2P tools, AXEPTool and AXMEDIS, first update of part of DE3.1.2”
TEO Content & User Management Portal	This module will be developed on IPTV Content Management Portal platform and used for TEO AXMEDIS content and user management. This module will be used by TEO VOD Manager and TEO IPTV Administrator.
AXMEDIS VOD Portal	AXMEDIS VOD PORTAL, to developed on TEO Middleware platform. This portal will be used in TEO demonstrator to present AXMEDIS content to End User. AXSTB Player will use this portal to retrieve information about content and usage information.
AXSTB Player	AXMEDIS PC Player, described in DE3.1.2.2.5 “Specification of External AXMEDIS Editors/Viewers and Players, first update of DE3.1.2 part B”, which is ported into Kreatel STB and used by End User.
Billing DB	Existing database, where TEO users billing information is stored.
Content downloader P2P	Component for placing downloaded AXMEDIS object to VOD server, updating AXMEDIS object metadata and placing metadata and stream information into AXDB.
Data integration module (DIM)	TEO solution for integration different data formats in databases.
End User	User of IPTV services having STB with AXMEDIS STB

	Player installed.
Gala TV	Existing TEO IPTV service (www.galatv.lt).
Headend server	IPTV content distribution and broadcasting server, used for transferring MPEG-2 designated content via TCP/IP network.
iAccessService	VOD Access Service interface.
iAXSTBPlayer	AXSTB Player interface.
iContentManagement	Content Management module interface.
IPTV DB	TEO existing Database for storing clients, services and accounting information.
iManagement	TEO Content & User Management Portal interface
iUserManagement	User Management module interface.
iVODPortal	VOD Portal interface.
MPEGTSAdaptationPlugin	Plug-in for scrambling MPEG-2 Transport Stream (TS) resource using DVB-CSA algorithm so that content is ready for streaming over IP and protected and collecting information about all scrambled elementary MPEG-2 TS streams (ES).
TEO AXMEDIS system	TEO AXMEDIS demonstrator system.
TEO IPTV Administrator	TEO AXMEDIS system user who administers the system, implements introduction of new services, and performs user administration, reporting and billing.
TEO VOD Manager	TEO AXMEDIS system user who manages VOD Server content.
TV Portal	Existing Gala TV Portal.
VOD Access Service	Component for proper End User AXSTB Player connection with user DB's and services. VOD Access Service communicates with VOD server, Billing DB, IPTV DB, AXDB and AXMEDIS PMS to present required information to End User and react to End User actions.
VOD Portal	Portal for presenting VOD server content for End User.
VOD server	Experimental VOD content storage server, to be set up for AXMEDIS demonstration.

2 General Use Cases and scenarios

There are three actors in TEO AXMEDIS system: End User, TEO IPTV Administrator and TEO VOD Manager. In VRS Factory is one actor – VRS Producer.

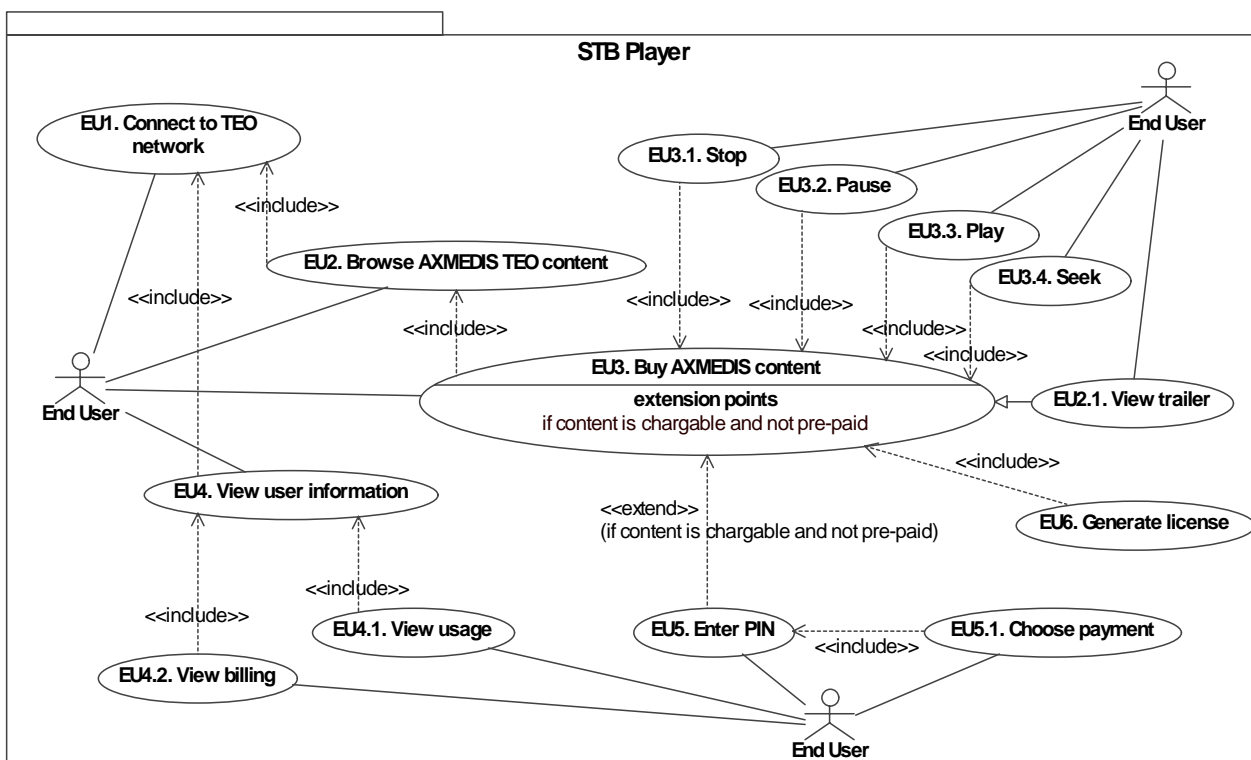
End User of AXSTB Player for IPTV network is one of the most important actors. End User uses STB with AXSTB Player and can connect to IPTV network for watching AXMEDIS video content. End User interface allows browsing through local AXDB and playing selected objects. If selected object is chargeable, End User is prompted to enter his PIN and choose the payment method. End User interface also allows browsing through global AXMEDIS network and viewing object descriptions. End User can select objects in global AXMEDIS network and subscribe them, but this feature will not be implemented in TEO demonstrator. End User can view his usage and billing information by using his AXSTB Player.

Another important actor in TEO IPTV network is TEO VOD Manager, which is responsible for content management in TEO AXMEDIS IPTV network. This actor downloads AXMEDIS objects from global AXMEDIS network and prepares them for streaming in VOD server. TEO VOD Manager downloads objects according to End Users requests or to the business plan.

TEO IPTV Administrator is responsible for user management. He can add, delete, modify user.

VRS Producer is responsible for content production for TEO and Elion demonstrators. For TEO demonstrator VRS Producer creates streaming video AXMEDIS objects, for Elion demonstrator – AXMEDIS objects for PC platform.

2.1 Use Case model describing End User actions



In this Use Case model End Users actions are described. End User can:

- connect to TEO IPTV network;
- browse and see metadata of AXMEDIS video content (if video object has a trailer, user can watch it for free);
- buy and play AXMEDIS video content (before object playing license must be generated (EU6.))):
 - o enter PIN;
 - o choose payment method;
 - o stop playing video;
 - o pause video;
 - o continue playing;
 - o seek to desired point of video;
- view user information:
 - o view usage information;
 - o view billing information.

In sections 2.1.1 – 2.1.14 these use cases are described more detail.

2.1.1 End User connects to TEO AXMEDIS network

UCId	EU1.
Use case	Connect to TEO network
Description	User connects to TEO IPTV network
Actors	End User
Assumptions	End User has STB, which is connected to IPTV network through the DSL network. Each STB has ID, which is used for user identification.
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends connect request to VOD Portal (iVODPortal). 2. VOD Portal (iVODPortal) sends user identification information to VOD Access Service (iAccessService). 3. A VOD Access Service (iAccessService) checks received information according to user rights set in IPTV DB, and gives the answer to VOD Portal (iVODPortal). 4. VOD Portal (iVODPortal) sends the answer to end user interface (End User).
Post-conditions	User is connected to IPTV network or user has received the response about unsuccessful connection to IPTV network
Variations	When End User connects to TEO AXMEDIS network, authorization is performed automatically and no user intervention is required.
Asynchronous actions	Periodic connection operations must be done to make sure, that this user can still use IPTV service.

2.1.2 End User browses TEO AXMEDIS content

UCId	EU2.
Use case	Browse AXMEDIS TEO content
Description	User can browse AXMEDIS TEO content in order to find the desired object
Actors	End User

Assumptions	End User is connected to TEO IPTV network, has the rights for watching AXMEDIS content and selected browsing menu.
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends content browsing request to VOD Portal (iVODPortal). 2. VOD Portal (iVODPortal) sends request for the list of categories of content to VOD Access Service (iAccessService) 3. VOD Access Service (iAccessService) gets the requested list from AXDB. 4. The requested list of categories of content is sent to VOD Access Service (iAccessService) which sends it to VOD Portal (iVODPortal) and browsing manager presents this list to the user interface (End User). 5. End user interface (End User) selects the category from the list presented by VOD Portal (iVODPortal). 6. VOD Portal (iVODPortal) sends request for the selected category content to VOD Access Service (iAccessService). 7. VOD Access Service (iAccessService) gets the requested category content from AXDB. 8. The requested category content is sent to VOD Access Service (iAccessService) which sends it to VOD Portal (iVODPortal) and browsing manager presents this category to the user interface (End User). 9. End user interface (End User) selects the object from the category content presented by VOD Portal (iVODPortal). 10. VOD Portal (iVODPortal) sends request for the selected object to VOD Access Service (iAccessService). 11. VOD Access Service (iAccessService) gets the requested object from AXDB. 12. The requested object is sent to VOD Access Service (iAccessService) which sends it to VOD Portal (iVODPortal) and browsing manager presents this category to the user interface (End User).
Post-conditions	End user has received information about desired object.
Variations	Every time, when object is selected/inserted/deleted/modified in AXDB, event log update is performed.
Asynchronous actions	

2.1.3 End User views trailers

UCId	EU2.1
Use case	View trailer
Description	User can view existing trailers in TEO AXDB.
Actors	End User
Assumptions	End User is connected to TEO IPTV network, has the rights for watching AXMEDIS content and has selected browsing menu and desired AXMEDIS video object.
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends the request for playing selected AXMEDIS video object trailer to local player interface

	(iAXSTBPlayer). 2. Local player interface (iAXSTBPlayer) sends the request for playing the AXMEDIS object trailer to VOD Access Service (iAccessService). 3. VOD Access Service (iAccessService) sends the request about the user playing the AXMEDIS object trailer to AXDB. 4. VOD Access Service (iAccessService) sends the request to VOD server to stream the object to the user. 5. VOD server starts streaming video to the local player interface (iAXSTBPlayer). 6. Local player interface (iAXSTBPlayer) plays video stream to end user interface (End User).
Post-conditions	End user watches the requested trailer.
Variations	Every time, when trailer is selected/inserted/deleted/modified in AXDB, event log update is performed.
Asynchronous actions	

2.1.4 End User buys AXMEDIS content

UCId	EU3
Use case	Buy AXMEDIS content
Description	User buys and plays AXMEDIS video object which is streamed from VOD server
Actors	End User
Assumptions	User is connected to TEO IPTV network, has the rights for watching AXMEDIS content and has selected the object he wants to play.
Steps	1. End user interface (End User) sends the request for playing selected AXMEDIS video object to local player interface (iAXSTBPlayer) 2. Local player interface (iAXSTBPlayer) sends the request to VOD Access Service (iAccessService) to check payment conditions. 3. VOD Access Service (iAccessService) checks user payment conditions, sending request to IPTV DB. (If object is chargeable and end user has no pre-payment, VOD Access Service sends request to player interface about required payment) 4. VOD Access Service (iAccessService) sends license request for selected object to AXMEDIS PMS and sends received license to player interface (iAXSTBPlayer) (EU6.). 5. Local player interface (iAXSTBPlayer) acquires AXMEDIS stream info resource using AxObjectManager. 6. Local player interface (iAXSTBPlayer) unprotects AXMEDIS stream info resource using ProtectionProcessing. 7. Local player interface (iAXSTBPlayer) convert to unprotected AXMEDIS resource using createAxResource. 8. Local player interface (iAXSTBPlayer) get a data source for reading stream information using AxResource. 9. Local player interface (iAXSTBPlayer) sends the request to VOD server to stream the object to the user.

	10. Local player interface (iAXSTBPlayer) sends request to VOD Access Service (iAccessService) for plying. 11. VOD Access Service sends request for start streaming to VOD Server. 12. VOD server starts streaming video to the local player interface (iAXSTBPlayer). 13. Local player interface (iAXSTBPlayer) plays video stream to end user interface (End User).
Post-conditions	The information about playing the object is stored in AXDB and AXMEDIS video object is represented to end user.
Variations	Every time, when object is played/stopped/paused/seeked in VOD server, event log update in AXDB is performed. If object is chargeable End User must enter PIN number and choose payment method as described in sections 2.1.12 and 2.1.13.
Asynchronous actions	

2.1.5 End User stops playing AXMEDIS object video

UCId	EU3.1.
Use case	Stop
Description	User can abort playing AXMEDIS video object which is streamed from VOD server.
Actors	End User
Assumptions	User is connected to TEO IPTV network, has the rights for watching AXMEDIS content and is playing AXMEDIS video object which is streamed from VOD server.
Steps	1. End user interface (End User) sends the request to stop playing AXMEDIS video object to local player interface (iAXSTBPlayer). 2. Local player interface (iAXSTBPlayer) sends the request to stop playing the object to VOD Access Service (iAccessService). 3. VOD Access Service (iAccessService) sends the request to stop streaming the video to VOD server. 4. VOD server aborts streaming AXMEDIS video object to end user. 5. Local player interface (iAXSTBPlayer) sends request to end user interface (End User) to display browser.
Post-conditions	The information about stopping playing of the object is stored in AXDB and end user interface has switched to browser mode.
Variations	Every time, when object is played/stopped/paused/seeked in VOD server, event log update in AXDB is performed.
Asynchronous actions	

2.1.6 End User pauses playing AXMEDIS object video

UCId	EU3.2.
Use case	Pause
Description	User can pause playing AXMEDIS video object which is streamed from

	VOD server.
Actors	End User
Assumptions	User is connected to TEO IPTV network, has the rights for watching AXMEDIS content and is playing AXMEDIS video object which is streamed from VOD server.
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends the request to pause playing AXMEDIS video object to local player interface (iAXSTBPlayer). 2. Local player interface (iAXSTBPlayer) sends the request to pause playing the object to VOD Access Service (iAccessService). 3. VOD Access Service (iAccessService) sends the pausing video request to the to VOD server. 4. VOD server pauses streaming of AXMEDIS video object to end user. 5. Local player interface (iAXSTBPlayer) displays the paused video frame to end user interface (End User).
Post-conditions	The information about pausing the playing of the object is stored in AXDB and end user can see the current frame of paused video.
Variations	Every time, when object is played/stopped/paused/seeked in VOD server, event log update in AXDB is performed.
Asynchronous actions	

2.1.7 End User continues playing AXMEDIS object video

UCId	EU3.3.
Use case	Play
Description	User can continue playing paused AXMEDIS video object which is streamed from VOD server.
Actors	End User
Assumptions	User is connected to TEO IPTV network, has the rights for watching AXMEDIS content and has paused playing AXMEDIS video object which is streamed from VOD server.
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends the request to continue playing paused AXMEDIS video object to local player interface (iAXSTBPlayer). 2. Local player interface (iAXSTBPlayer) sends the request to continue playing the paused video object to VOD Access Service (iAccessService). 3. VOD Access Service (iAccessService) sends the request to continue playing the video to VOD server. 4. VOD server continues streaming of AXMEDIS video object to local player interface (iAXSTBPlayer). 5. Local player interface (iAXSTBPlayer) plays the video stream to end user interface (End User).
Post-conditions	The information about playing the object is stored in AXDB and end user continues watching video.
Variations	Every time, when object is played/stopped/paused/seeked in VOD server, event log update in AXDB is performed.
Asynchronous	

actions	
Design suggestions	
Issues	

2.1.8 End User seeks AXMEDIS object video

UCId	EU3.4.
Use case	Seek
Description	User can seek the desirable point of AXMEDIS video object which is streamed from VOD server.
Actors	End User
Assumptions	User is connected to TEO IPTV network, has the rights for watching AXMEDIS content and is playing AXMEDIS video object which is streamed from VOD server.
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends the request for seeking required point of AXMEDIS video object to local player interface (iAXSTBPlayer): <ol style="list-style-type: none"> a. End user enters time value; b. End user interface (End User) checks entered value. 2. Local player interface (iAXSTBPlayer) sends the request for seeking required point of video object to VOD Access Service (iAccessService). 3. VOD Access Service (iAccessService) sends the request for seeking the video to VOD server. 4. VOD server starts streaming AXMEDIS video object to local player interface (iAXSTBPlayer) at the requested point. 5. Local player interface (iAXSTBPlayer) plays video stream to end user interface (End User).
Post-conditions	The information about seeking the object is stored in AXDB and end user continues viewing video at requested point.
Variations	Every time, when object is played/stopped/paused/seeked in VOD server, event log update in AXDB is performed.
Asynchronous actions	
Design suggestions	
Issues	

2.1.9 End User selects to view his information

UCId	EU4
Use case	View user information
Description	User can get the information about his settings, rights, usage and get the billing information
Actors	End User
Assumptions	User is connected to TEO IPTV network
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends the request for displaying user information to VOD Portal (iVODPortal). 2. VOD Portal (iVODPortal) displays the options for viewing user

	information to end user interface (End User).
Post-conditions	User can select which user information to view.
Variations	
Asynchronous actions	

2.1.10 End User selects to view his usage information

UCId	EU4.1.
Use case	View usage
Description	User can get the information about the content he has used.
Actors	End User
Assumptions	User is connected to TEO IPTV network
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends usage request to VOD Portal (iVODPortal). 2. VOD Portal (iVODPortal) sends usage request to VOD Access Service (iAccessService) 3. VOD Access Service (iAccessService) gets the user usage information from AXDB. 4. The user usage information is sent to VOD Access Service (iAccessService) which sends it to VOD Portal (iVODPortal) and browsing manager presents this information to the user interface (End User).
Post-conditions	User can see his usage information.
Variations	
Asynchronous actions	

2.1.11 End User selects to view his billing information

UCId	EU4.2.
Use case	View billing
Description	User can get his billing information.
Actors	End User
Assumptions	User is connected to TEO IPTV network
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends billing information request to VOD Portal (iVODPortal). 2. VOD Portal (iVODPortal) sends billing information request to VOD Access Service (iAccessService) 3. VOD Access Service (iAccessService) gets the user billing information from Billing DB. 4. The user billing information is sent to VOD Access Service (iAccessService) which sends it to VOD Portal (iVODPortal) and browsing manager presents this information to the user interface (End User).
Post-conditions	User can see his billing information.
Variations	
Asynchronous	

actions	
----------------	--

2.1.12 End User enters his PIN

UCId	EU5
Use case	Enter PIN
Description	User is requested and enters his PIN.
Actors	End User
Assumptions	User is connected to TEO IPTV network and wants to watch chargeable object.
Steps	<ol style="list-style-type: none"> 1. User enters his PIN. 2. End user interface (End User) sends entered user PIN to VOD Portal (iVODPortal). 3. VOD Portal (iVODPortal) sends request to check PIN to VOD Access Service (iAccessService) 4. VOD Access Service (iAccessService) compares received user PIN with user PIN in IPTV DB. 5. VOD Access Service (iAccessService) sends the request about processing the PIN to VOD Portal (iVODPortal). 6. VOD Portal (iVODPortal) presents payment method options to the user interface (End User).
Post-conditions	End user has entered correct PIN and user can select payment method or information about incorrect PIN is presented.
Variations	
Asynchronous actions	

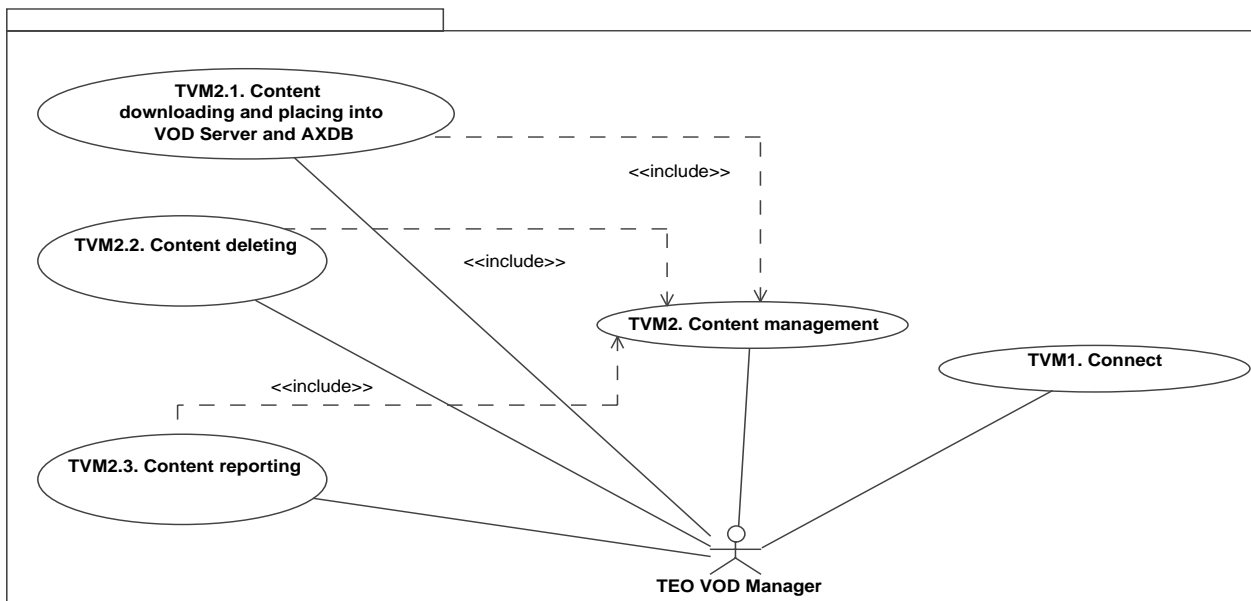
2.1.13 End User chooses payment method

UCId	EU5.1.
Use case	Choose payment
Description	End user can select payment method for viewing AXMEDIS objects.
Actors	End User
Assumptions	User is connected to TEO IPTV network and wants to watch chargeable object.
Steps	<ol style="list-style-type: none"> 1. End user interface (End User) sends setting payment request to VOD Portal (iVODPortal). 2. Local VOD Portal (iVODPortal) sends request to VOD Access Service (iAccessService) about the chosen payment method. 3. VOD Access Service sends the request about chosen payment method to IPTV DB. 4. VOD Access Service sends request about the chosen payment method to Billing DB. 5. End user receives notification about successful payment.
Post-conditions	User payment is confirmed and end user can view the selected object.
Variations	
Asynchronous actions	

2.1.14 License generation

UCId	EU6.
Use case	Generate license
Description	When End User buys content or want watch free content, license must be generated. AXSTBPlayer initialize license generation process, by sending request to VOD Access Service. VOD Access Service interacts with PMS server to generate and retrieve license for content playing purposes.
Actors	End User
Assumptions	End User is connected to TEO IPTV network and wants to watch chargeable (or not) AXMEDIS object.
Steps	<ol style="list-style-type: none"> 1. VOD Access Service interface AccessManager after successful checkpayment() operation initializes license generation process. 2. VOD Access Service interface AccessManager sends initLicenseEndUser() request to PMS server License Generator. 3. License Generator returns the temporal identifier of the license. This identifier is usable while the license is being created. 4. VOD Access Service interface AccessManager sends addGrantEndUser() request to PMS server License Generator to add play right to End User. 5. VOD Access Service interface AccessManager sends finaliseLicenseEndUser() request to PMS server License Generator. 6. License Generator builds the license and, if it is correct, then stores it in the database, and returns a String with the license identifier. 7. VOD Access Service interface AccessManager sends retrieveLicense() request to PMS server License Manager. 8. License Manager retrieves a license from the license database. 9. VOD Access Service interface AccessManager sends retrieved license to AXSTBPlayer interface Player.
Post-conditions	AXSTBPlayer receives generated license and can start object unprotection procedure.
Variations	
Asynchronous actions	

2.2 Use Case model describing TEO VOD Manager actions



In this Use Case model TEO VOD Manager actions are described. TEO VOD Manager can:

- Connect to TEO Content & User Management Portal;
- Manage TEO AXMEDIS content:
 - o Download AXMEDIS content from global AXMEDIS DB, using AXEPTool and place into VOD Server and AXDB;
 - o Delete content from local TEO AXDB;
 - o Make and view reports about usage of AXMEDIS content in TEO IPTV network.

In sections 2.2.1 – 2.2.5 these use cases are described in more detail.

2.2.1 TEO VOD Manager connects to TEO Content & User Management Portal

UCId	TVM1.
Use case	Connect
Description	TEO VOD Manager connects to TEO Content & User Management Portal.
Actors	TEO VOD Manager
Assumptions	TEO VOD Manager knows user name and password.
Steps	<ol style="list-style-type: none"> 1. TEO VOD Manager interface (TEO VOD Manager) sends connection request to TEO Content & User Management Portal interface (iManagement). 2. TEO Content & User Management Portal interface (iManagement) checks received user name and password in IPTV DB and represents the options of managing TEO AXMEDIS system to TEO VOD Manager interface (TEO VOD Manager).
Post-conditions	TEO VOD Manager is connected to TEO Content & User Management Portal and can perform required operations.
Variations	

Asynchronous actions	
-----------------------------	--

2.2.2 TEO VOD Manager selects content management

UCId	TVM2.
Use case	Content management
Description	TEO VOD Manager can manage content in local DB
Actors	TEO VOD Manager
Assumptions	TEO VOD Manager is connected to TEO Content & User Management Portal .
Steps	<ol style="list-style-type: none"> 1. TEO VOD Manager interface (TEO VOD Manager) sends content management request to content management interface (iContentManagement). 2. Content management interface (iContentManagement) represents options for content management to TEO VOD Manager interface (TEO VOD Manager).
Post-conditions	TEO VOD Manager can select required content management operation.
Variations	
Asynchronous actions	

2.2.3 TEO VOD Manager downloads content from global AXMEDIS network to local DB

UCId	TVM2.1.
Use case	Content downloading and placing into VOD Server and AXDB
Description	TEO VOD Manager downloads required objects from global AXMEDIS network and puts into VOD Server and AXDB.
Actors	TEO VOD Manager
Assumptions	TEO VOD Manager is connected to TEO Content & User Management Portal and knows which object or object group must be downloaded into local AXDB
Steps	<ol style="list-style-type: none"> 1. TEO VOD Manager sends object searching request to content management interface (iContentManagement). 2. Content management interface (iContentManagement) uses iContentDownloader interface for searching AXMEDIS object using AXEPTool. 3. Content management interface (iContentManagement) presents search results to TEO VOD Manager interface. 4. TEO VOD Manager selects required object or object group and sends getting the object request to content management interface (iContentManagement). 5. Content management interface (iContentManagement) sends getObject request to iContentDownloader interface. 6. iContentDownload interface sends getAXObject request to AXEPTool. 7. AXEPTool presents downloaded object to iContentDownloader

	<p>interface.</p> <p>8. iContentDownloader interface puts object into VOD Server and AXDB:</p> <ol style="list-style-type: none"> places full object into VOD Server; modifies object by deleting video data; places modified object (metadata and stream information) into AXDB.
Post-conditions	New object is placed in VOD Server and AXDB
Variations	
Asynchronous actions	Every time, when object is selected/inserted/deleted/modified in AXDB, event log update is performed.

2.2.4 TEO VOD Manager deletes content from local DB

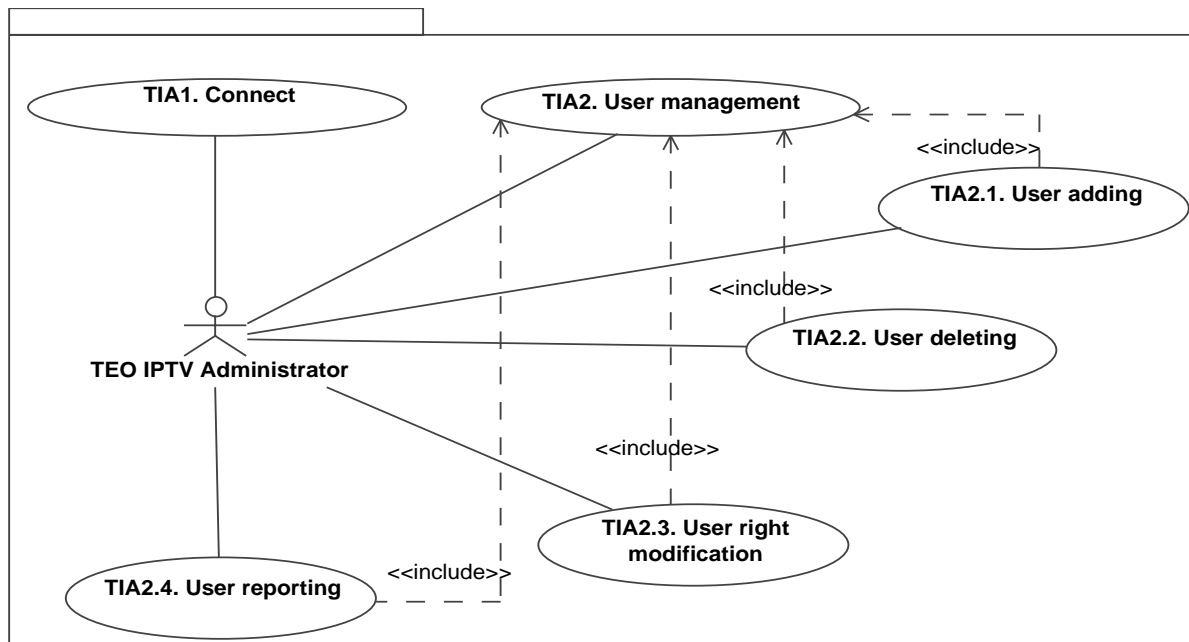
UCId	TVM2.2.
Use case	Content deleting
Description	TEO VOD Manager can delete the selected content from local DB.
Actors	TEO VOD Manager
Assumptions	TEO VOD Manager is connected to TEO Content & User Management Portal and knows which object or object group must be deleted.
Steps	<ol style="list-style-type: none"> TEO VOD Manager sends request for the list of categories of content to content management interface (iContentManagement). Content management interface (iContentManagement) sends getting the category list request to AXDB. The requested list of categories of content is sent to TEO VOD Manager. TEO VOD Manager selects the category from the received list and sends getting the list objects request to content management interface (iContentManagement). Content management interface (iContentManagement) sends request to AXDB and receives category objects list. Content management interface (iContentManagement) sends received category objects list to TEO VOD Manager. TEO VOD Manager selects the object from the received category content list and sends getting the object request to content management interface (iContentManagement). Content management interface (iContentManagement) sends request to AXDB and receives the object. Content management interface (iContentManagement) sends the received object to TEO VOD Manager. TEO VOD Manager sends deleting request to content management interface (iContentManagement). Content management interface (iContentManagement) sends deleting request to AXDB. Content management interface (iContentManagement) sends deleting request to VOD server.
Post-conditions	The object is deleted from AXDB and VOD server
Variations	
Asynchronous	Every time, when object is selected/inserted/deleted/modified in AXDB,

actions	event log update is performed.
----------------	--------------------------------

2.2.5 TEO VOD Manager makes content report

UCId	TVM2.3.
Use case	Content reporting
Description	TEO VOD Manager views usage information about desired AXMEDIS object.
Actors	TEO VOD Manager
Assumptions	TEO VOD Manager is connected to content management system and knows the object about which usage information is required.
Steps	<ol style="list-style-type: none"> 1. TEO VOD Manager sends request for the list of categories of content to content management interface (iContentManagement). 2. Content management interface (iContentManagement) sends getting the category list request to AXDB. 3. The requested list of categories of content is sent to TEO VOD Manager. 4. TEO VOD Manager selects the category from the received list and sends getting the list objects request to content management interface (iContentManagement). 5. Content management interface (iContentManagement) sends request to AXDB and receives category objects list. 6. Content management interface (iContentManagement) sends received category objects list to TEO VOD Manager. 7. TEO VOD Manager selects the object from the received category content list and sends getting the object request to content management interface (iContentManagement). 8. Content management interface (iContentManagement) sends request to AXDB and receives the object. 9. Content management interface (iContentManagement) sends received object to TEO VOD Manager. 10. TEO VOD Manager sends getting the usage request to content management interface (iContentManagement). 11. Content management interface (iContentManagement) sends request to AXDB and receives usage information. 12. Content management interface (iContentManagement) sends received object usage information to TEO VOD Manager.
Post-conditions	TEO VOD Manager can see AXMEDIS object usage information.
Variations	
Asynchronous actions	

2.3 Use Case model describing TEO IPTV Administrator actions



In this Use Case model TEO IPTV Administrator actions are described. TEO IPTV Administrator can:

- Connect to TEO Content & User Management Portal;
- Manage TEO AXMEDIS users:
 - o Add user to TEO AXMEDIS DB;
 - o Delete user from TEO AXMEDIS DB;
 - o Set user rights to use AXMEDIS content in TEO IPTV network;
 - o Make and view reports about user actions in TEO AXMEDIS DB.

In sections 2.3.1 – 2.3.6 these use cases are described in more detail.

2.3.1 TEO IPTV Administrator connects to TEO Content & User Management Portal

UCId	TIA1.
Use case	Connect
Description	TEO IPTV Administrator connects to TEO Content & User Management Portal.
Actors	TEO IPTV Administrator
Assumptions	TEO IPTV Administrator knows user name and password.
Steps	<ol style="list-style-type: none"> 1. TEO IPTV Administrator interface (TEO IPTV Administrator) sends connection request to TEO Content & User Management Portal interface (iManagement). 2. TEO Content & User Management Portal interface (iManagement) checks received user name and password in IPTV DB and represents the options of managing TEO AXMEDIS system to TEO IPTV Administrator interface (TEO IPTV Administrator).
Post-conditions	TEO IPTV Administrator is connected to TEO Content & User

	Management Portal and can perform required operations.
Variations	
Asynchronous actions	

2.3.2 TEO IPTV Administrator selects user management

UCId	TIA2.
Use case	User management
Description	TEO IPTV Administrator can manage AXMEDIS users in TEO IPTV network.
Actors	TEO IPTV Administrator
Assumptions	TEO IPTV Administrator is connected to TEO Content & User Management Portal.
Steps	<ol style="list-style-type: none"> 1. TEO IPTV Administrator interface (TEO IPTV Administrator) sends user management request to user management interface (iUserManagement). 2. User management interface (iUserManagement) displays the options for user management to TEO IPTV Administrator interface (TEO IPTV Administrator).
Post-conditions	TEO IPTV Administrator can select required user management operation.
Variations	
Asynchronous actions	

2.3.3 TEO IPTV Administrator adds new user

UCId	TIA2.1.
Use case	User adding
Description	TEO IPTV Administrator adds new user to IPTV DB.
Actors	TEO IPTV Administrator
Assumptions	TEO IPTV Administrator is connected to user management system and knows all required information about new user.
Steps	<ol style="list-style-type: none"> 1. TEO IPTV Administrator interface (TEO IPTV Administrator) checks entered user information. 2. TEO IPTV Administrator sends adding the user request to user management interface (iUserManagement). 3. User management interface (iUserManagement) sends request to IPTV DB. 4. User management interface (iUserManagement) returns information about user adding to TEO IPTV Administrator interface (TEO IPTV Administrator).
Post-conditions	New user is added to IPTV DB.
Variations	
Asynchronous actions	

2.3.4 TEO IPTV Administrator deletes user

UCId	TIA2.2.
Use case	User deleting
Description	TEO IPTV Administrator deletes the selected user from IPTV DB.
Actors	TEO IPTV Administrator
Assumptions	TEO IPTV Administrator is connected to user management system and knows which user must be deleted.
Steps	<ol style="list-style-type: none"> 1. TEO IPTV Administrator sends getting the user list request to user management interface (iUserManagement). 2. User management interface (iUserManagement) requests the user list from IPTV DB. 3. User management interface (iUserManagement) receives the user list from IPTV DB and sends it to TEO IPTV Administrator interface (TEO IPTV Administrator). 4. TEO IPTV Administrator interface (TEO IPTV Administrator) selects the user and sends deleting user request to user management interface (iUserManagement). 5. User management interface (iUserManagement) sends the request to IPTV DB for user deleting. 6. User management interface (iUserManagement) receives acknowledgement about user deleting and sends it to TEO IPTV Administrator interface (TEO IPTV Administrator).
Post-conditions	User is deleted from IPTV DB.
Variations	
Asynchronous actions	

2.3.5 TEO IPTV Administrator modifies user rights

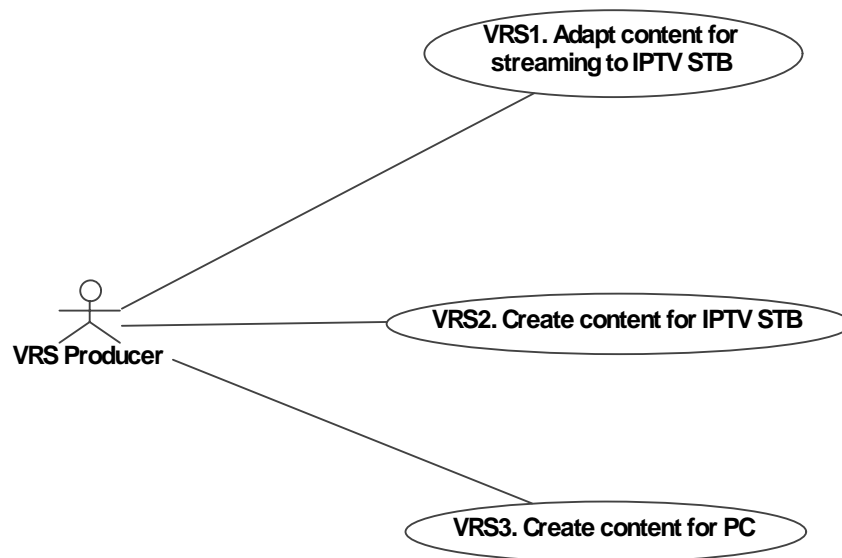
UCId	TIA2.3.
Use case	User right modification
Description	TEO IPTV Administrator modifies user rights according to the requirements.
Actors	TEO IPTV Administrator
Assumptions	TEO IPTV Administrator is connected to user management system and knows which user rights must be modified and how.
Steps	<ol style="list-style-type: none"> 1. TEO IPTV Administrator sends getting the user list request to user management interface (iUserManagement). 2. User management interface (iUserManagement) requests user list from IPTV DB. 3. User management interface (iUserManagement) receives user list from IPTV DB and sends it to TEO IPTV Administrator interface (TEO IPTV Administrator). 4. TEO IPTV Administrator interface (TEO IPTV Administrator) selects user and sends modifying user rights request to user management interface (iUserManagement). 5. User management interface (iUserManagement) sends request to IPTV DB for user rights modification.

	6. User management interface (iUserManagement) receives acknowledgement about user rights modification and sends it to TEO IPTV Administrator interface (TEO IPTV Administrator).
Post-conditions	User rights are modified.
Variations	
Asynchronous actions	

2.3.6 TEO IPTV Administrator makes user report

UCId	TIA2.4.
Use case	User reporting
Description	TEO IPTV Administrator views the report about user, his content usage and billing information.
Actors	TEO IPTV Administrator
Assumptions	TEO IPTV Administrator is connected to user management system and knows which user or user group information is needed.
Steps	<ol style="list-style-type: none"> 1. TEO IPTV Administrator sends getting the user list request to user management interface (iUserManagement). 2. User management interface (iUserManagement) requests user list from IPTV DB. 3. User management interface (iUserManagement) receives user list from IPTV DB and sends it to TEO IPTV Administrator interface (TEO IPTV Administrator). 4. TEO IPTV Administrator interface (TEO IPTV Administrator) selects user and sends getting user report request to user management interface (iUserManagement). 5. User management interface (iUserManagement) sends request for user personal information to IPTV DB. 6. User management interface (iUserManagement) sends request for user content usage information to AXDB. 7. User management interface (iUserManagement) sends request for selected user billing information to Billing DB. 8. User management interface (iUserManagement) receives all requested information and sends it to TEO IPTV Administrator interface (TEO IPTV Administrator).
Post-conditions	TEO IPTV Administrator has received user usage and billing information.
Asynchronous actions	

2.4 Use Case model describing content production in VRS Factory



In this Use Case model VRS Content Producer actions are described. VRS Content Producer can:

- Adapt video content for streaming to IPTV STB;
- Create content for IPTV STB for streaming (TEO demonstrator);
- Create content for PC (Elion demonstrator).

Content creation for both platforms consists of several steps. These steps can be performed automatically using AXCP scripts or manually using AXMEDIS Editor, DRM Editor, AXEPTool, etc. Content adaptation for streaming to IPTV STB must be done manually regarding necessity use of external tools.

In sections 2.4.1 – 2.4.3 use cases for content creation in ELTEO demonstrator setup are described in more detail.

2.4.1 Content adaptation for streaming to IPTV STB

UCId	VRS1.
Use case	Adapt video content for streaming to IPTV STB
Description	VRS Content Producer adapts video content for IPTV STB (TEO demonstrator) manually using external tools (Sony Vegas, Elecard XMuxer)
Actors	VRS Content Producer
Assumptions	VRS Content Producer has video material, which can be used for creating new AXMEDIS object and has tools for MPEG2-TS file formation
Steps	<ol style="list-style-type: none"> 1. Firstly in the process it is necessary to define inserted ad regions, exporting the timecodes to XML file. For this job Sony Vegas external tool is used. Timecode XML file later will be used as metadata in AxObject to define non skippable regions location for PCR of the STB. Also the same external tool - Sony Vegas is used for splitting interleaved broadcast ready media file to video and audio streams according to desired template of STB – video stream MPEG-2 CBR 4Mbps, 720x576, 25 fps and audio stream MPEG-1 224 Kbps. 2. External tool Elecard XMuxer is called to format MPEG-2 TS file from

	the audio and video streams.
Post-conditions	Video content is prepared in required format for streaming.
Variations	
Asynchronous actions	

2.4.2 Producer creates AXMEDIS object for streaming to IPTV STB

UCId	VRS2.
Use case	Create content for IPTV STB
Description	VRS Content Producer creates streaming video AXMEDIS object for IPTV STB (TEO demonstrator)
Actors	VRS Content Producer
Assumptions	VRS Content Producer has prepared streaming video content, which can be used for creating new AXMEDIS object and has installed AXMEDIS tools
Steps	<ol style="list-style-type: none"> 1. New AXMEDIS object is created and MPEG-2 TS video resource and metadata are added. (This Use Case is described in “DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b”) 2. Executing MPEGTSAadaptation plugin for scrambling MPEG-2 Transport Stream (TS) resource using DVB-CSA algorithm. The MPEGTSAadaptationPlugin encrypt all video and audio channels of stream and leave unencrypted frames containing Program Clock Reference (PCR) which is critical for streaming control. Each scrambled frame of transport stream has to be marked as scrambled according to ISO13818-1 standard. The second task of plug-in is to collect information about all scrambled elementary MPEG-2 TS streams (ES). This task associates each ES with DVB-CSA Control Word (CW) which was used for scrambling the ES. 3. New AXMEDIS object is protected and distribution license is generated (This Use Case is described in “DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b”) 4. New AXMEDIS object is published in B2B P2P Axmedis network using AXEPTool (This Use Case is described in “DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b”).
Post-conditions	New streaming video AXMEDIS object is created and published in B2B P2P Axmedis network
Variations	
Asynchronous actions	

2.4.3 VRS Producer creates AXMEDIS object content for PC

UCId	VRS3.
Use case	Create content for PC
Description	VRS Producer creates Axmedis objects for PC platform (Elion demonstrator)
Actors	VRS Producer
Assumptions	VRS Producer has video material, which can be used for creating new AXMEDIS object and has installed AXMEDIS tools
Steps	<ol style="list-style-type: none"> 1. MultimediaAdaptation plugin is executed for resource processing (mpeg-4 for Elion PC). 2. VideoAdaptation plugin is executed for video resizing (640x480, 15fps

	<p>for Elion PC).</p> <ol style="list-style-type: none"> 3. New AXMEDIS object is created and video resource and metadata are added. (This Use Case is described in “DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b”) 4. New AXMEDIS object is protected and distribution license is generated (This Use Case is described in “DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b”) 5. New AXMEDIS object is published in B2B P2P Axmedis network using AXEPTool (This Use Case is described in “DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b”).
Post-conditions	New AXMEDIS object for PC platform is created and published in B2B P2P Axmedis network
Variations	
Asynchronous actions	

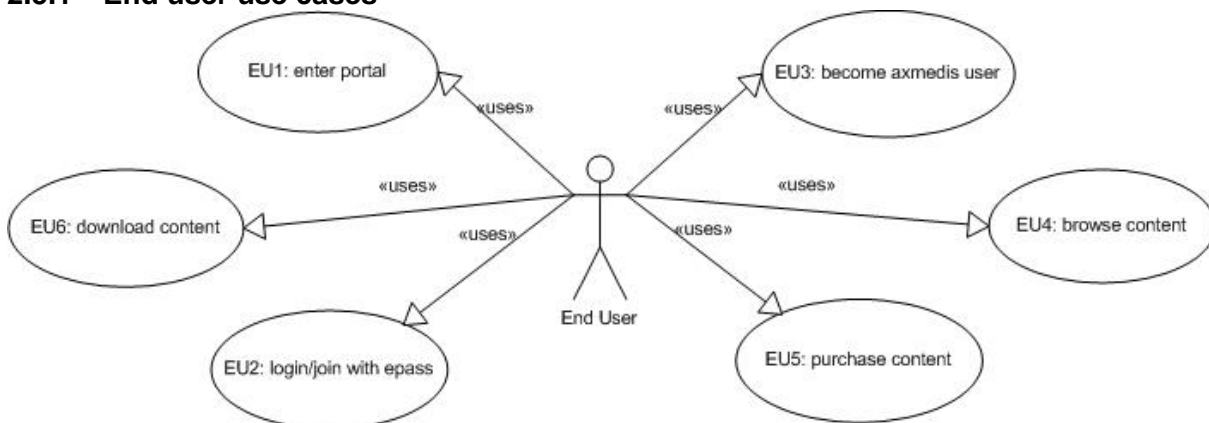
2.5 General use cases and scenarios of Elion PC demonstrator (Elion)

ELION AXMEDIS exploitation system uses two actors: End User and ELION Manager. General goal of the system is to provide classical VOD model for End Users.

End User is anonymous web user who owns e-pass account or is able to register for one. End user must also use a windows based pc with ability to download files off internet and the sufficient rights to install AXMEDIS player. In addition user must own internet bank account in one of the supported local banks or a mobile phone from local service providers to perform payments for commercial media, integration of credit card payment is scheduled for q3 / q4 2007.

ELION Manager will be referred simply as Manager is person responsible for downloading AXMEDIS objects off AXMEDIS B2B P2P network and publishing them to ELION exploitation system for End User consumption.

2.5.1 End user use cases



This use case model describes all main points of End User actions tied or related for AXMEDIS system consumption. The End User can;

- Enter www.hot.ee portal
- Create e-pass account or use existing account to login to www.hot.ee for service consumption
- Associate epass user account with AXMEDIS for AXMEDIS system exploitation
- Acquire AXMEDIS Player after association
- Download AXMEDIS Objects, purchase rental (24 hour playback) rights for commercial AXMEDIS Objects.

2.5.1.1 EU1: Enter portal

UCId	EU1
Use case	Enter to www.hot.ee portal
Description	User enters the community & communication portal
Actors	End User
Assumptions	User has internet connection, windows based operating system and supported web browser.
Steps	5. User enters the web address http://www.hot.ee
Post-conditions	
Variations	
Asynchronous actions	

2.5.1.2 EU2: Login/join with epass

UCId	EU2
Use case	User logs in using e-pass
Description	User is authenticated
Actors	End User
Assumptions	User has internet connection; windows based operating system, supported browser and e-pass account.
Steps	<ol style="list-style-type: none"> 1. User clicks button “sisene” 2. User is redirected to epass.elion.ee for authorization 3. User enters username and password 4. Epass performs authentication and redirects user back to hot.ee with confirmation and base data.
Post-conditions	User is authenticated for service usage.
Variations	If user does not have the e-pass account it can be created by clicking “Registreeri” and following registration procedure, after registration is done user can complete authorization procedure for hot.ee
Asynchronous actions	

2.5.1.3 EU3: Become Axmedis User

UCId	EU3
Use case	Become AXMEDIS User
Description	User is registered and certified on AXMEDIS network, this is a single time operation performed only once for every unique epass account.
Actors	End User
Assumptions	User has internet connection, windows based operating system, supported browser and has been authenticated by epass.
Steps	<ol style="list-style-type: none"> 1. User clicks AXMEDIS menu 2. Hot performs user check on ELAXAPI_WS using method IsEnabled() method and determines that user has not been associated with AXMEDIS 3. User is redirected to webpage describing service, conditions and related data 4. User presses “Nõustun tingimustega” button and accepts legal terms 5. ELAXAPI_WS performs user registration with method RegisterUser() for automatic AXMEDIS registration. 6. User is presented with AXMEDIS Player download
Post-conditions	User is registered on AXMEDIS network and has AXMEDIS Player downloaded and installed.
Variations	User rejects the terms and conditions and is redirected back to main page.
Asynchronous actions	

2.5.1.4 EU4: Browse content

UCId	EU4
Use case	Browse content
Description	User is presented with available content listing to choose from.
Actors	End User
Assumptions	User has internet connection, windows based operating system, supported

	browser, AXMEDIS player, is authenticated by epass and signed up on AXMEDIS.
Steps	<ol style="list-style-type: none"> 1. Hot.ee uses ELAXAPI_WS to acquire available content listing using GetList() web method 2. Hot.ee parses the result and converts it into html format 3. User is presented with content listing containing <ol style="list-style-type: none"> a. Short description of object in Estonian language b. Price of the object in Estonian currency c. A link for download d. A link for license purchasing
Post-conditions	User has chosen an interesting object for purchasing.
Variations	
Asynchronous actions	

2.5.1.5 EU5: Purchase content/issue license

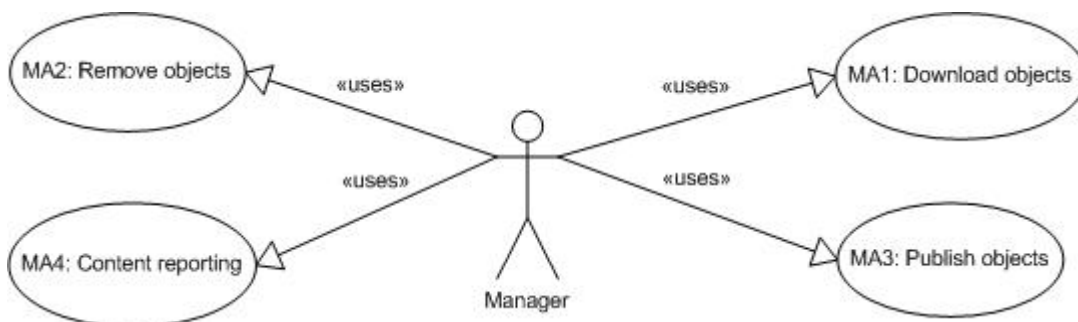
UCId	EU5
Use case	Purchase content
Description	User is directed into payment gateway.
Actors	End User
Assumptions	User has internet connection, windows based operating system, supported browser, AXMEDIS player, is authenticated by epass and signed up on AXMEDIS. In addition user must own internet bank account with supported banks or own a mobile phone subscription with supported local mobile service providers. Credit card support is planned to be added in Q3/Q4 2007
Steps	<ol style="list-style-type: none"> 1. Hot.ee redirects user to epass.elion.ee payment gateway with payment request 2. User is presented with payment options 3. User performs payment according to selected option 4. Epass.elion.ee redirects user back to hot.ee with confirmed payment data 5. Hot.ee invokes GetLicense() method on ELAXAPI_WS for license acquisition 6. ELAXAPI_WS performs all necessary procedures to complete license generation with PMS
Post-conditions	User has purchased 24 hour playback rights for specified object and all of its commercial resources.
Variations	User cannot perform payment, the procedure is cancelled and user is redirected back into browsing page.
Asynchronous actions	Epass.elion.ee uses asynchronous methods to notify about user payment in case that user closes web page or web page communication is corrupted, this data is compared to actions performed by user web interface.

2.5.1.6 EU6: Download content

UCId	EU6
Use case	Download content
Description	User downloads AXMEDIS object for exploitation.
Actors	End User

Assumptions	User has internet connection, windows based operating system, supported browser, AXMEDIS player, is authenticated by epass and signed up on AXMEDIS.
Steps	<ol style="list-style-type: none"> 1. User clicks on objects download link and saves the object on PC 2. User uses AXMEDIS Player to exploit the object according to license limitations.
Post-conditions	User consumes AXMEDIS Object
Variations	User can download the object before step EU5 for exploiting the free of charge resources in the object or have the object from previous licensed use, however user will be required to perform all steps leading up to EU5 to acquire a valid license to consume protected commercial resources.
Asynchronous actions	

2.5.2 Manager use cases



These use cases describe possible actions that Manager can perform to manage content on distribution point. Due to small scale of ELION's operations we do not foresee currently need for management interface, as such all actions are performed by hand in simple manner. All operations are being done over Terminal Services session by connecting to axmedis.neti.tv node responsible for handling all AXMEDIS operations in ELION distributor.

2.5.2.1 MA1: Downloading objects

UCId	MA1
Use case	Download objects
Description	Manager downloads new AXMEDIS objects for distribution
Actors	Manager
Assumptions	Manager is connected to axmedis.neti.tv using terminal services.
Steps	<ol style="list-style-type: none"> 1. Manager uses AXEPTool B2B P2P tool to locate objects for distribution 2. Manager downloads objects using AXEPTool to axmedis.neti.tv file system.
Post-conditions	New AXMEDIS object on file system.
Variations	Manager uploads AXMEDIS object to axmedis.neti.tv node that was acquired by other means than AXEPTool or produced locally.
Asynchronous actions	

2.5.2.2 MA2: Remove objects

UCId	MA2
Use case	Remove object

Description	Manager removes object
Actors	Manager
Assumptions	Manager is connected to axmedis.neti.tv using terminal services.
Steps	<ol style="list-style-type: none"> 1. Manager deletes physical object from file system 2. Manager deletes indexing entry from cache database using local SQL Server manager express interface.
Post-conditions	AXMEDIS Object is removed from distribution.
Variations	
Asynchronous actions	

2.5.2.3 MA3: Publish objects

UCId	MA3
Use case	Publish objects
Description	Manager publishes object
Actors	Manager
Assumptions	Manager is connected to axmedis.neti.tv using terminal services.
Steps	<ol style="list-style-type: none"> 1. Manager copies downloaded object into publishing folder 2. Manager uses AXMEDIS Editor to view object and acquire AXOID 3. [optional] Manager uses AXMEDIS Editor to modify the object. 4. Manager uses local SQL Server Manager Express interface to add object data to distribution cache, following data is entered; <ol style="list-style-type: none"> a. Filename b. AXOID c. Short description in Estonian language d. Price of object [optional] 5. Manager saves entry with the object becoming publicly available
Post-conditions	User consumes AXMEDIS Object
Variations	User can download the object before step EU5 for exploiting the free of charge resources in the object or have the object from previous licensed use, however user will be required to perform all steps leading up to EU5 to acquire a valid license to consume protected commercial resources.
Asynchronous actions	

2.5.2.4 MA4: Content reporting

UCId	MA4
Use case	Content reporting
Description	Manager works with AII and CAMART to configure action log storage
Actors	Manager
Assumptions	Manager is connected to axmedis.neti.tv node using terminal services or uses AII web interface
Steps	<ol style="list-style-type: none"> 1. AII & CAMART are used to schedule action log collection, placement and data format for accounting information exchange; this is standard procedure for AXMEDIS platform. 2. Secondary entries are exported from cache database to verify money transactions made by epass using local Win Forms tool in CSV format.

Post-conditions	Automatic storage & distribution of action logs based reporting with backup verification on payment statements.
Variations	
Asynchronous actions	

3 General architecture and relationships among the specified modules

3.1 General architecture of video on demand distribution via IPTV STB (TEO)

The basic structure for AXMEDIS content distribution toward TEO AXMEDIS IPTV network is presented below. TEO AXMEDIS IPTV demonstrator setup is based on existing TEO Gala TV infrastructure. It will use existing IPTV architecture components in addition with newly developed components.

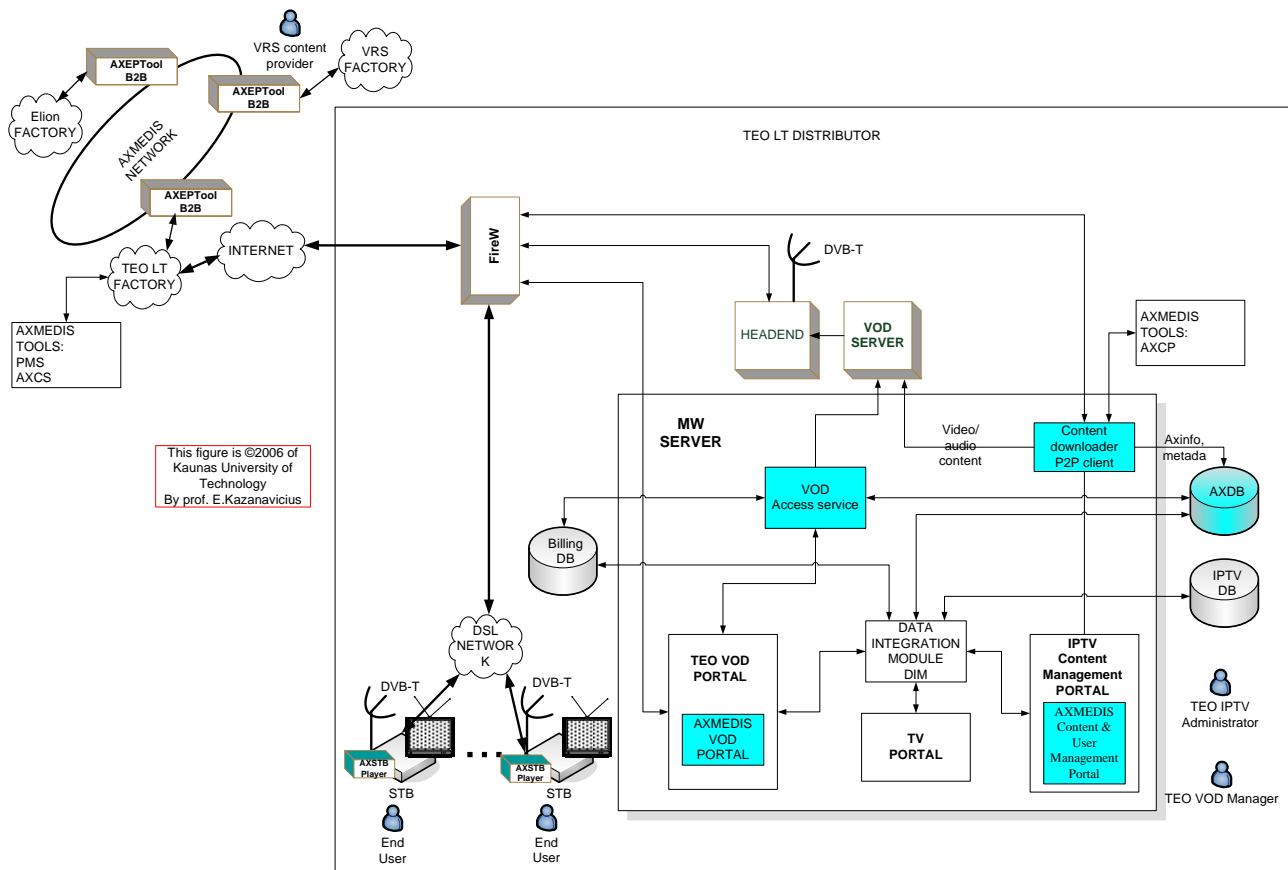
The existing TEO IPTV architecture components are:

- IPTV database;
- Billing DB;
- Data integration module (DIM);
- TEO VOD Portal;
- TV Portal;
- IPTV Content Management Portal;
- VOD Server;
- Headend server.

Newly developed or existing AXMEDIS Framework components:

- AXDB (existing);
- AXMEDIS VOD Portal (new);
- AXMEDIS Content & User Management Portal(new);
- VOD Access Service (new);
- AXEPTool B2B (existing);
- Content downloader P2P client (new);
- AXSTB Player (new (PC player ported on STB));
- AXCS (existing);
- AXOM (existing);
- PMS Server/Client (existing)
- AXMEDIS Query User Interface (existing);
- AXCP (existing).

Entire ELTEO demonstrator will consist of TEO, Elion and VRS factories. Elion demonstrator will use AXMEDIS tools for PC platform. TEO demonstrator will use adopted AXMEDIS tools with newly developed components for STB platform. VRS demonstrator will use AXMEDIS tools with added functionality to produce required content for Elion and TEO demonstrators. Specifications in this document concern TEO IPTV and VRS demonstrator.



TEO AXMEDIS demonstrator is comprised of:

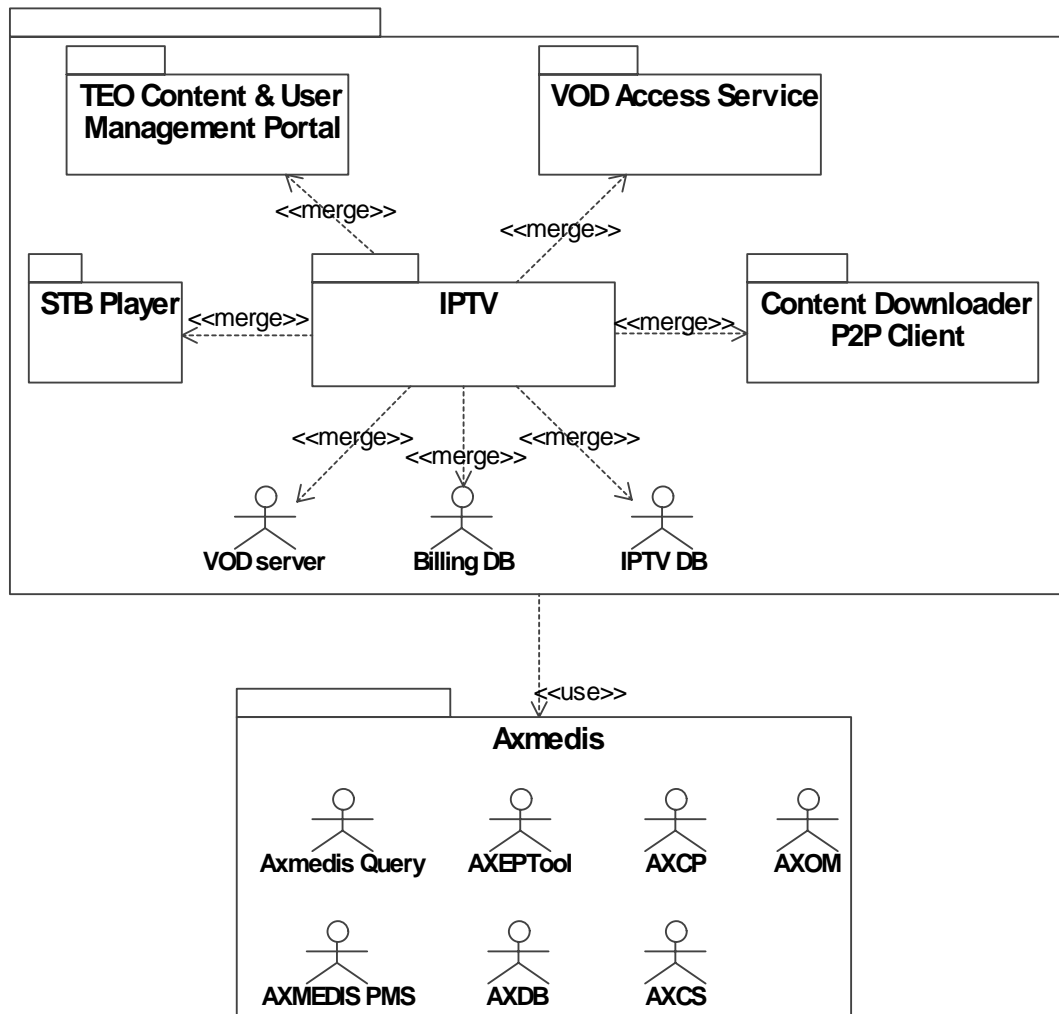
- MW server and related components, as follows:
 1. Existing TEO Gala IPTV database IPTV DB. In this DB information about clients, services and accounting information is stored. Communication with IPTV DB is performed using DIM-ADO.NET.
 2. Existing Billing DB database, where TEO users billing information is stored. Communication with this DB is performed using webservice, which are described in detail in section 5.2.1.
 3. In AXDB the AXMEDIS object metadata and stream information (AxStreamResource) is stored. The MPEG-2 TS resources are excluded due to storage efficiency. This DB is based on AXMEDIS database, specified in DE3.1.2.2.9 “Specification of AXMEDIS database and query support, first update of part of DE3.1.2”.
 4. Data integration module (DIM) is existing TEO solution for integration different data formats in databases.
 5. TV Portal – existing portal for Gala TV. This portal is used in TEO Gala TV for IPTV services:
 - On-line TV channels;
 - Electronic program guide;
 - Gaming, etc.;
 6. AXMEDIS VOD PORTAL, to be developed on TEO Middleware platform. This portal will be used for TEO demonstrator to present AXMEDIS content for End User. AXSTB Player will use this portal to achieve information about content and usage information.
 7. AXMEDIS Content & User Management Portal, a new component to be developed on IPTV Content Management Portal platform and used for TEO

AXMEDIS content and user management. This module will be implemented in IPTV Content Management Portal and used by TEO VOD Manager and TEO IPTV Administrator.

8. VOD Access Service – a component to be newly developed for TEO demonstrator setup. This component will be responsible for proper End User AXSTB Player connection with used DB's and services. VOD Access Service communicates with VOD server, Billing DB, IPTV DB, AXDB and AXMEDIS PMS to present required information to End User and react to End User actions.
 9. AXEPTool B2B tool is used for connecting Middleware server with AXMEDIS network and downloading AXMEDIS data from it. This tool will be used according to specifications described in DE3.1.2.2.10 "Specification of AXMEDIS P2P tools, AXEPTool and AXMEDIS, first update of part of DE3.1.2".
 10. Content downloader P2P client will be a newly developed component for placing downloaded AXMEDIS object to VOD server, updating AXMEDIS object metadata and placing metadata and stream information into AXDB. Content downloader P2P client will use AXCP for automatically downloading new objects designed for TEO demonstrator. This component is a part of AXMEDIS Content & User Management Portal and will be used by TEO VOD Manager for content management purposes.
- VOD SERVER – VOD content storage server. VOD Server will be compatible with existing GALA TV network resources (Kreatel Application platform). The VOD will be capable of streaming DVB-CSA encrypted streams.
 - Headend server – IPTV content distribution and broadcasting server, used for transferring MPEG-2 designated content via TCP/IP network.
 - AXSTB Player – based on AXMEDIS PC Player, described in DE3.1.2.2.5 "Specification of External AXMEDIS Editors/Viewers and Players, first update of DE3.1.2 part B", which is ported into Kreatel STB and used by End User. For AXSTB Player initial draft user interface is described in sections 3.1.4-3.1.8. AXSTB Player communicates with VOD Server, content and user databases, PMS Server through newly developed VOD Access Service.

TEO VOD Manager will use IPTV Content management portal, which enables him to place AXMEDIS objects into content storage.

Below the packages specified for TEO AXMEDIS IPTV demonstrator implementation are presented.



IPTV package is a general package and it includes:

- AXSTB Player;
- VOD Access Service;
- Content Downloader P2P Client;
- TEO Content & User Management Portal:
 - o Content management;
 - o User management.

AXSTB Player package is responsible for responding to End User actions. The components of this package will be installed into Kreatel STB.

TEO Content & User Management Portal has all the tools for managing AXMEDIS content in the local DB. This package is also responsible for TEO AXMEDIS user management. TEO Content & User Management Portal will be used by TEO VOD Manager.

Content Downloader P2P Client package is responsible for placing downloaded AXMEDIS object into VOD server, updating AXMEDIS object metadata and placing metadata and stream information into AXDB. Content downloader P2P client will use AXCP for automatically downloading new objects designed for TEO demonstrator.

VOD Access Service package is responsible for proper End User AXSTB Player connection with AXMEDIS content. This service will act in middleware server.

IPTV package will use VOD server, which will be responsible for storing objects and streaming them to the end users STB. Billing DB will be used for storing and reporting user billing information. IPTV DB will be responsible for storing users' personal information, their rights and permissions. VOD server, Billing DB and IPTV DB are a part of existing IPTV system and will be used for AXMEDIS demonstrator setup.

A set of AXMEDIS tools will be used for proper operation of TEO AXMEDIS demonstrator. AXMEDIS Query and AXEPTool will be used for content searching and downloading from global AXMEDIS network. AXMEDIS PMS server is required for licensing operations. AXDB will be used for storing AXMEDIS objects and their metadata in local TEO IPTV network.

In the presented AXSTB Player and VOD Access Service class diagram two packages from package diagram (AXSTB Player and VOD Access Service) are specified in detail. In AXSTB Player package Player interfaces is specified. Player interface is used for playing streamed video from VOD server. Player interface use iAccessService interface from VOD Access Service package. iAccessService interface communicates with VOD server, Billing DB, IPTV DB, AXDB, and AXMEDIS PMS.

AXSTB Player utilizes AXMEDIS Object Model and Manager to access and play resources. Please, note that the original AXMEDIS object from VRS is never decrypted in TEO VOD servers. The video content reaches client STB in scrambled and untouched form. In order to be able to decrypt the scrambled audio/video content, Player has to utilize AxObjectManager and protection processor to gain Control Words. Without control words it is impossible to gain access to scrambled stream data. This solution utilizes hardware to achieve max performance and AXOM to achieve max AXMEDIS DRM compliance.

The following new classes will be developed in AXMEDIS Framework on STB side:

- Player – represents AXSTB Player which is compliant with TEO VOD server and supports playing of AXMEDIS Objects containing scrambled resources.
- AxStreamResource – holds information how to access scrambled resource.
- AxMPEGStreamInfo – holds information how to descramble specific stream of scrambled resource.

Three interfaces are specified in TEO Content & User Management Portal package: iManagement, iUserManagement and iContentManagement. iManagement interface is used for connecting to TEO Content & User Management Portal. In order to perform this operation Management interface communicates with IPTV DB. iUserManagement interface is responsible for user adding, deleting, modifying and reporting operations. iUserManagement interface communicates with IPTV DB, Billing DB and AXDB. iContentManagement interface is used for content downloading, preparing, deleting and reporting. This interface communicates with AXDB, AXEPTool, AXMEDIS Query and VOD Server.

In AXELTEO demonstrator VRS is responsible for content preparation. VRS must prepare video content which will be used for streaming from VOD Server. AXCP Plug-in MPEGTSAdaptationPlugin will be developed for content preparation in suitable format. This plug-in takes mpeg streaming content through AXMPEGTSRESOURCE_IN parameter from AXMEDIS object and:

1. encodes mpeg stream and returns by AXMPEGTSRESOURCE_OUT parameter;

2. generates MPEG-TS streaming information, with descrambler control words, and returns this info as a resource by AXMPEGSTREAMINFO_OUT parameter.

MPEGTSAdaptationPlugin will be implemented in VRS Factory or used directly by AXEDITOR.

3.2 General architecture of content production & processing demonstrator (VRS)

VRS role in AXMEDIS demonstrator is to provide video content for ELTEO: IPTV STB (TEO) and PC (Elion). Also the objective is to test AXMEDIS Editor for manual content production and AXCP tools for automated content processing.

All VRS media content resources are based on VRS own authored TV series “Funny hidden camera”:

- collection of video clips from “VRS Kamera” TV program series;
- a number of “thematic” clips - home video, local celebrities, wedding, etc...

The special compilation of video clips with or without ads will be produced to test some commercial model for VoD clients of ELTEO demonstrator:

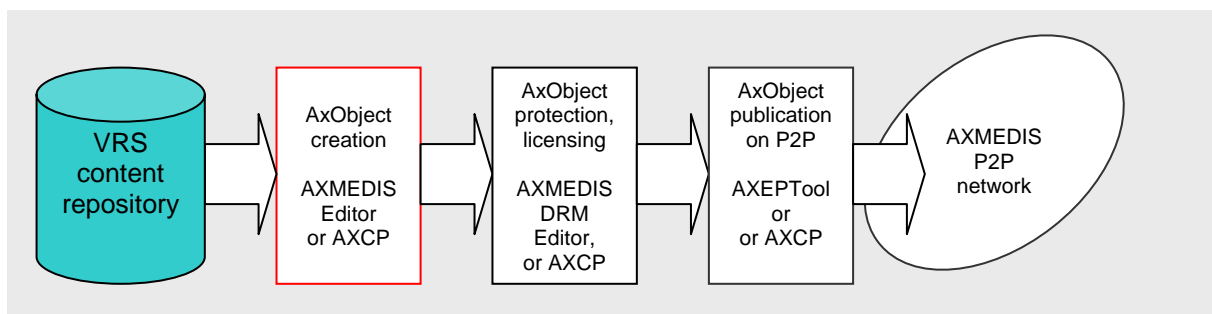
- 5 unique “Home video” blocks with 5 ad inserts;
- 5 unique “Home video” blocks with 2 ad inserts;
- 5 unique “Home video” blocks with no ads;
- 5 unique “Home video” blocks arranged in different manner with 5 ad inserts;

The main objectives of VRS content producing:

- to create protected AXMEDIS object;
- to publish it to P2P B2B network;
- to issue the mother licenses for TEO VoD users and Elion PC users.

The figure below illustrates the basic VRS factory model and it is true when producing the common protected Axmedis objects for main user formats-PC (Elion), etc...

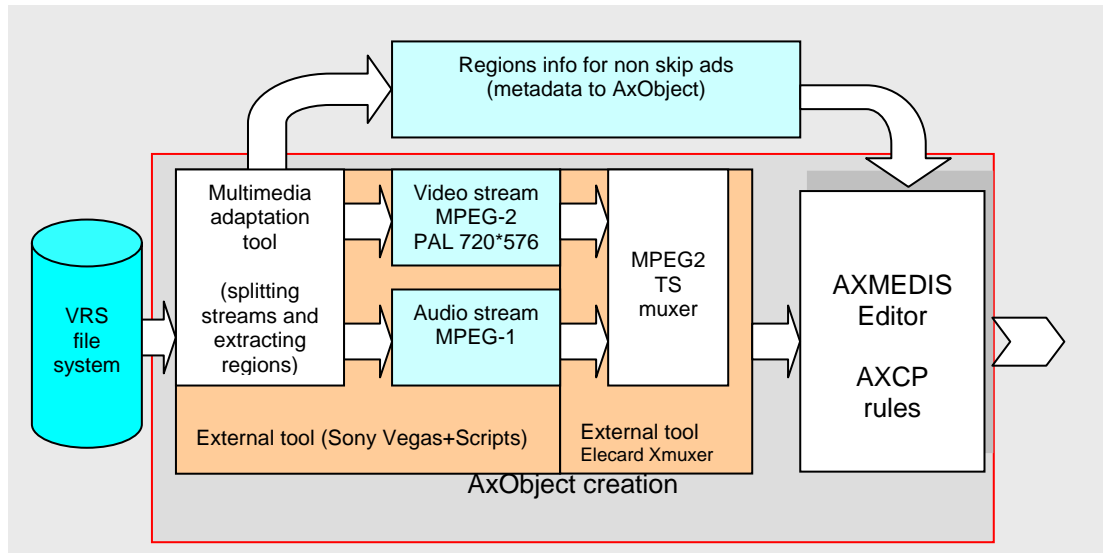
This basic workflow can be done manually using AXMEDIS Editor, AXMEDIS DRM Editor and AXEPTool, or using the number of standard AXMEDIS rules (Create AxObject, etc...) can be processed with the AXCP tools.



The explication below shows the extension of AXMEDIS Object creation section to produce protected AxObject for ELTEO demonstrator.

- The regions for non skippable ads must be defined;

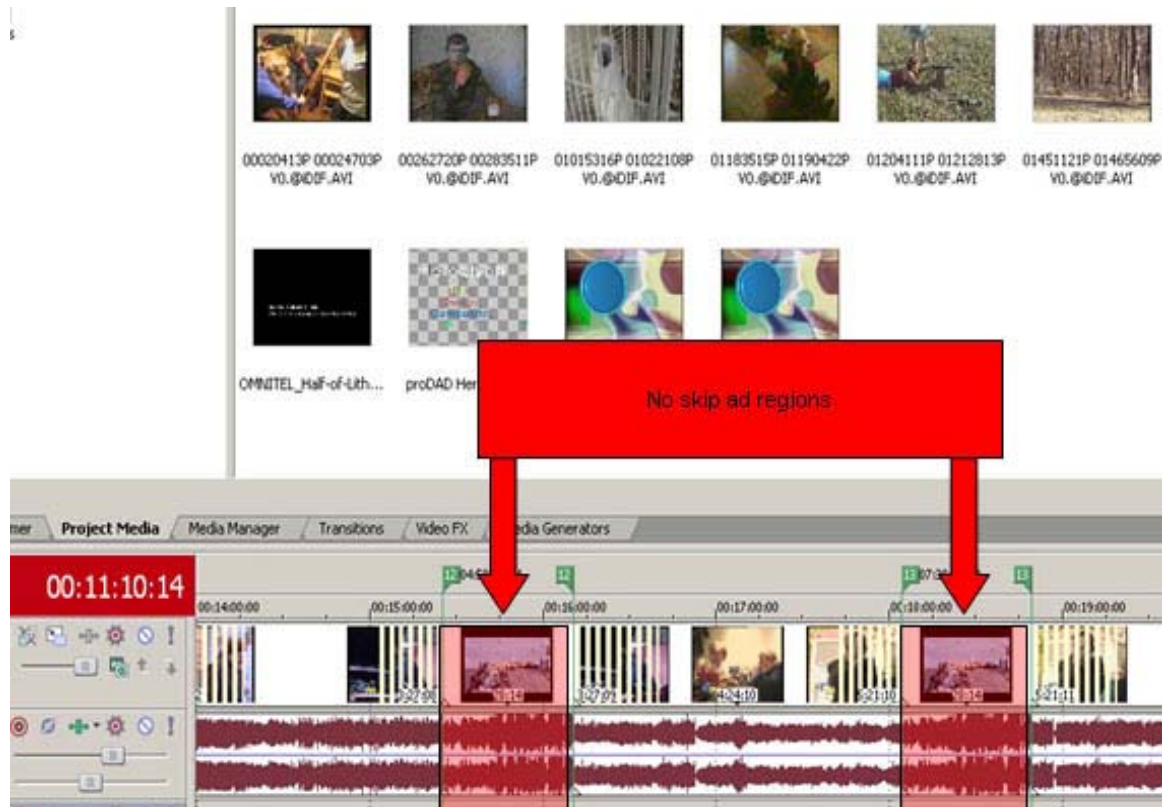
- Due to the IPTV STB hardware requirements (RAW UDP streaming) - MPEG2 TS resource must be embedded into AxObject;
- This MPEG21 Digital Item scrambled with AXCP MPEGTSAadaptationPlugin using DVB-CSA algorithm;
- This transformed AXMEDIS Object protected using AXOM Protection Processor and put to AXDB;



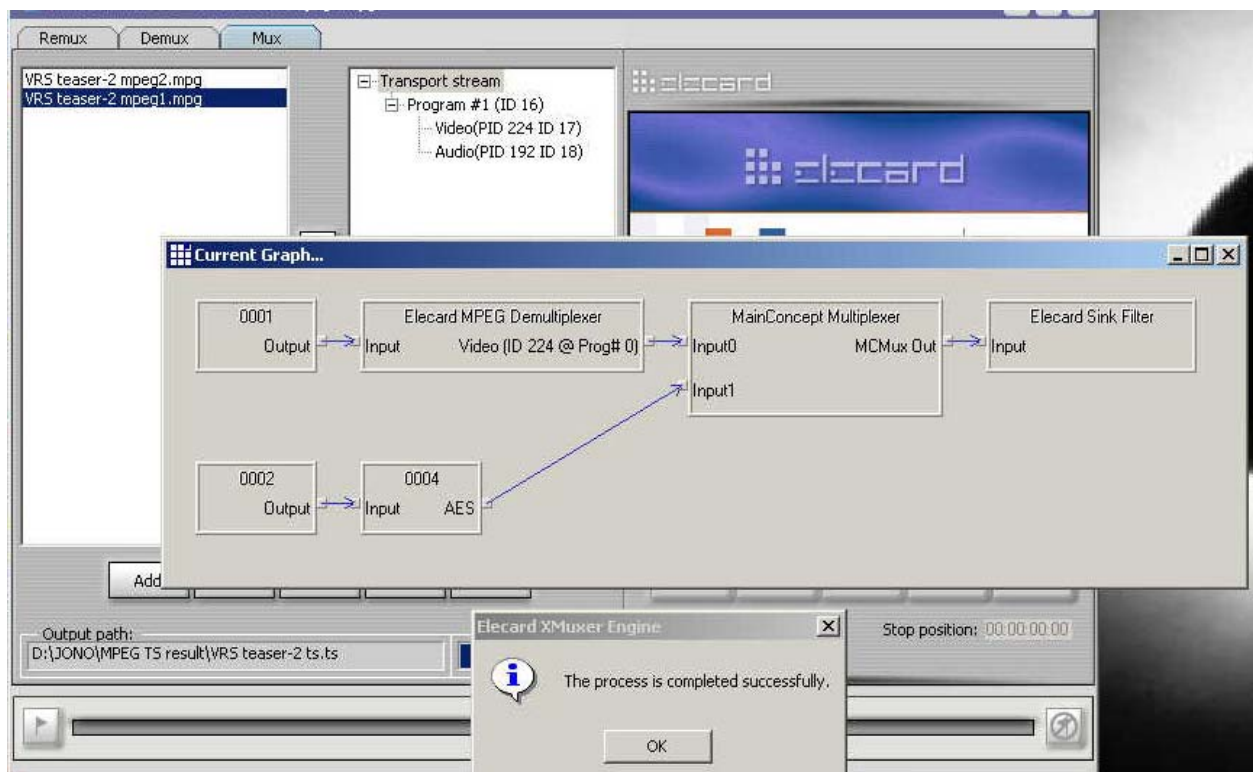
- Metadata in Axmedis objects describes general ranking of VRS media files – themes of video content etc. Also metadata contains the regions for embedded non-skippable ads, with the beginning and end time based on programme clock reference transported within MPEG-TS.

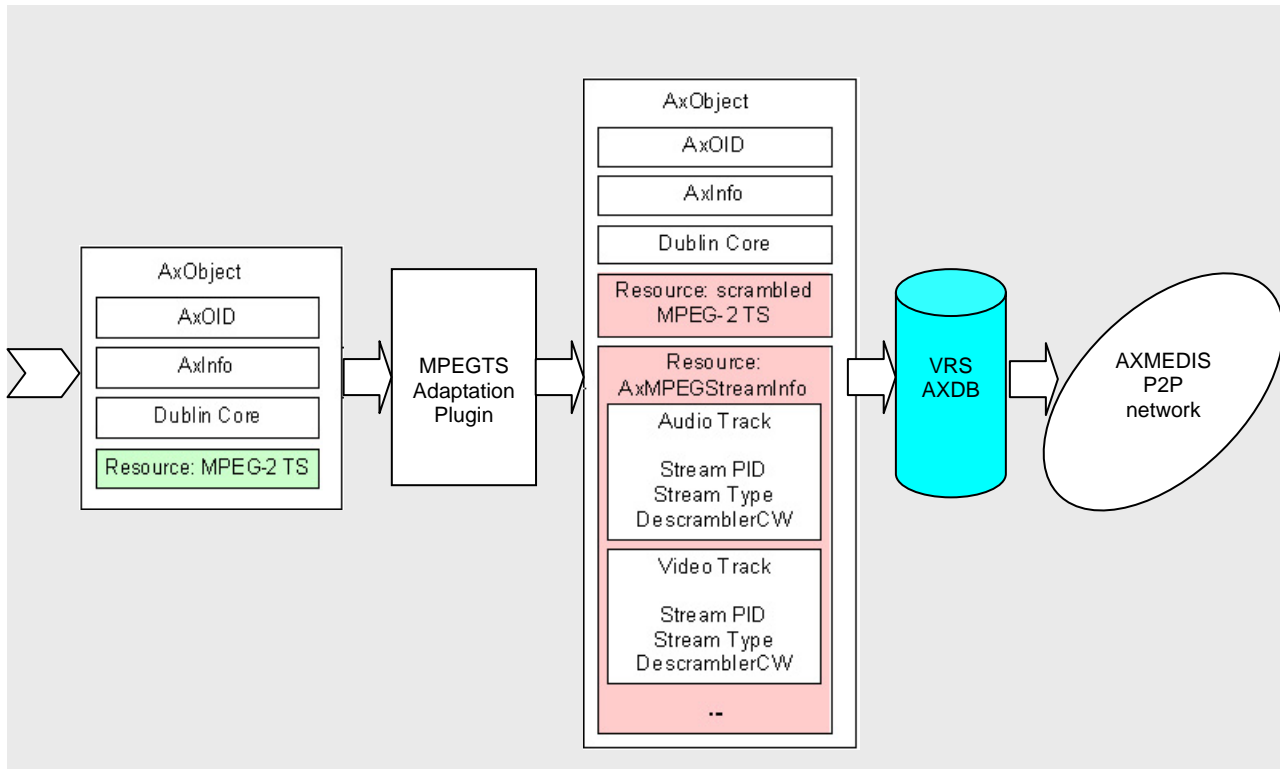
Firstly in the process it is necessary to define inserted ad regions, exporting the timecodes to XML file. This later will be used as metadata in AxObject to define non skippable regions location for PCR of the STB. Also the same external tool - Sony Vegas is splitting interleaved broadcast ready media file to video and audio streams according to desired template of STB – video stream MPEG-2 CBR 4Mbps, 720x576, 25 fps and audio stream MPEG-1 224 Kbps.

The picture below illustrates creation of regions for non skippable ads with external tool.



Another external tool Elecard XMuxer is called to format MPEG-2 TS file from the streams:





The produced content has to be adapted to streaming requirements at VRS Factory using AXMEDIS Editor or AXCP. For this purpose the AXCP MPEGTSAdaptationPlugin has to be developed. This plug-in shall perform two tasks:

- Scramble MPEG-2 Transport Stream (TS) resource using DVB-CSA algorithm. The MPEGTSAdaptationPlugin must encrypt all video and audio channels of stream and leave unencrypted frames containing Program Clock Reference (PCR) which is critical for streaming control. Each scrambled frame of transport stream has to be marked as scrambled according to ISO13818-1 standard.
- The second task of plug-in is to collect information about all scrambled elementary MPEG-2 TS streams (ES). This task associates each ES with DVB-CSA Control Word (CW) which was used for scrambling the ES. This information is returned by the plug-in as a new RESOURCE which is appended to the transformed AXMEDIS Object.

After adoption the transformed AXMEDIS Object must consist of:

1. Scrambled MPEG-2 Transport Stream;
2. Information about scrambled MPEG-2 TS elementary streams and DVB-CSA CWs in resource AxMPEGStreamInfo.

This transformed AXMEDIS Object is protected using AXOM and has to be put to VRS AXDB.

3.3 End User scenarios (Sequence diagrams)

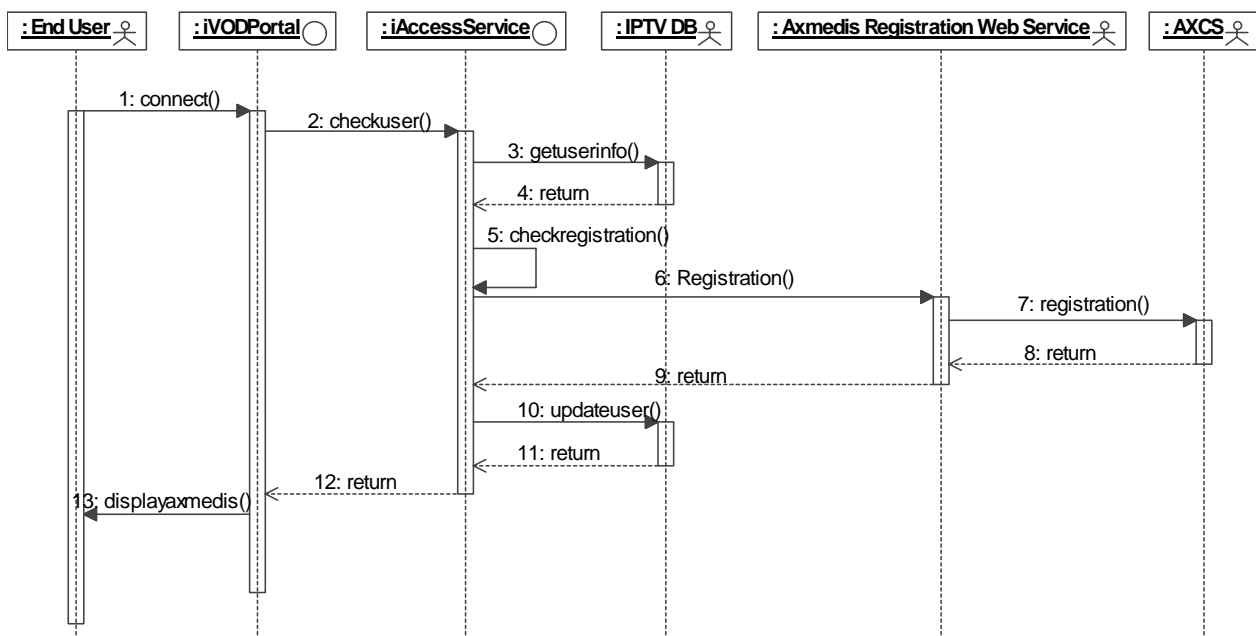
End User scenarios graphically describe End User use cases. Each sequence diagram corresponds to a particular use case. In this section End User interface prototypes are also presented. These prototypes are intended to describe the information and its position in a window, which is presented

to End User, but they do not represent final interface designs. All End User interfaces will be implemented in Gala TV manner and graphical view will be different from presented in this document.

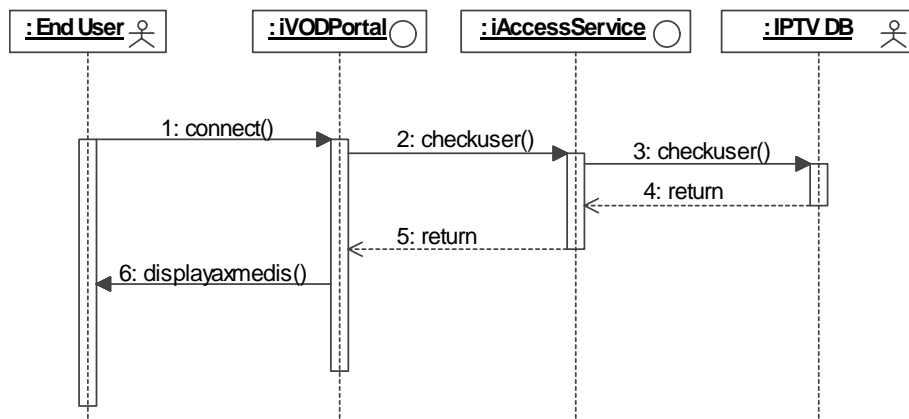
End user interface will be developed to be correctly displayed on a TV set using PAL standard 720x576 pixels. Working place in this screen must use only 90% of picture, because TV set could cut this picture. Working panel in this picture must fit in 460x480 pixels. PIN code window and all others popup windows will be developed to apply 500x108 pixels.

3.3.1 End User connection to TEO network scenario (Sequence diagram)

All users on STB are registered in TEO IPTV DB. For this reason users do not need to register with login and password. End User will be registered in AXMEDIS Certifier and Supervisor on the fly and receive unique global ID, which will be stored in IPTV DB. This step is required only the first time then logging into AXMEDIS Portal.



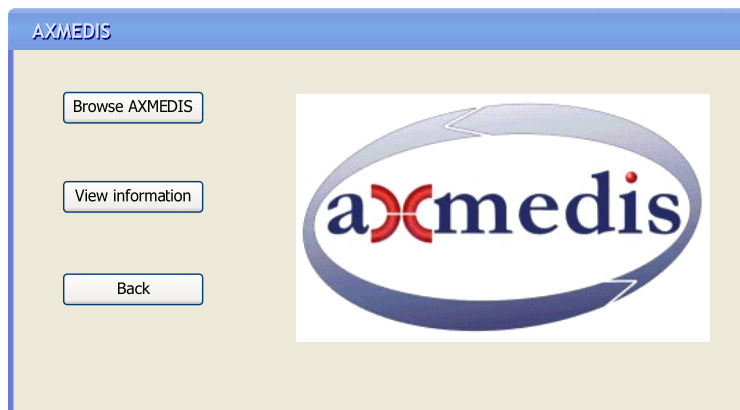
If End User connects to TEO AXMEDIS not for the first time and is already registered in AXCS, then unique global AXMEDIS ID will be requested from IPTV DB. Every user in IPTV DB will have STB unique information and AXMEDIS global unique ID.



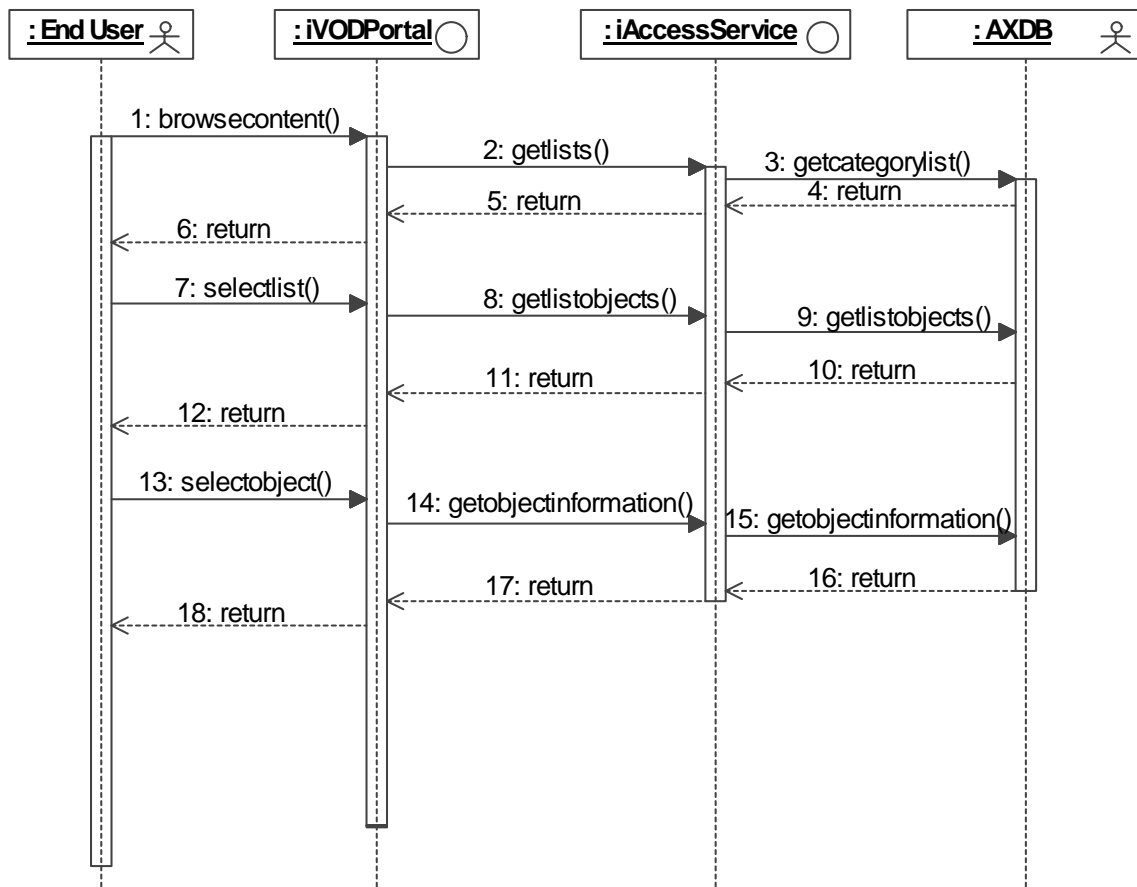
End User connection to TEO AXMEDIS network is possible if he is the subscriber of GALA TV. When End User additionally subscribes AXMEDIS content license, AXMEDIS button will be enabled in GALA TV portal. Connection to TEO AXMEDIS network is done by pressing AXMEDIS button.



End User interface periodically checks user connection possibility, because TEO VOD Manager can disable or delete user at any time. After successful connection, End User interface switches to the view similar to the picture presented below. When End User connects to TEO AXMEDIS network he can browse AXMEDIS content, view his personal information or go back to GALA TV portal.

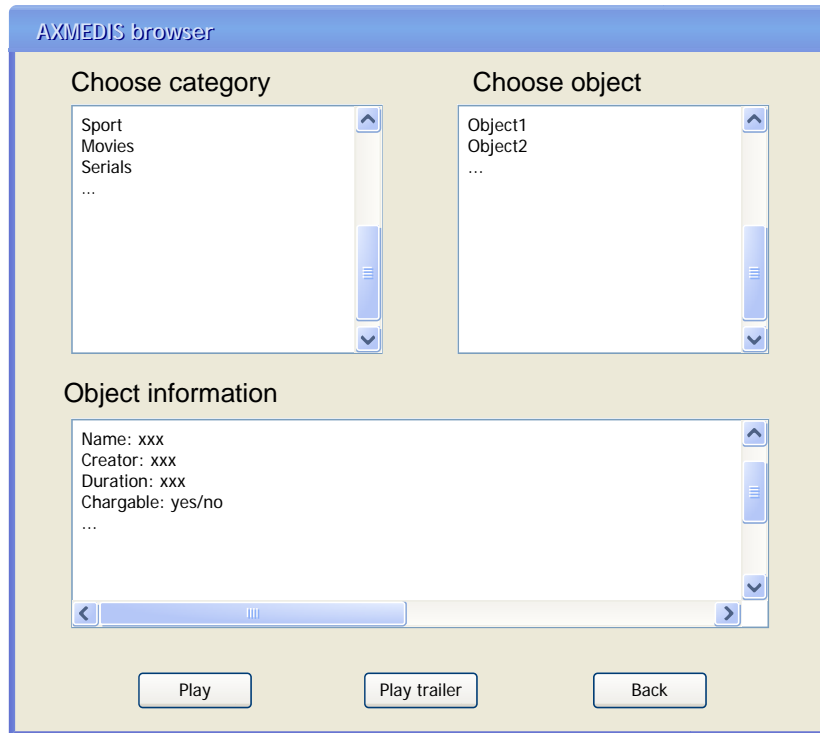


3.3.2 End User browsing in TEO AXMEDIS DB scenario (Sequence diagram)

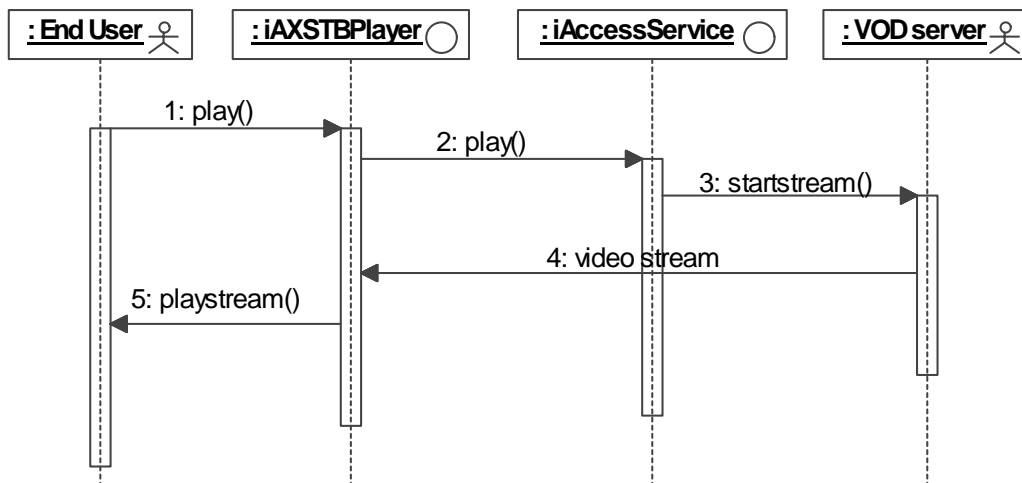


When End User selects the option to browse local AXMEDIS content, End user interface enables the options of selecting the desired object, viewing its description, playing selected object or switching back to previous window.

All AXMEDIS objects are grouped into categories according to their content and/or genre. User selects a desired category and then the object from that category. When user selects the desired object he can see the detailed object description and the Play button is activated. If selected object has a trailer, Play trailer button is also activated. End user can always view the object trailer for free. Most of AXMEDIS objects are chargeable and End user must pay for them. In object information area End user can see all information about selected object, including presence of trailer,. If End user selects Play or Play trailer options, his interface switches to player mode.



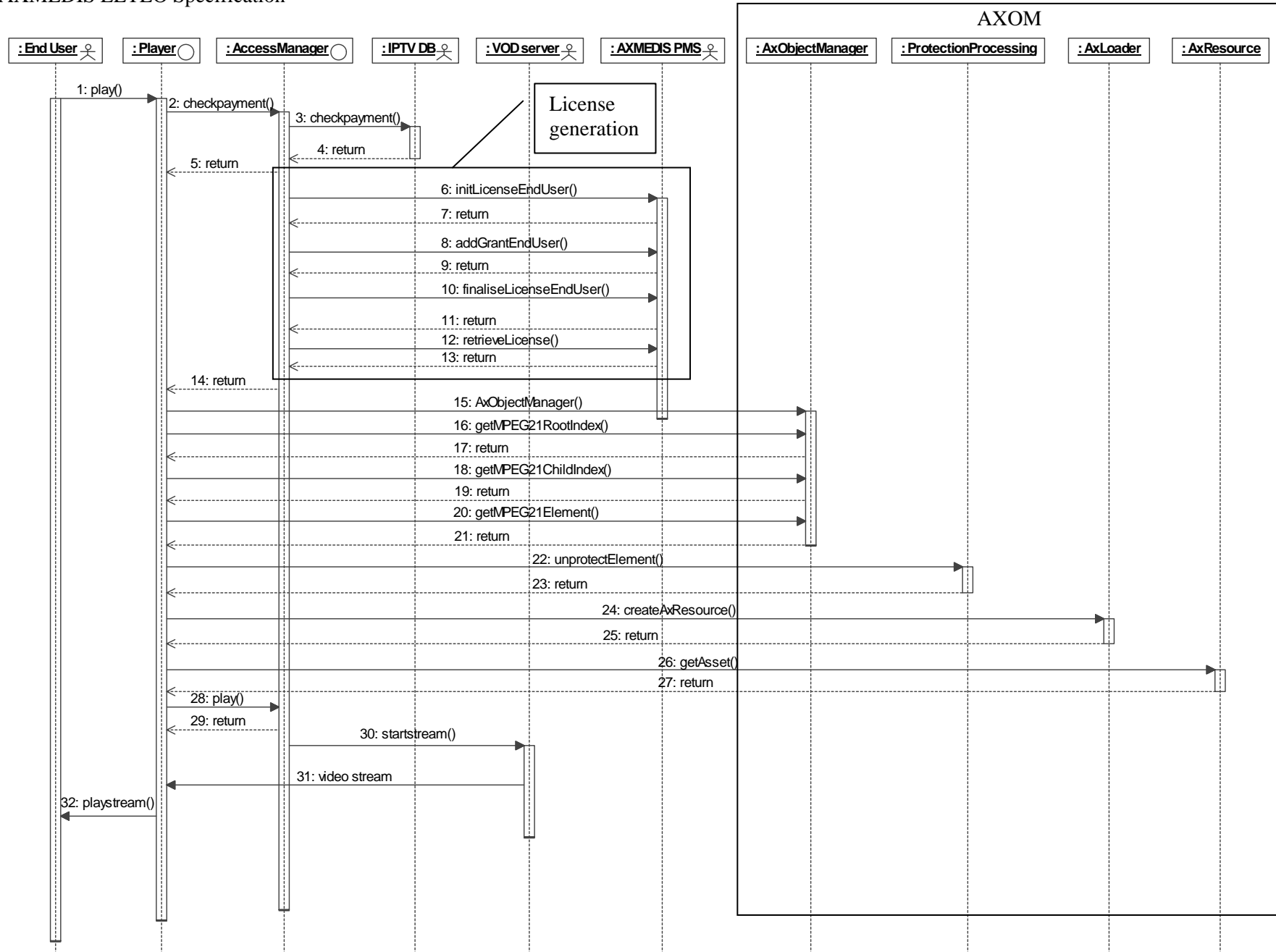
3.3.3 End User viewing TEO AXMEDIS content trailers scenario (Sequence diagram)



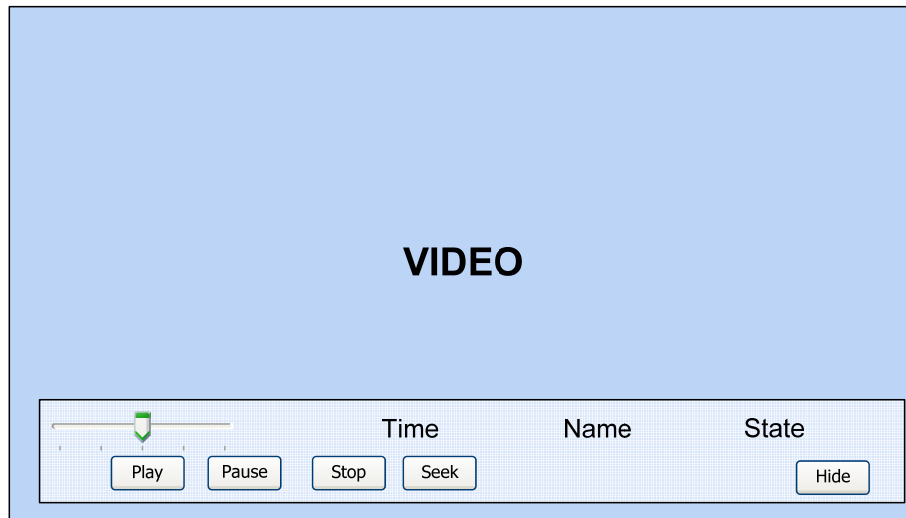
When End User selects Play trailer option his interface switches to player mode. This interface mode and user capabilities are described more detail in sections 3.1.4-3.1.8.

3.3.4 End User buys TEO AXMEDIS content scenario (Sequence diagram)

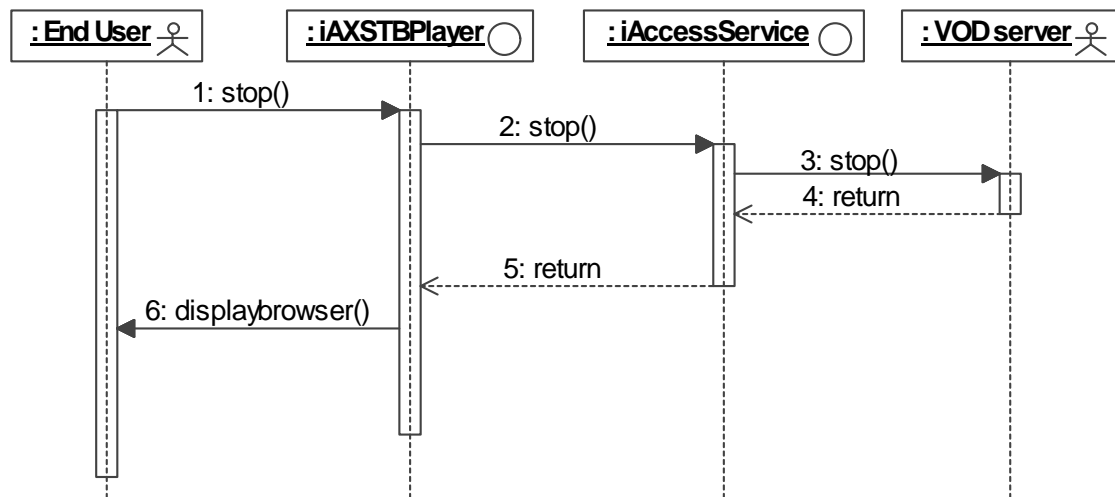
When End user selects an object, it can be purchased and played in AXSTB Player. If object is free of charge iAccessService selects required video object from database and calls VOD server to stream this object to End user. If object is chargable End User is prompted to enter PIN number as described in section 3.1.12. All actions with AXMEDIS objects are logged. After pressing Play (or Play trailer) button user interface switches to player mode. The entire user interface screen is filled with video.



In player mode End user can select the option to show video player toolbox and status bar. In this status bar End user can see the short information (duration, play time, object name, status) about video object. Toolbox gives End user the possibility to play, pause, stop or seek video. User can perform all these actions by pressing the corresponding remote control button. If user doesn't want to see toolbox and status bar, he can hide them by pressing the corresponding button.

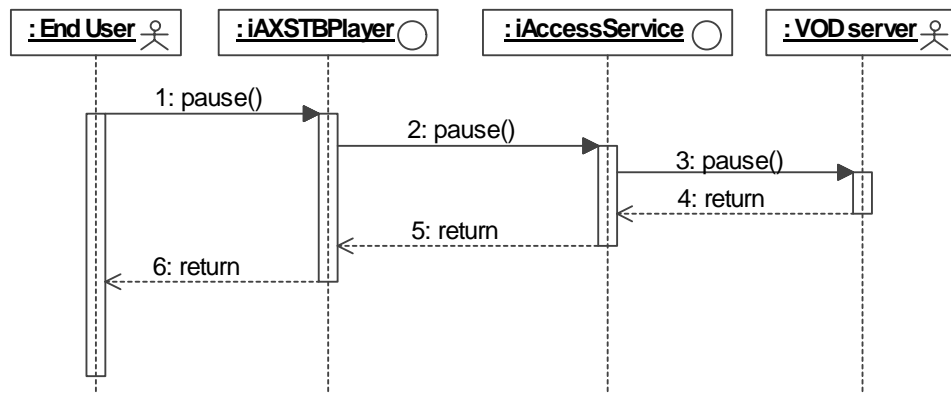


3.3.5 End User stopping TEO AXMEDIS content scenario (Sequence diagram)

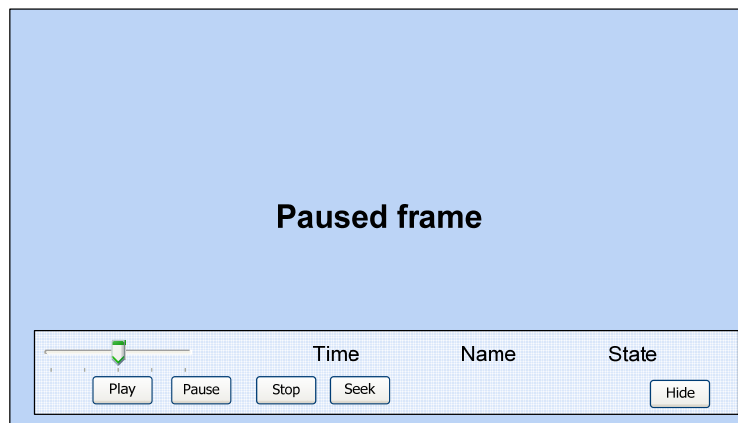


When End User presses Stop button in AXSTB Player, VOD server stops video streaming to STB. Every time when any operation with video object is performed, AXDB updates its log. When video streaming is stopped, end user interface switches to browser mode, which is described in section 3.1.2.

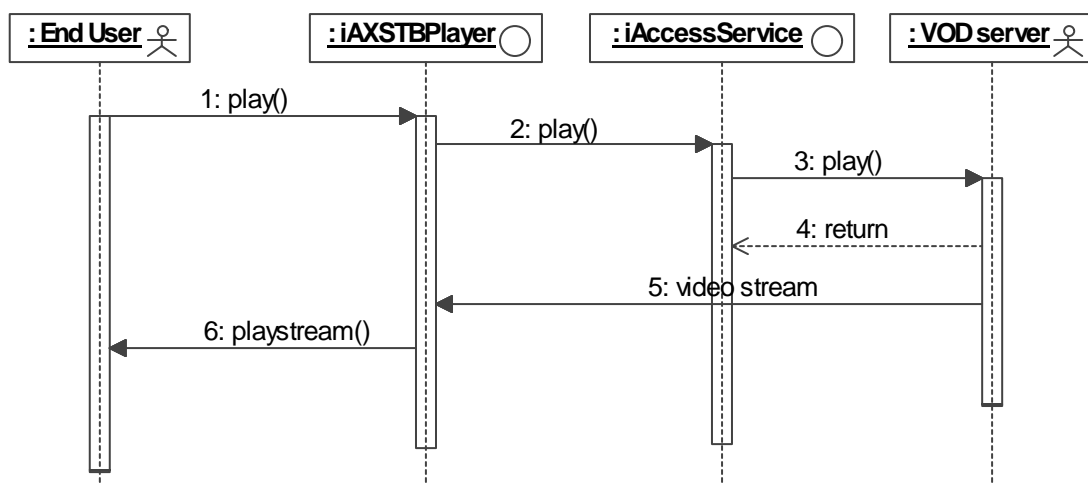
3.3.6 End User pausing TEO AXMEDIS content scenario (Sequence diagram)



When End user views video object, which is streamed from VOD server, he can pause playing at any time. After pressing Pause button, VOD server suspends video streaming and End user views the last frame in his player. In this time End user can also show/hide player toolbox and status bar area. AXMEDIS DRM can restrict pausing video stream. In this case Pause button will be inactive.

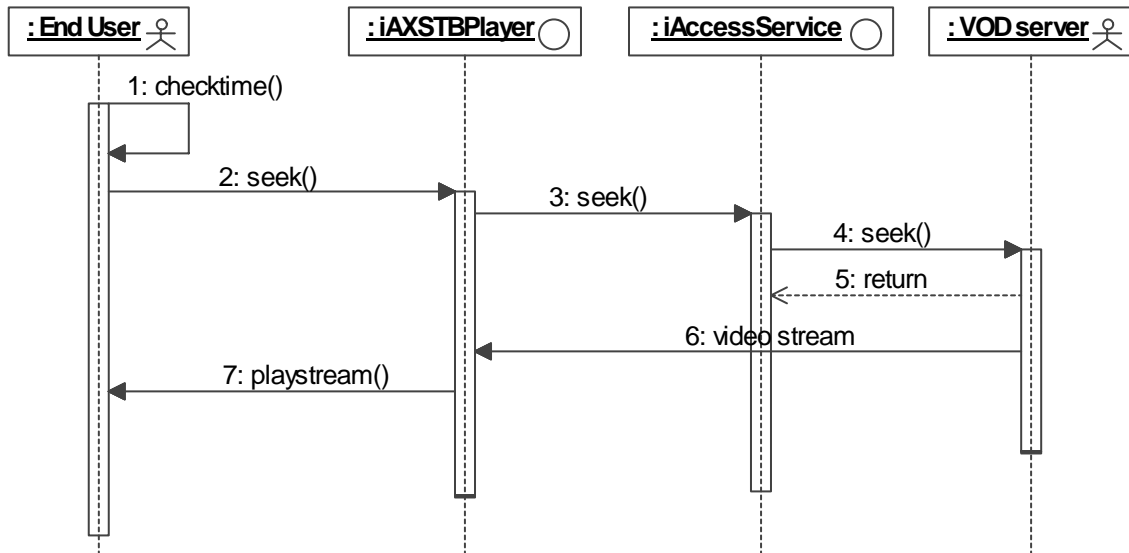


3.3.7 End User continue playing TEO AXMEDIS content scenario (Sequence diagram)

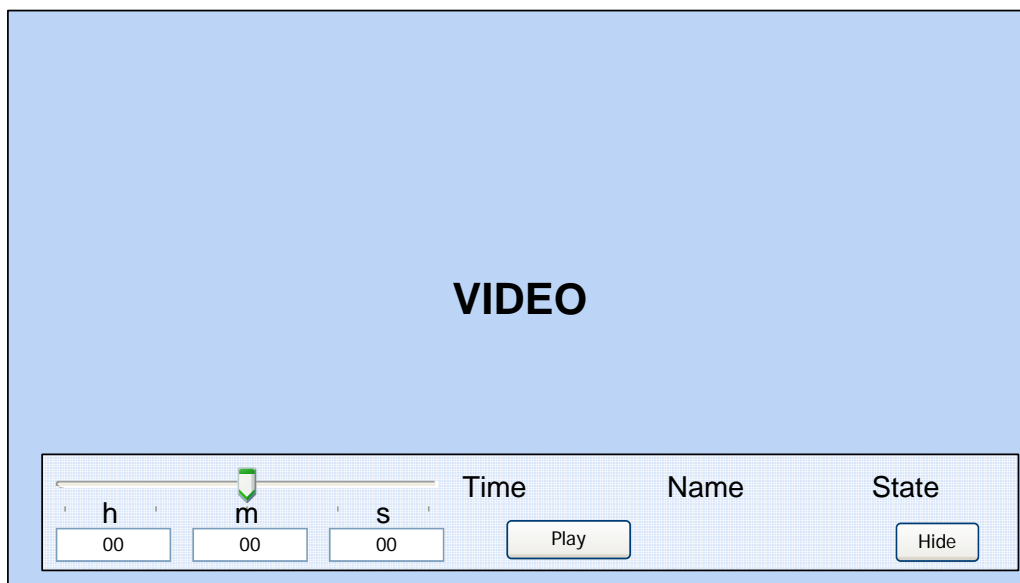


In AXSTB Player End user can continue playing video after pausing. In this case, End user must press Play button and VOD server continues video streaming. As mentioned before, AXMEDIS DRM can restrict pausing video object. In this case Play button will also be inactive.

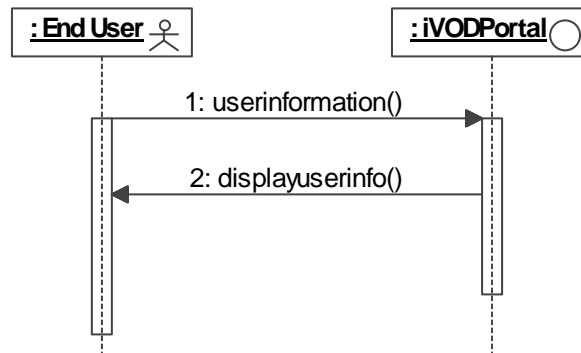
3.3.8 End User seeking TEO AXMEDIS content scenario (Sequence diagram)



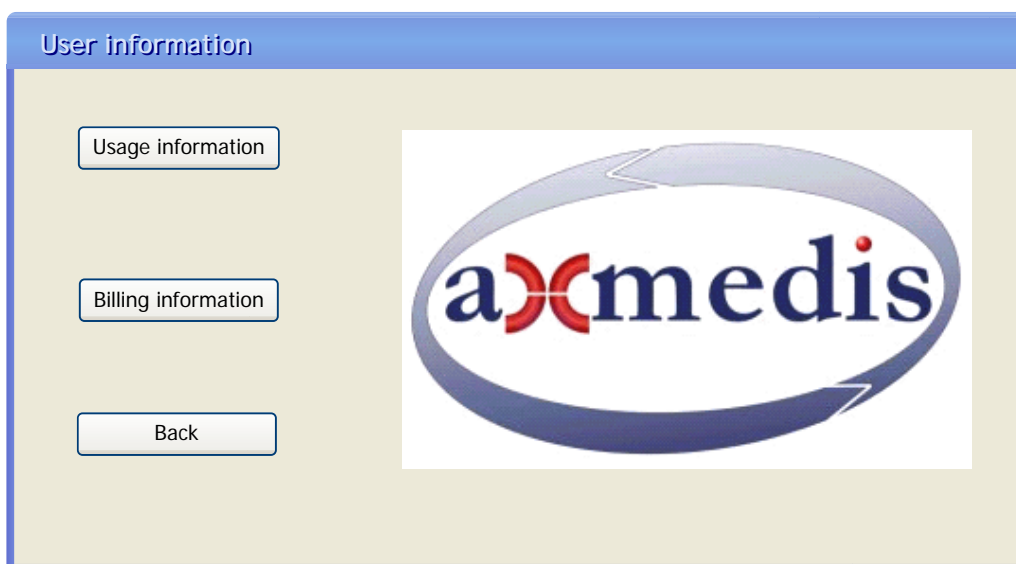
TEO VOD server ensures possibility for End user to reach desirable point of video at any time. End user can enter required time and from that moment VOD server starts video streaming. If End user presses Seek button, seek toolbox appears at the bottom of the screen. In this toolbox End user can enter desirable start time of video and press Play button to jump to the entered time. End user also can use the slider to change video object time. If End user changes his mind and wants to continue viewing video, he can hide seek toolbox.



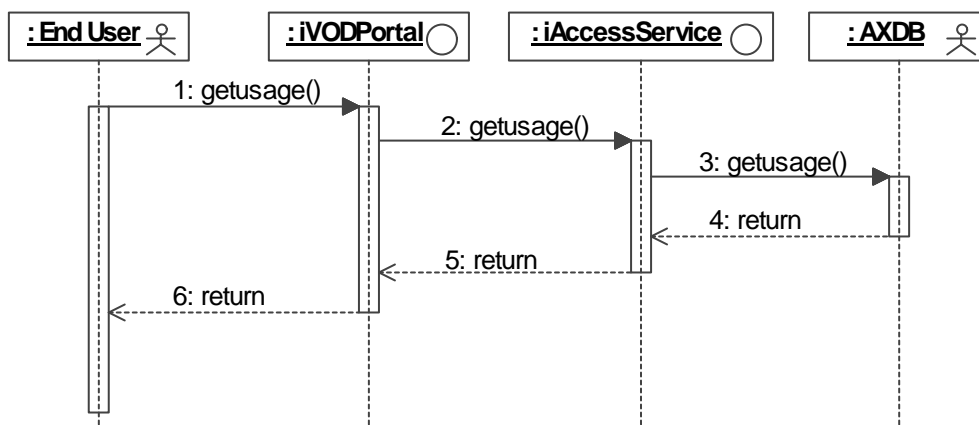
3.3.9 End User viewing his information scenario (Sequence diagram)



In TEO AXMEDIS system End user can see its usage and billing information in his TV screen. If End user wants to see required information, he must press View information button in TEO AXMEDIS home page. Afterwards End user screen switches to user information window and End user can select which personal information he wants to see – usage or billing information. In this window End user can go back to previous window.



3.3.10 End User viewing his usage information scenario (Sequence diagram)

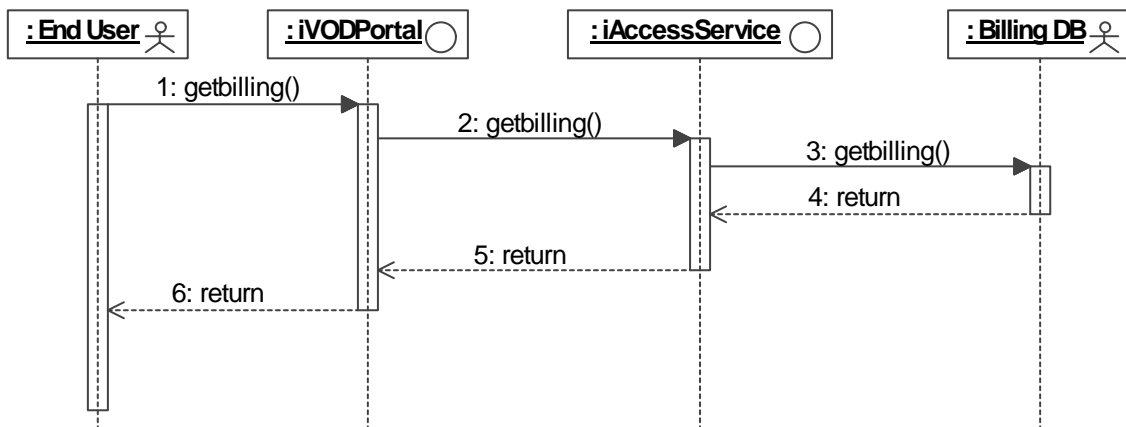


End user usage information is stored in AXDB. Every time when End user plays/stops/selects or do other actions with AXMEDIS content, event log is updated. Thus End user can see various information about his usage. A possible usage information window prototype is presented below. End user can see list of viewed objects, viewing dates/times, viewing counts etc.

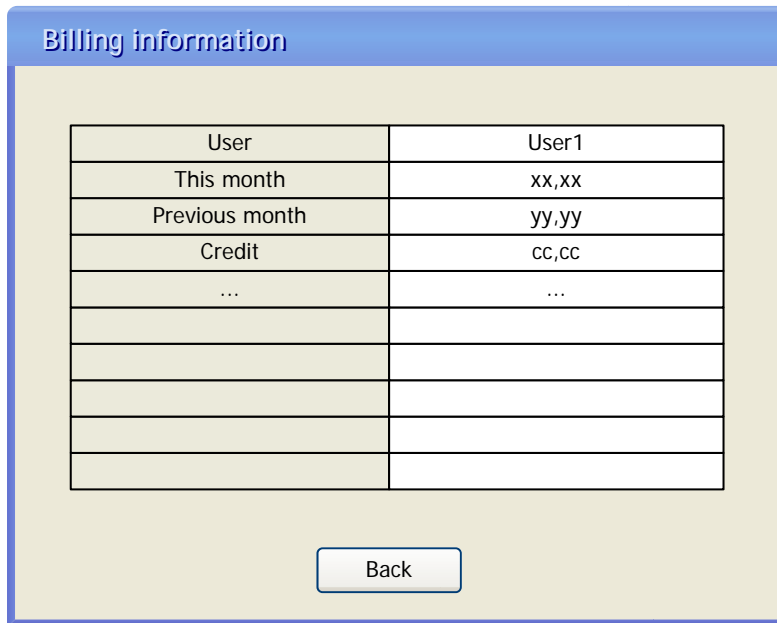
Usage information

Object	Last view	View count	...
Object1	2007 01 02	2	
Object2	2007 01 03	1	
Object3	2007 01 04	2	

3.3.11 End User viewing his billing information scenario (Sequence diagram)



Most of AXMEDIS objects will be chargeable and End user must pay for them. At any time End user can select view billing information option to view information about his billing. The prototype of billing information window is presented below. In this window End user can see the amount of money he has spent this month on watching AXMEDIS content, the bill for previous month and other billing information.

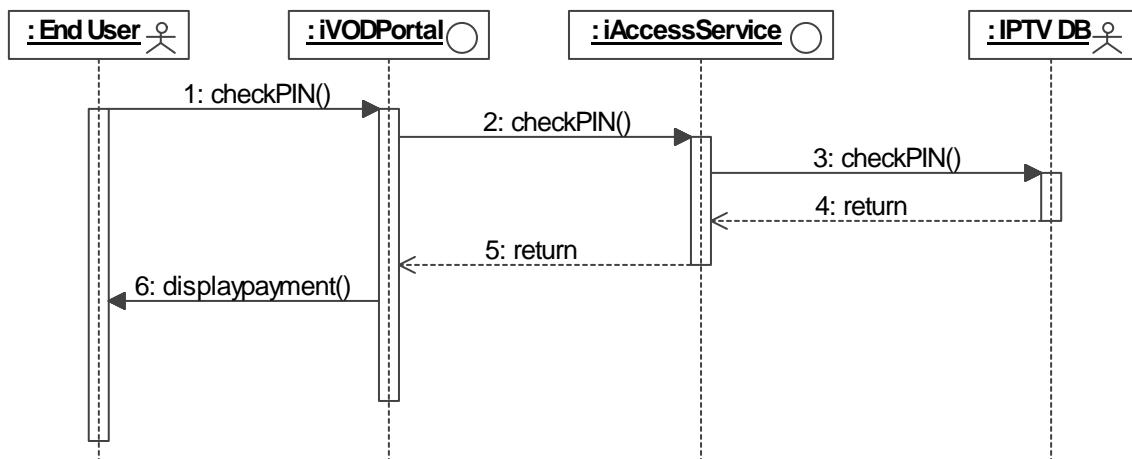


Billing information

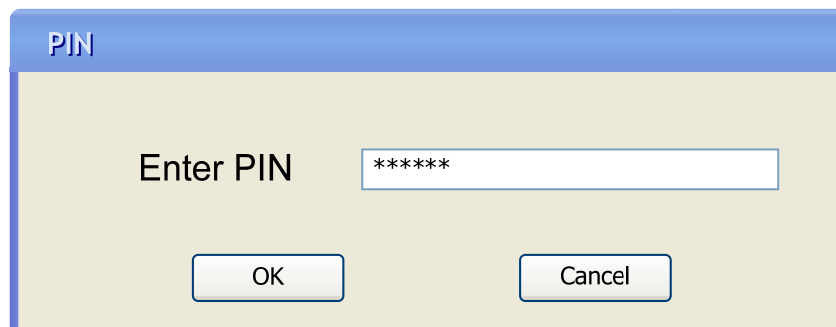
User	User1
This month	xx,xx
Previous month	yy.yy
Credit	cc,cc
...	...

Back

3.3.12 End User entering his PIN scenario (Sequence diagram)



When End User chooses to play chargeable AXMEDIS content, by pressing a Play button, he will be prompted to enter a PIN, which is provided to End user when he signs a contract for TEO IPTV services. By entering his PIN End user agrees to pay the required amount of money for viewing selected content. If entered PIN is correct End user interface switches to window, which asks about the payment method.

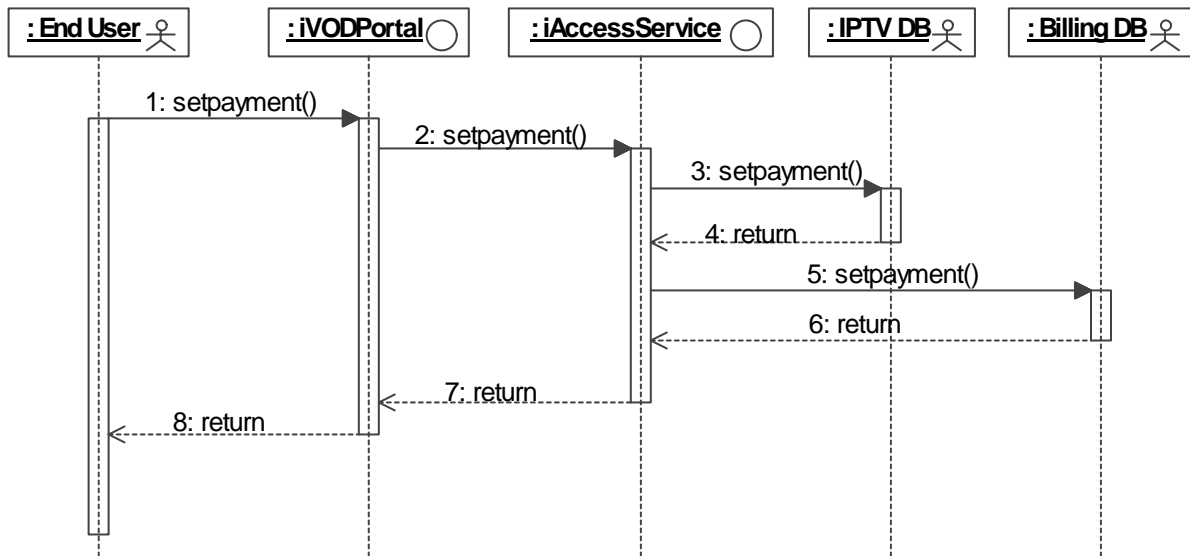


PIN

Enter PIN

OK Cancel

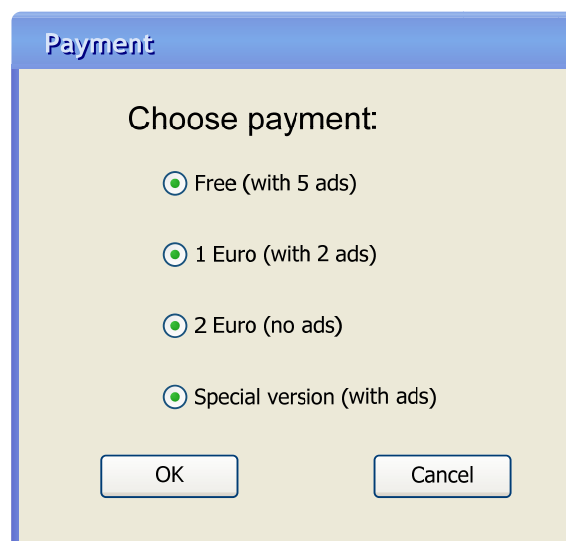
3.3.13 End User choosing his payment method scenario (Sequence diagram)



Most of AXMEDIS video objects used in TEO AXMEDIS network will be chargeable and End user will pay for them. Before the user views a piece of video content on demand (or at the point of replay), he is given the choice to choose from different pricing models or other incentives to watch this content. These could include:

- Watch for free, but also view 5' non-skippable ads embedded into a piece;
- Pay EUR 1 but watch 2' non-skippable ads before the start of the programme;
- Pay EUR 2 and watch the movie with no ads;
- Buy a “special” version of a programme (such as the director’s cut of a movie) by agreeing to watch non skippable adverts.

End user interface window, in which user can select one of payment choices is presented below. After selecting the payment method and pressing OK button, End user can view selected video object in his TV.



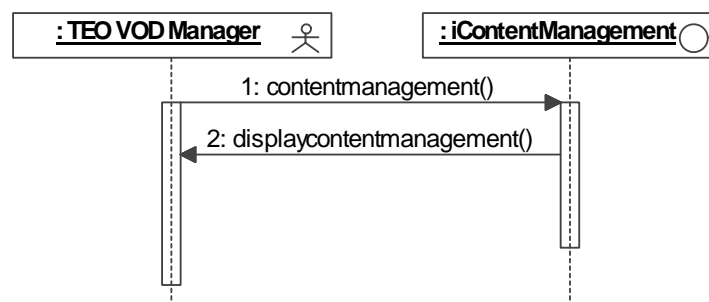
3.4 TEO VOD Manager scenarios (Sequence diagrams)

TEO VOD Manager (employed by TEO) is responsible for content management in TEO AXMEDIS IPTV network. This actor downloads AXMEDIS objects from global AXMEDIS network via AXEPTool and prepares them for streaming in VOD server. TEO VOD Manager downloads AXMEDIS objects at his own discretion or business model. The key roles of TEO VOD Manager are content and user management.

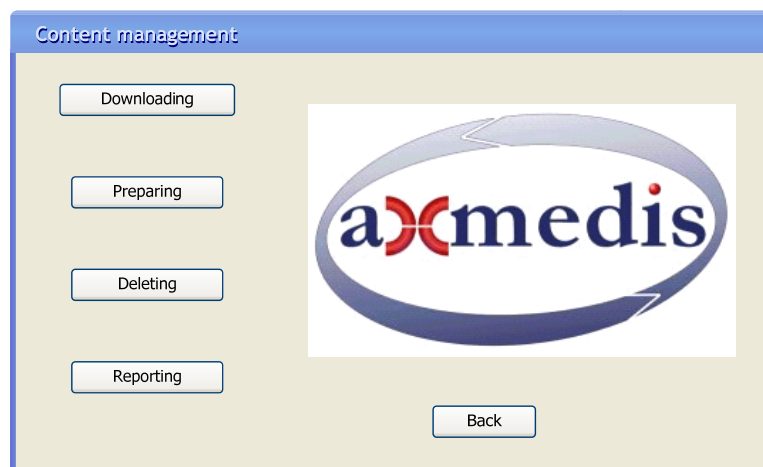
In content management role TEO VOD Manager will use AXEPTools, provided by AXMEDIS Framework and select and download content from global AXMEDIS Databases. The main job of TEO VOD Manager will be to prepare AXMEDIS object for streaming from VOD Server. To do this TEO VOD Manager must save AXMEDIS Object in two locations: VOD Server and AXDB.

TEO VOD Manager can delete, add and change AXMEDIS object in VOD server and AXDB by using his content management tools. Another function, which will be implemented in management tool, is reporting, where TEO VOD Manager can generate report about object usage.

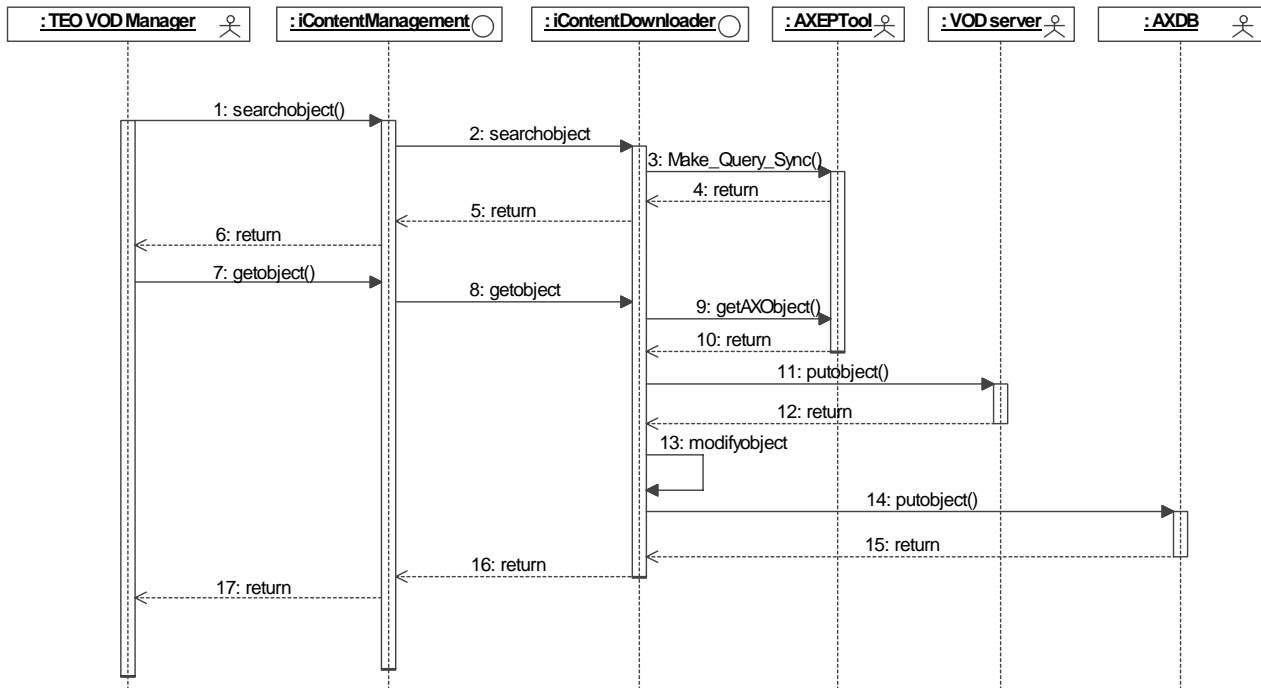
3.4.1 TEO VOD Manager content management scenario (Sequence diagram)



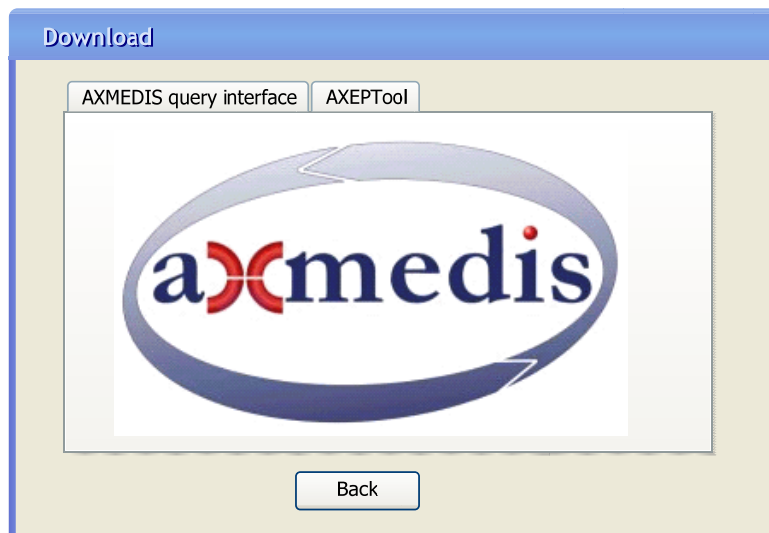
TEO VOD Manager is responsible for content management in TEO AXMEDIS IPTV network. When he connects to TEO Content & User Management Portal he can select content management option and interface switches to the window, which prototype is presented below. In this window TEO VOD Manager can select various content management operations. He can select downloading AXMEDIS object from global AXMEDIS network option, preparing content for streaming from VOD server option, content deleting option or content usage reporting option. In this window TEO VOD Manager can also press Back button and return to the previous window.



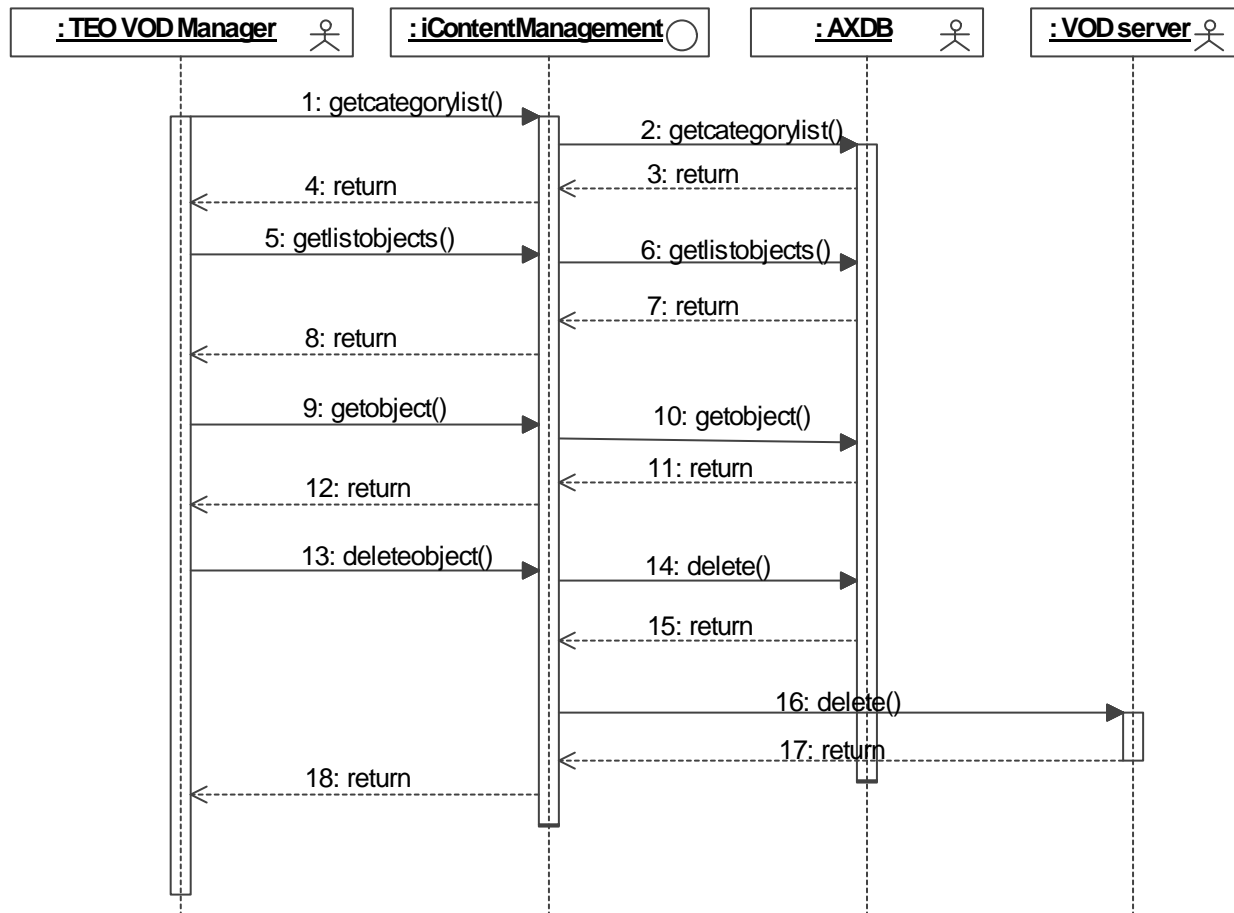
3.4.2 TEO VOD Manager downloading AXMEDIS content and placing into VOD Server and AXDB (Sequence diagram)



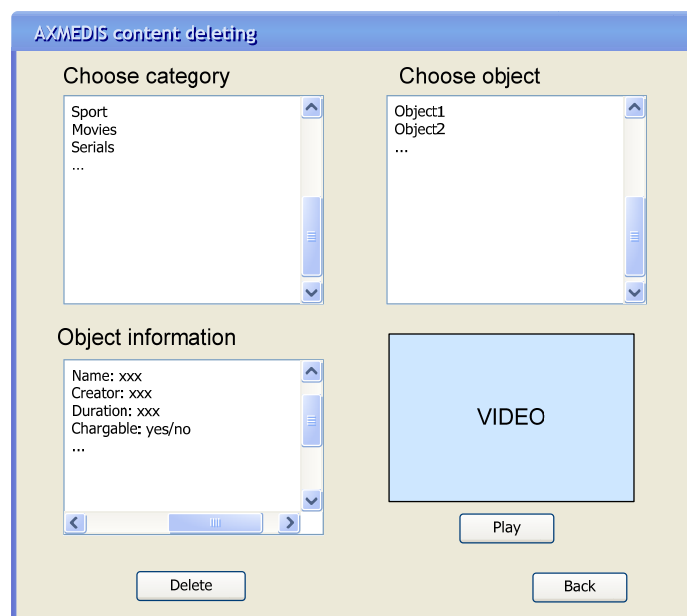
In content management system TEO VOD Manager can download AXMEDIS object from global AXMEDIS network. In this case, AXMEDIS query is used for searching for the required object. When the object is found it can be downloaded by using AXEPTTool. AXMEDIS query and AXEPTTool are tools which will be used as they are. Downloaded object will be saved in local AXDB and VOD Server.



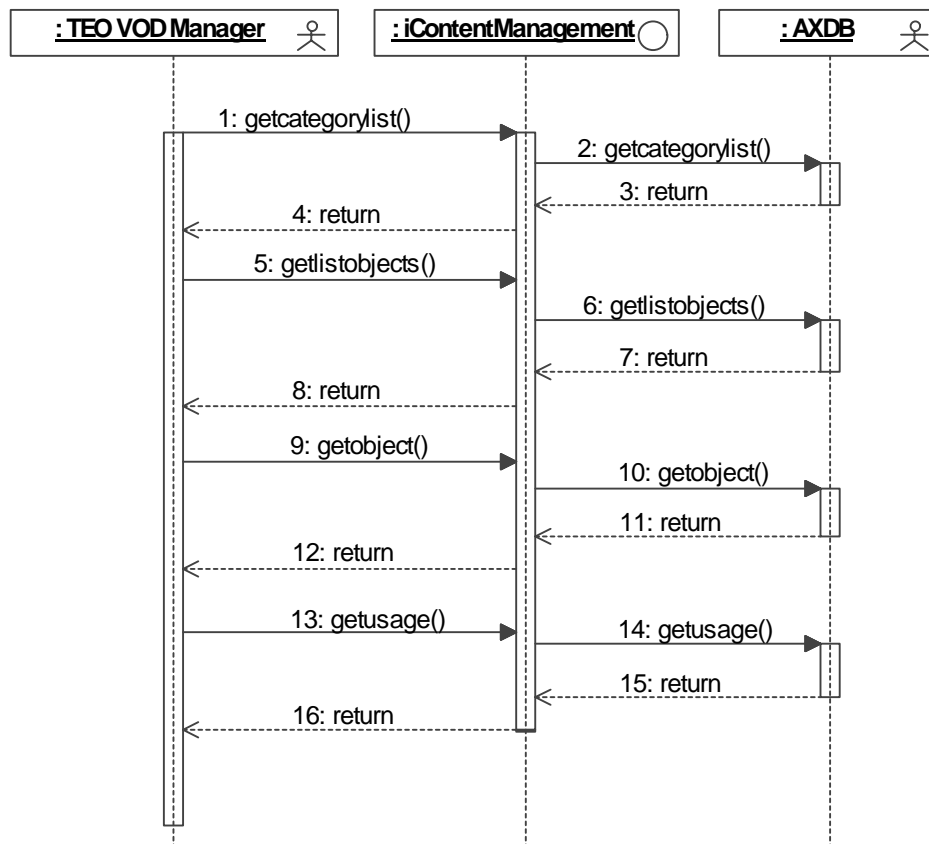
3.4.3 TEO VOD Manager content deleting scenario (Sequence diagram)



TEO VOD Manager can delete required object. In AXMEDIS content for deleting window, TEO VOD Manager must select object or object group and press the Delete button. In order to avoid accidental deleting manager can be warned by the adequate pop-up message. In AXMEDIS content deleting window manager can see the object description and play video data.



3.4.4 TEO VOD Manager content reporting scenario (Sequence diagram)



TEO VOD Manager can view content usage information. He can select Reporting option and make the required report. In this report TEO VOD Manager can see the information about usage of various objects or object groups. These reports can be extended according to user requirements.

AXMEDIS content deleting

Choose category

Sport
Movies
Serials
...

Choose object

Object1
Object2
...

Object information

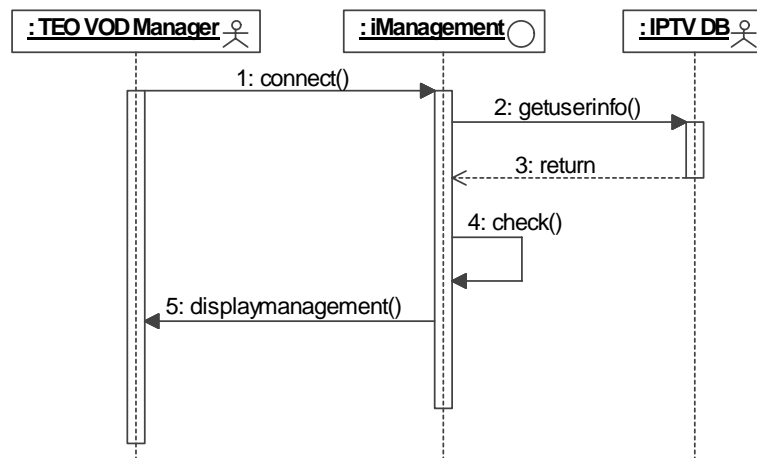
Name: xxx
Creator: xxx
Duration: xxx
Chargable: yes/no
...

Object	Last view	View count	Earnings	...
Object1	2007 01 02	2	1	
Object2	2007 01 03	1	2	
Object3	2007 01 04	2	2	

Get report

Back

3.4.5 TEO VOD Manager connecting scenario (Sequence diagram)



In order to perform any actions in TEO Content & User Management Portal, the manager must be connected. In Connect window TEO VOD Manager must enter his user name and password. Management system compares entered data with data in IPTV DB and if entered data matches information about the user from IPTV DB the user will be connected to the system.

The 'Connect' dialog box is used for authentication. It features a title bar labeled 'Connect'. Inside, there are two input fields: 'User name' with a placeholder 'Enter user name' and 'Password' with a placeholder 'Enter password'. At the bottom, there are two buttons: 'OK' and 'Cancel'.

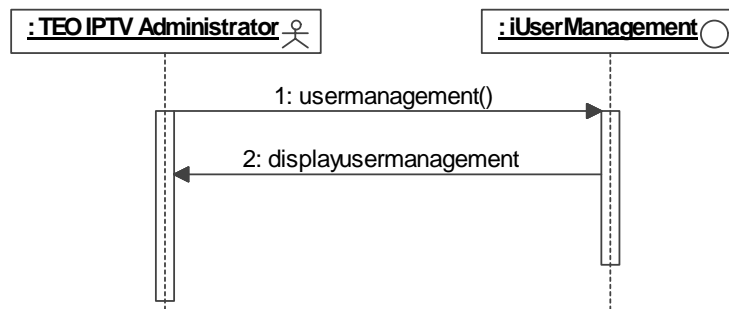
When connected TEO VOD Manager can select Content management option. He can also exit TEO Content & User Management Portal by pressing Exit button.

The main menu window is titled 'AXMEDIS'. It contains two buttons on the left: 'Content management' and 'Exit'. On the right side, there is a large logo for 'axmedis' with a stylized 'a' and 'c' in red and blue, and a circular arrow graphic around it.

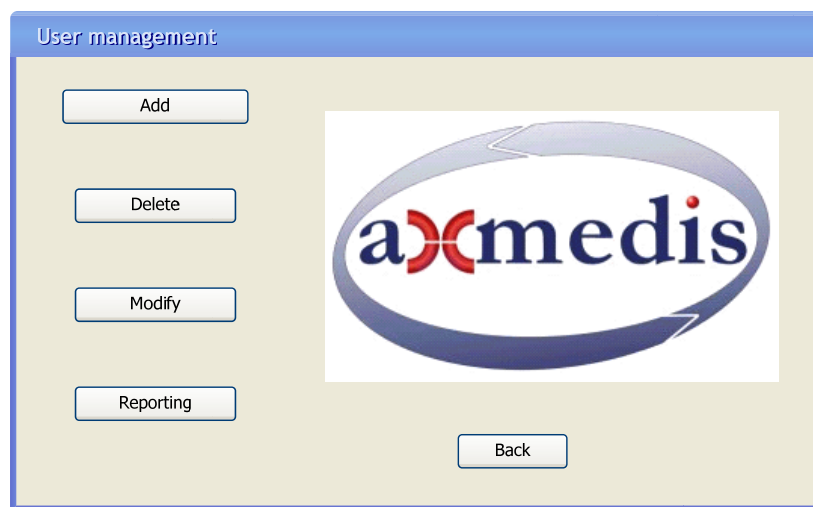
3.5 TEO IPTV Administrator scenarios (Sequence diagrams)

TEO IPTV Administrator (employed by TEO) is responsible for user management in TEO IPTV network. In user management role TEO IPTV Administrator can add, delete and modify properties of TEO IPTV AXMEDIS users. There will be a possibility to make and view reports about user actions and billing information from AXDB and Billing DB.

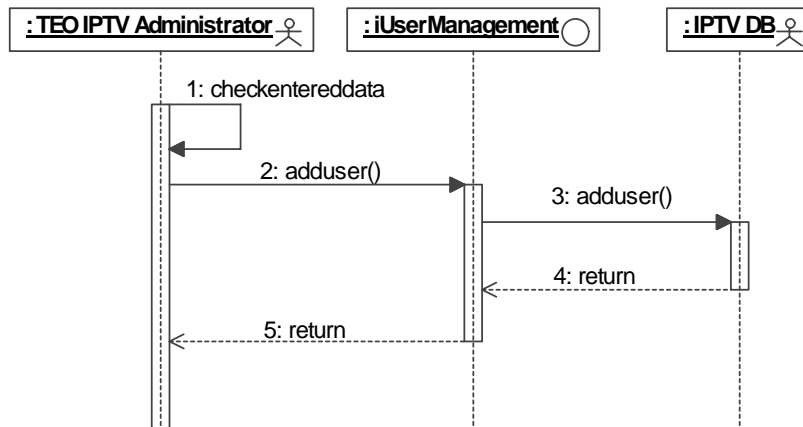
3.5.1 TEO IPTV Administrator user management scenario (Sequence diagram)



When manager connects to management system he can select User management option and then User management window is displayed. In this window TEO IPTV Administrator can choose user adding, deleting, modifying or reporting options.



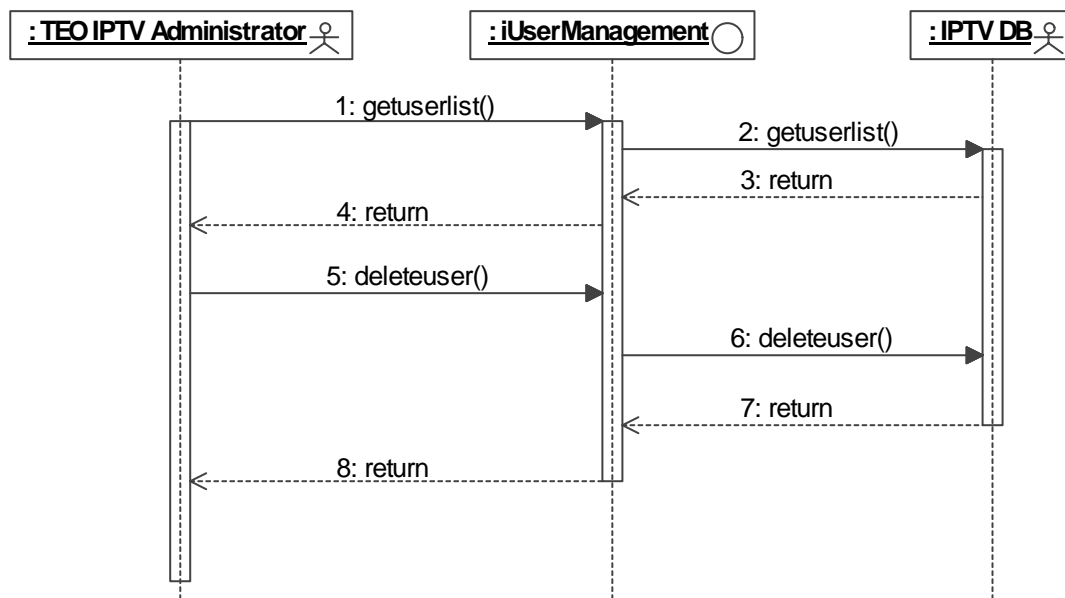
3.5.2 TEO IPTV Administrator user adding scenario (Sequence diagram)



In user adding scenario TEO IPTV Administrator can add a new user. This user must be GALA TV user. In this case, TEO IPTV Administrator grants the access to AXMEDIS content to the new user.

Add user	
First name	aaa
Last name	bbb
Start date	2007 01 01
End date	2008 01 01
Axmedis	yes
...	...

3.5.3 TEO IPTV Administrator user deleting scenario (Sequence diagram)



In user deleting scenario TEO IPTV Administrator deletes required user from TEO AXMEDIS IPTV network. TEO IPTV Administrator is responsible only for TEO AXMEDIS IPTV users. In order to delete the user, manager must select required user and press Delete button. In order to avoid accidental user deleting from TEO AXMEDIS IPTV network manager can be warned by adequate pop-up message.

Delete user

Select user

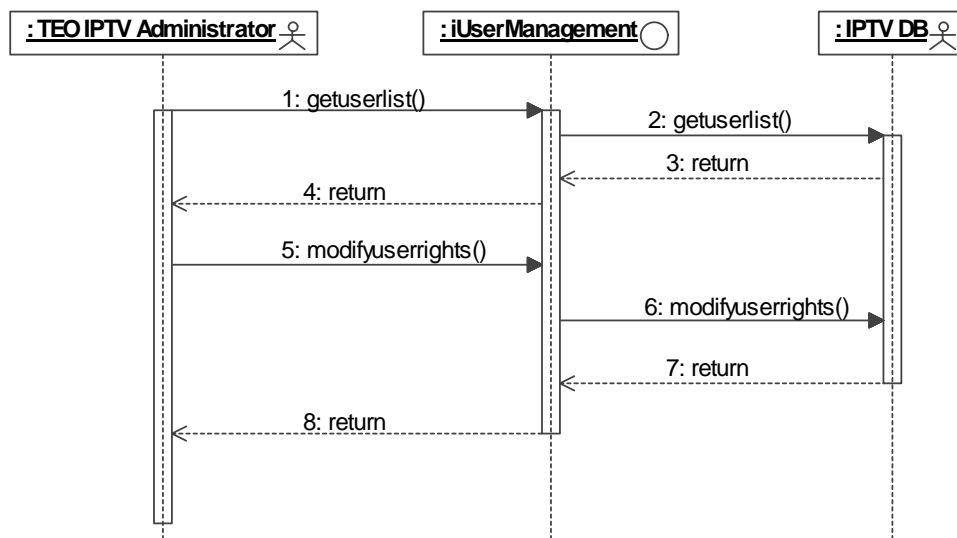
User1
User2
User3
...

User information

First name	aaa
Last name	bbb
Start date	2007 01 01
End date	2008 01 01
Axmedis	yes
...	...

Delete
Back

3.5.4 TEO IPTV Administrator user rights modification scenario (Sequence diagram)



In user rights modification scenario TEO IPTV Administrator modifies required user rights. Manager can modify user information and his rights to view AXMEDIS object.

User rights

Select user

User1
User2
User3
...

User information

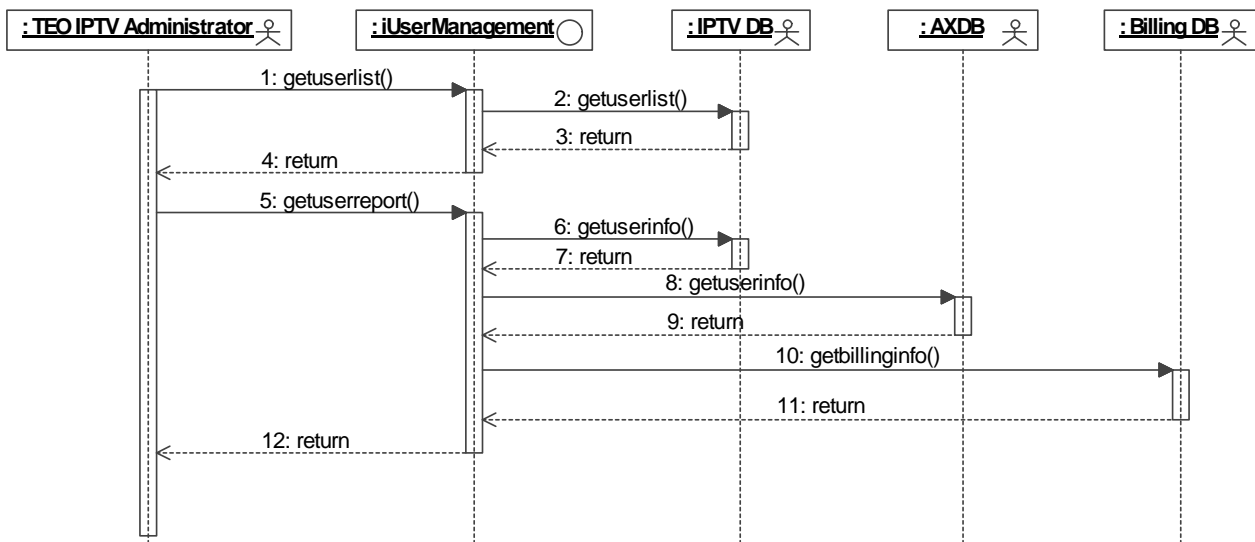
First name	aaa
Last name	bbb
Start date	2007 01 01
End date	2008 01 01
Axmedis	yes
...	...

User rights

Enabled	yes
Axmedis	yes
Start date	2007 01 01
End date	2008 01 01
Special content	yes
...	...

Update
Back

3.5.5 TEO IPTV Administrator user reporting scenario (Sequence diagram)



To get information about users actions in TEO AXMEDIS IPTV network, TEO IPTV Administrator can select user reporting option. In user reporting scenario TEO IPTV Administrator can select the user to view his personal information, usage information and billing information. These reports can be extended according to users requirements.

User rights

Select user

User1
User2
User3
...

User information

First name	aaa
Last name	bbb
Start date	2007 01 01
End date	2008 01 01
Axmedis	yes
...	...

Usage information

Movies	x
Sport	x
Music	x
Total	y
Duration	ii
...	...

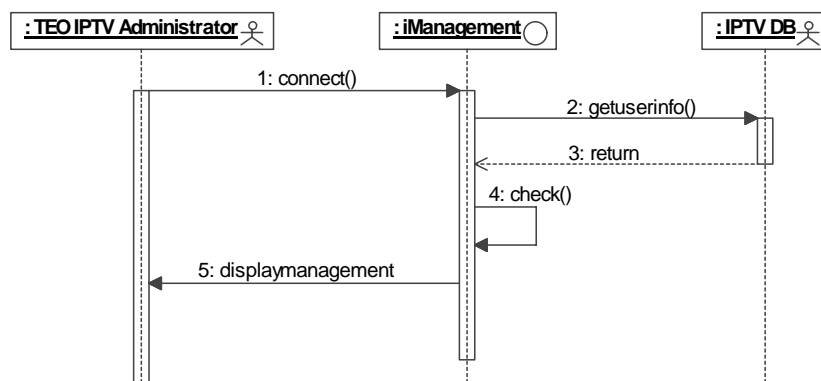
Billing information

Current	x
Previous month	xx
This year	xx
..	...

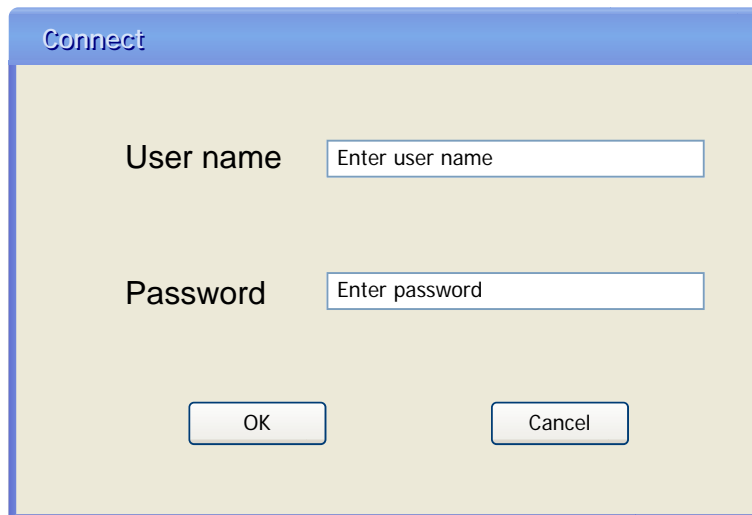
Get report

Back

3.5.6 TEO IPTV Administrator connecting scenario (Sequence diagram)



In order to perform any actions in TEO Content & User Management Portal, the manager must be connected. In Connect window TEO IPTV Administrator must enter his user name and password. Management system compares entered data with data in IPTV DB and if entered data matches information about the user from IPTV DB the user will be connected to the system.



The image shows a 'Connect' dialog box with a blue title bar. Inside, there are two input fields: 'User name' with a placeholder 'Enter user name' and 'Password' with a placeholder 'Enter password'. At the bottom, there are two buttons: 'OK' and 'Cancel'.

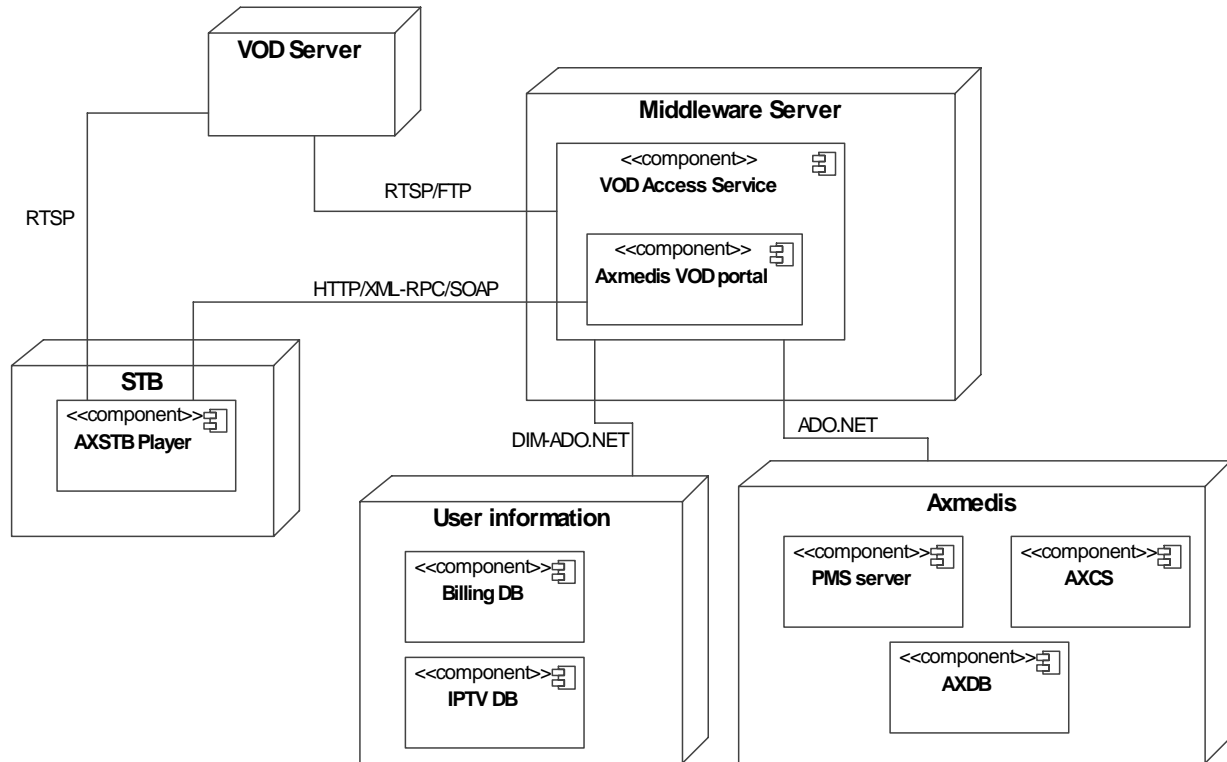
When connected TEO IPTV Administrator can select User management option. He can also exit TEO Content & User Management Portal by pressing Exit button.



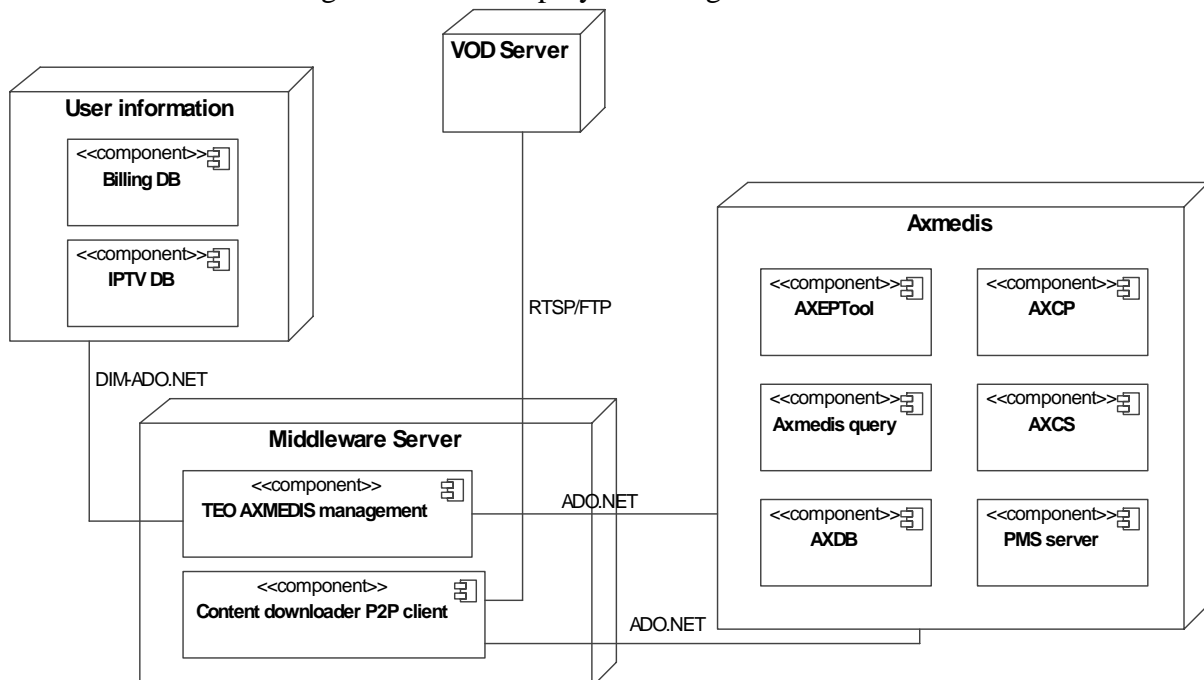
3.6 Implementation diagrams

The implementation strategy for AXSTB Player and VOD Access Service is specified in AXSTB Player and VOD access service deployment diagram. AXSTB Player will communicate with VOD Server using RTSP protocol. This protocol will be used for communication between VOD Access Service and VOD Server. VOD Access Service will be implemented in middleware server. VOD Access Service will communicate with user information databases (Billing DB and IPTV DB) by using DIM-ADO.NET protocol. ADO.NET protocol will be used for communication between VOD Access Service and AXMEDIS tools (PMS Server, AXCS, AXDB). AXSTB Player will use HTTP/XML-RPC/SOAP protocols for communicating with VOD Access Service.

AXSTB Player and VOD Access Service deployment diagram:



TEO Content & User Management Portal is used for content and user management by TEO IPTV Administrator and TEO VOD Manager. The implementation strategy for TEO Content & User Management Portal is specified in TEO Content & User Management Portal deployment diagram below. TEO Content & User Management Portal will be implemented in a middleware server. Content downloader P2P client will be used for placing downloaded AXMEDIS object by using AXEPTool to VOD server and by updating AXMEDIS object metadata and by placing metadata and stream information into AXDB. This component is used in AXMEDIS Content & User Management Portal module and will be used by TEO VOD Manager for content management. TEO Content & User Management Portal deployment diagram:



3.7 General architecture and relationships among the specified modules (Elion)

3.7.1 General architecture of video on demand distribution for PC (ELION)

ELION distributor basic structure is presented below. The distributor is made out of an integration module ELAXAPI_WS that is used as plug-in for ELION's HOT.EE community portal to enable content listing and usage. E-Pass account and payment gateway is used with HOT.EE as standard method of user profiling, authentication and charging users for commercial content exploitation. Payment methods as well as authentication and communication between hot.ee and epass are standardized by payment channel type or according to any other standardization demands. Both HOT.EE and epass.elion.ee are existing services that have been in use for extensive time. For Axmedis exploitation only hot.ee will have an extended interface scripts to provide end user interface for ELAXAPI_WS operations, e-pass does not need any extending or changing for AXMEDIS exploitation.

The existing component list is:

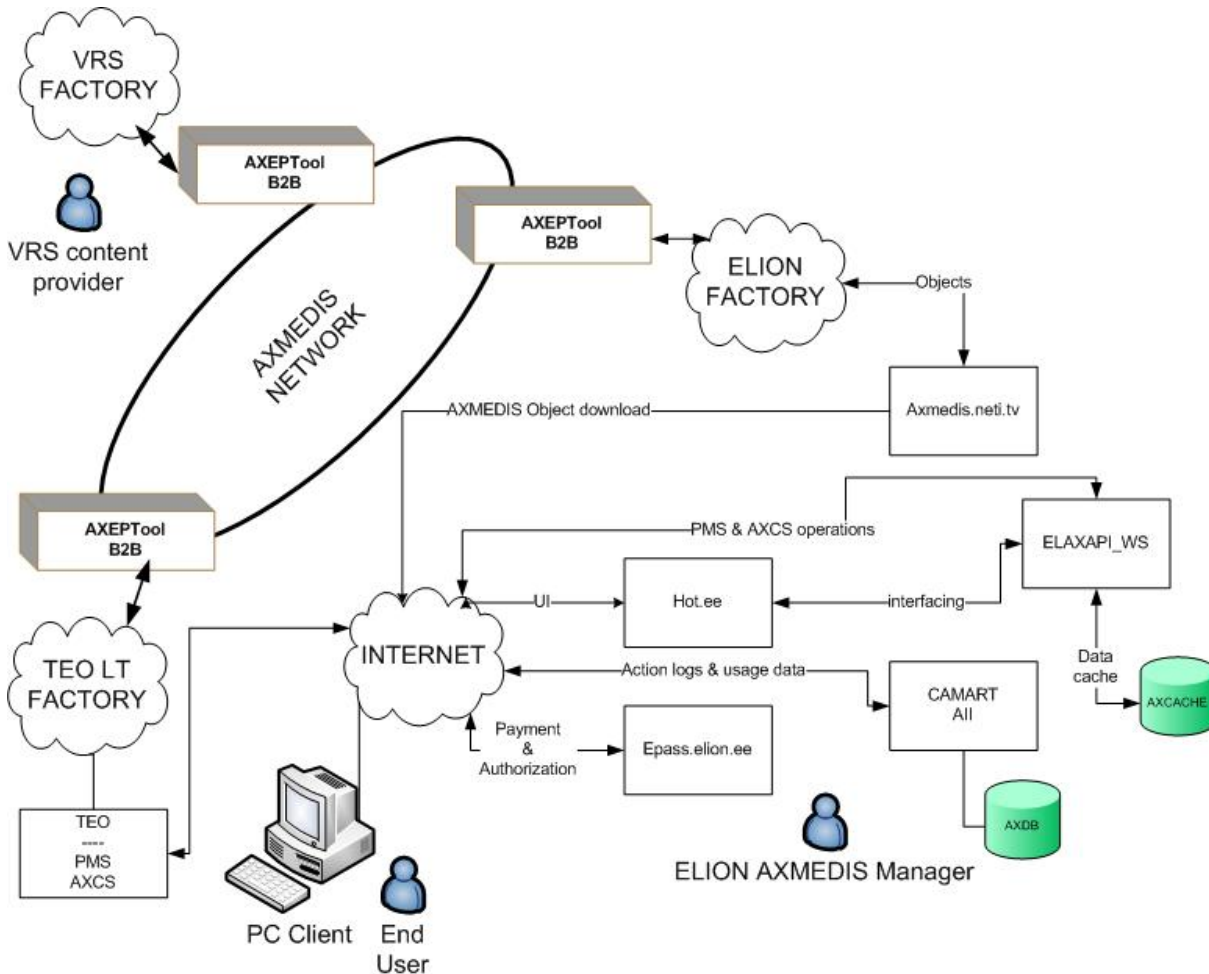
- E-pass account services & user billing
- Hot.ee Community & Communication portal
- Server & network infrastructure

Software to be developed for interfacing with AXMEDIS:

- ELAXAPI_WS web service for interfacing between hot.ee and AXMEDIS tools.

Newly developed or existing AXMEDIS framework components:

- AXDB (existing)
- CAMART (existing)
- AII (existing)
- AXCS (existing)
- PMS (existing)
- AXEPTool B2B (existing)
- AXCP (existing)
- AXPCPLAYER (existing)
- AXMEDIS Editor (existing)



ELION AXMEDIS demonstrator includes:

- Axmedis.neti.tv virtual sever node including:
 - AXEPTool B2B node for content download and sharing on AXMEDIS B2B Network
 - ELAXAPI_WS interfacing web service for end user UI representation on hot.ee
 - IIS web folder share for AXMEDIS object downloads to end user
 - IIS web folder share to distribute AXMEDIS PC Player
 - CAMART, AII and AXDB installations for action log gathering and distribution
 - AXCACHE database for ELAXAPI_WS web service data store
 - SQL Server manager express for AXCACHE.
- [Www.hot.ee](http://www.hot.ee) portal – hot.ee portal is ELION’s existing proprietary community & communication portal that will be featuring additional menu option to enable AXMEDIS object distribution and marketing.
- Epass.elion.ee – epass is ELION’s proprietary account services and billing system serving as single shared resource between several solutions.
- ELION will use TEO PMS & AXCS servers.

ELION Manager will use axmedis.neti.tv node to manage AXMEDIS object distribution and statistics gathering.

3.7.2 End user scenarios

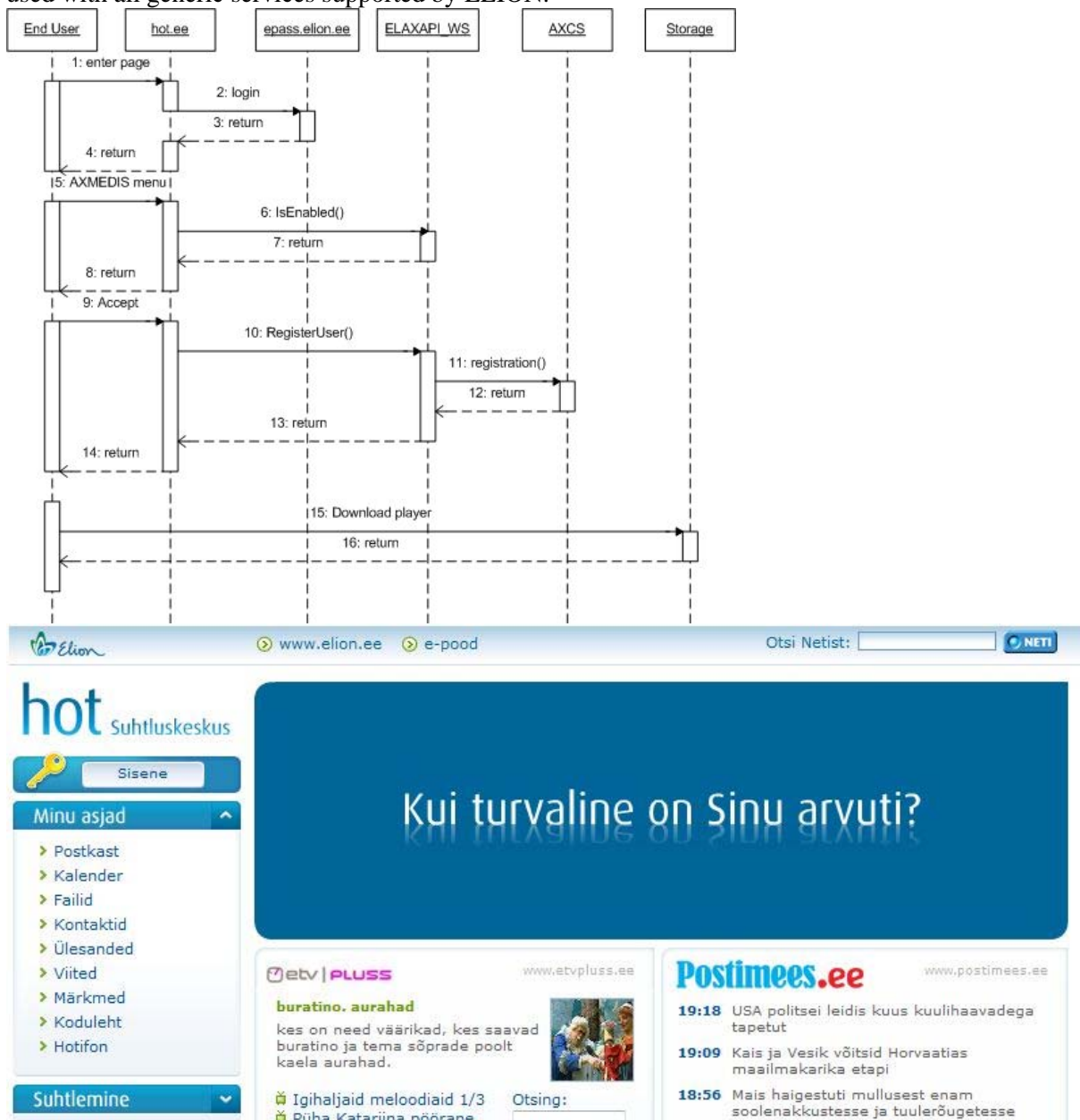
End User scenarios graphically describe End User use cases. Each sequence diagram corresponds to a particular use case. In this section End User interface prototypes are also presented. These prototypes are intended to describe the information and its position in a window, which is presented to End User, but they

do not represent final interface designs. All End User interfaces will be implemented in hot.ee manner and graphical view will be different from presented in this document.

3.7.2.1 Enable AXMEDIS service usage for the account

End user AXMEDIS registration is being done on demand and is processed only once for every unique epass account, as such the procedure will be shown in separate section to retain the separate common procedure from optional.

All hot.ee users must have epass account to be able to use portal services, the account is anonymous and everyone can register for it by following simple registration procedure, after the account is created it can be used with all generic services supported by ELION.



When End user types in the address www.hot.ee he arrives to home page where he can click the button "SISENE" shown in top left corner of the main navigation menu to enter epass authorization environment

Elion

www.elion.ee e-pood

Otsi Netist: NETI

e-pass

Sisenemine e-passi

Kasutajatunnus

Salasõna

Sisene **Tagasi**

- **Registreeri**
- Unustasid salasõna?
- Teenuste tellimine
- Sisene ID-kaardiga

➤ **E-pass** on Elioni poolt loodud interneti keskkondade autoriseerimise ja interneti sisuteenuseid vahendav süsteem.

➤ **Üks kasutajatunnus ja salasõna** - nendega saad lihtsalt ja turvaliselt siseneda erinevatesse e-passiga liitunud interneti keskkondadesse. [Loe lähemalt](#)

➤ **ETV Pluss** toote kasutamise reeglid ja kasutamiseks vajalikud sammud. [Loe lähemalt](#)

Elion

hot Suhtluskeskus

Andrius Lokotash

Seadid | Lahku

Minu asjad

Suhtlemine

Meelelahutus

- ETV pluss
- Mängukeskus
- MMS album
- Mobiil
- Telekava
- Axmedis

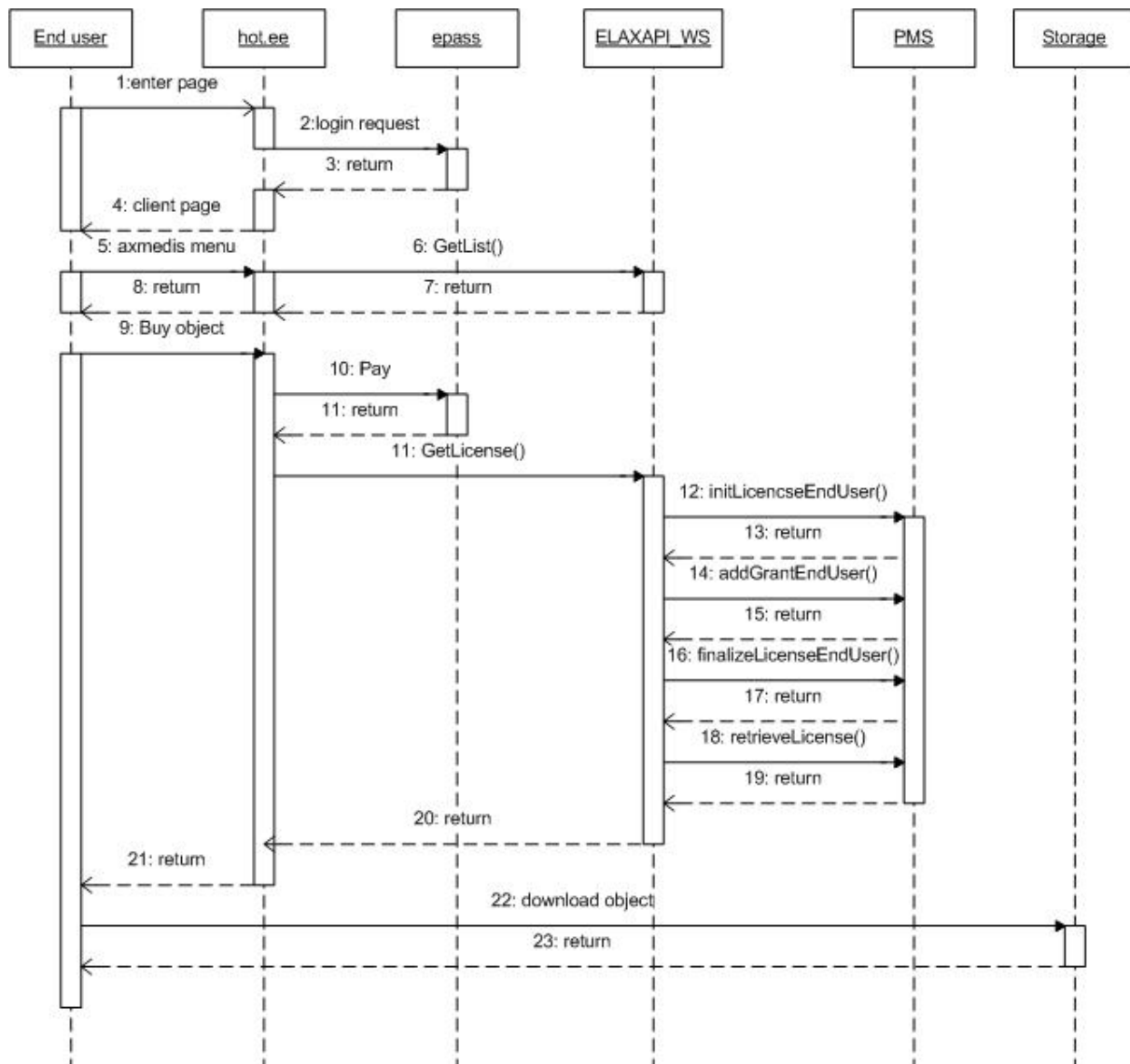
If user has registered then the registered username and password can be used for authorization, otherwise user must use the registration interface to make epass account.

After user has logged on the navigation menu will have AXMEDIS option under the entertainment menu.

If epass user has not been registered on AXMEDIS network a dialogue showing the service description and terms of use will be displayed to the end user with the option of accepting or declining the terms of use, if user declines the terms he is redirected to home page.

If user accepts the terms the user registration to AXCS is silently preformed on background and user is presented with AXMEDIS PC Player download link.

3.7.2.2 Main end user case



This scenario shows the common steps of common use after the account has been enabled to use AXMEDIS content.



After user has been authorized he can enter AXMEDIS menu. The page will present a listing of available files, the short description, price and links for purchasing the license and for object download.



After user clicks purchasing link he will be presented with a choice of payment channels and directed to that channel for making the payment. When payment is confirmed the system will generate the license for purchased object and user can download / use the content.

3.7.3 Manager scenario

Due to light weight of ELION AXMEDIS distributor the manager part of the system is in current state minimalistic, this may change with growth of the popularity on the product segment. All the work is preformed over terminal server session on axmedis.neti.tv node.

3.7.3.1 Downloading of the objects

Object download is accomplished with AXMEDIS AXEPTool B2B utility; the AXEPTool documentation demonstrates its full use. If needed the Manager uses also AXMEDIS Editor to modify downloaded object.

3.7.3.2 Publishing the object

To publish AXMEDIS object for distribution manager copies the physical object into distribution folder and then uses SQL Server manager express studio to enter the object data into indexing database for public access.

3.7.3.3 Remove object

To remove AXMEDIS object from distribution manager deletes or moves physical object from distribution folder and then uses SQL Server manager express studio to delete the indexing entry from database.

3.7.3.4 Content reporting

Content reporting is being done with CAMART and AII in standard way and as such is fully documented in the corresponding tool documentation. In addition to CAMART and AII action log distribution there is a second way to get payment logs using local Win Forms utility to export those entries into CVS format transaction comparison.

4 Specification of the modules to be produced

4.1 Module AXSTB Player (TEO)

Module/Tool Profile		
AXSTB Player		
Responsible Name	Jonas Ceponis	
Responsible Partner	TEO, KTU	
Status (proposed/approved)	proposed	
Implemented/not implemented	not implemented	
Status of the implementation	Specification	
Executable or Library/module (Support)	Specification	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	Linux	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/..... Not yet available	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs. Not yet available	
Reference to the AXFW location of the demonstrator executable tool for public download	Not yet available	
Address for accessing to WebServices if any, add accession information (user and Passwd) if any	Not yet available	
Test cases (present/absent)	Absent	
Test cases location	http:// Not yet available	
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- Not yet available --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major	Communication model and format (protected or not, etc.)

	components needed	
User Authentication	AXCS	
Formats Used	Shared with	format name or reference to a section
MPEG-2 TS		
AXMEDIS STB streaming		
DVB-CSA		
Protocol Used	Shared with	Protocol name or reference to a section
RTSP		
RFC 2326		
Used Database name		
AXDB		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Player	C++	
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
gdk	1.2	LGPL
gtk	1.2	LGPL
glib	1.2	LGPL
gmodule	1.2	LGPL
X11	6	LGPL
Xext	6	LGPL
GtkIpc	3.3.1	Proprietary
Ipc	3.3.1	Proprietary
XmlParser	3.3.1	Proprietary
Toi	2	Proprietary

4.1.1 General Description of the Module

The AXSTB Player will be developed by porting AXMEDIS PC Player onto the Set-top-box. Functional requirements for the AXSTB Player are as follows:

- Should have an attractive user interface integrated with GALA TV screenshot system;

- STB Remote control enabled user functionality;
- Should enable the user to Play, Stop, Pause and Seek the digital content being played;
- Digital formats to be supported by the AXSTB Player:
 - MPEG -2
 - MPEG -4.

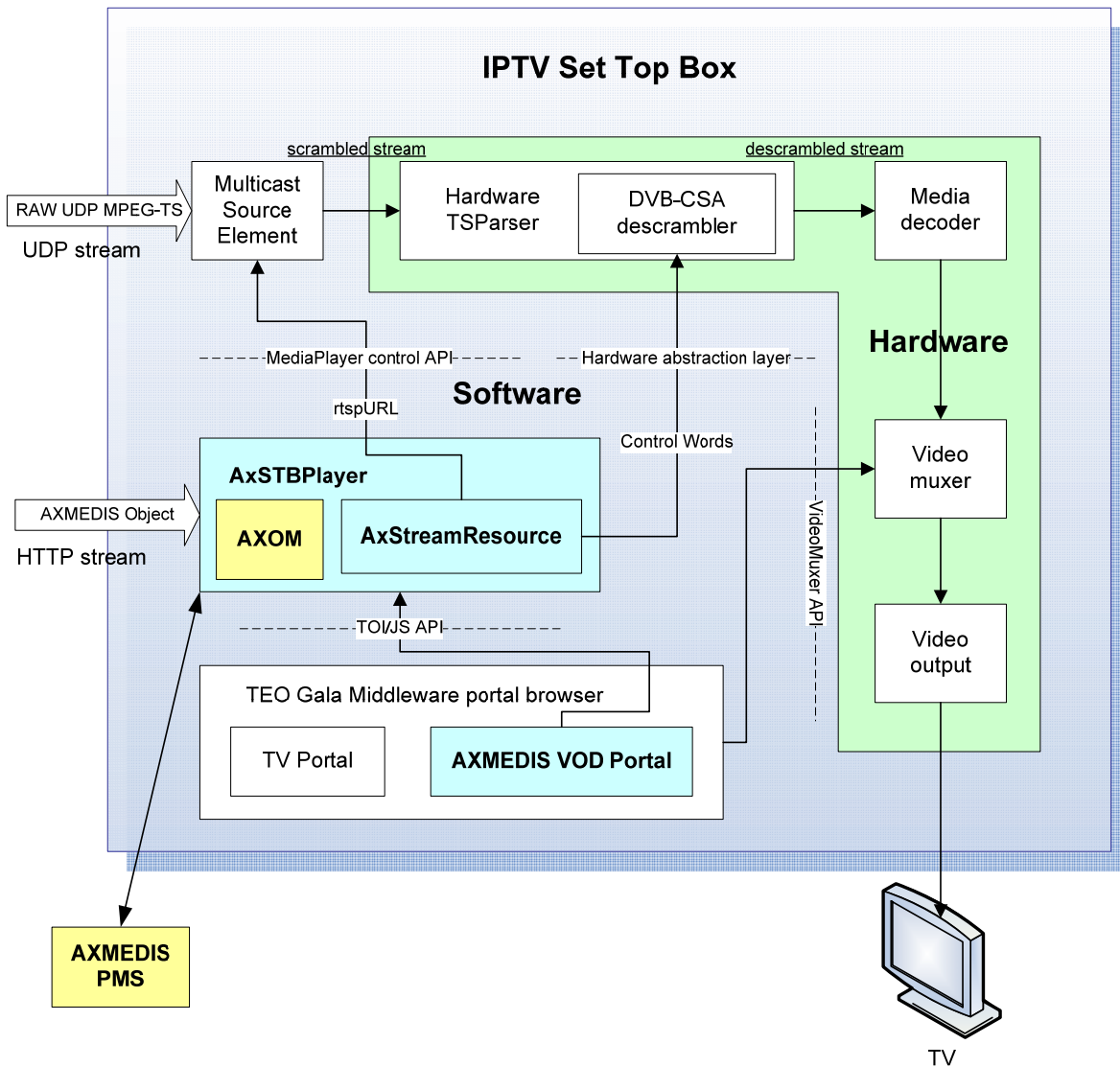
AXSTB Player utilizes AXOM to access the AXMEDIS object residing in TEO AXDB by providing AXOID. In TEO AXDB the AXMEDIS object metadata and stream information (AxStreamResource) is stored and the MPEG-2 TS resources are excluded due to storage efficiency.

When AXSTB Player tries to access AxStreamResource in AXMEDIS object, AXOM protection processor is executed to deprotect the object according to the issued license. If the user has bought the object and has valid license, AxStreamResource is decoded from object and AXSTB Player gains access to the following information:

1. Elementary audio/video streams composing MPEG-2 Transport Stream;
2. How to access MPEG-2 Transport Stream;
3. Hardware DVB-CSA Descrambler Control Words for each Elementary Stream.

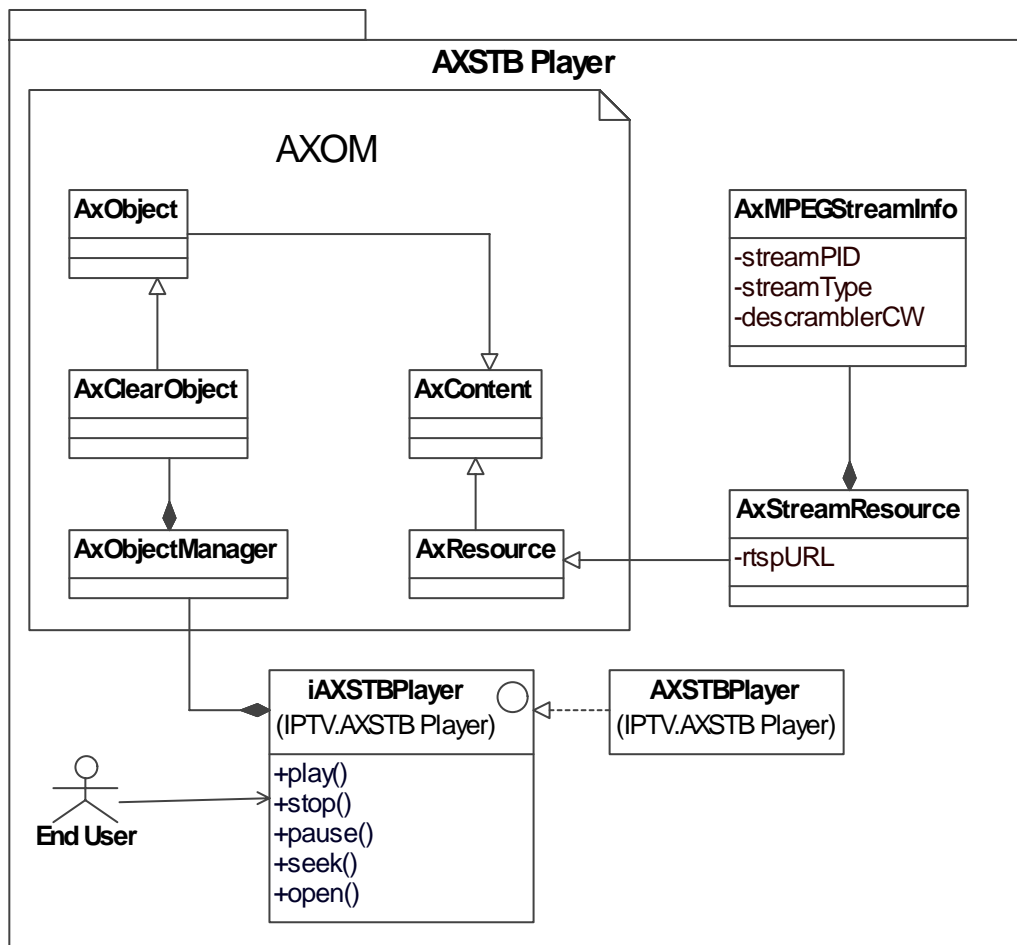
Therefore AXSTB Player can initiate the streaming of AXMEDIS Object MPEG-2 TS resource over network towards STB. On the other hand, AXSTB Player has access to the hardware DVB-CSA descrambler via Hardware abstraction layer and can setup hardware to process required streams by providing adequate descrambling Control Words. After initiation of MPEG-2 TS streaming from VOD server, STB starts receiving live scrambled video/audio stream. This stream is forwarded from multicast/unicast source element to DVB-CSA descrambler. Here the streams are descrambled by using control words. The descrambled streams are fed to Hardware MPEG decoder (Media decoder) and the decoded video is multiplexed with VOD portal view and sent to the video output of STB towards TV.

AXSTB Player is controlled by AXMEDIS VOD Portal which receives user's directions from remote control unit. In the picture below, the generalized Kreatel STB architecture and its modification to support playing of AXMEDIS Objects from VOD server is presented.



4.1.2 Module Design in terms of Classes

AXSTB Player package is specified in detail in the presented AXSTB Player class diagram. In AXSTB Player package Player interfaces is specified. Player interface is used for playing streamed video from VOD server. Player interface use iAccessService interface from VOD Access Service package. iAccessService interface communicates with VOD server, Billing DB, IPTV DB, AXDB, and AXMEDIS PMS. AXSTB Player class diagram is presented below.



AXSTB Player utilizes AXMEDIS Object Model and Manager to access and play resources. The original AXMEDIS object from VRS is never decrypted in TEO VOD servers. The video content reaches client STB in scrambled and untouched form. In order to be able to decrypt the scrambled audio/video content, Player has to utilize **AxObjectManager** and protection processor to gain Control Words. Without control words it is impossible to gain access to scrambled stream data. This solution utilizes hardware to achieve max performance and AXOM to achieve max AXMEDIS DRM compliance.

The following new classes will be developed in AXMEDIS Framework on STB side:

Player – represents TEO AXSTB Player which is compliant with TEO VOD server and supports playing of AXMEDIS Objects containing scrambled resources.

AxStreamResource – holds information how to access scrambled resource.

AxMPEGStreamInfo – hold information how to descramble specific stream of scrambled resource.

4.1.3 User interface description

Initial User interface is presented in sections 3.1.1-3.1.13. In these sections End User interface prototypes are presented. These prototypes describe only the information and its position in window, which is presented to End User.

4.1.4 Technical and Installation information

AXSTB Player can be installed into STB, which:

- supports of MPEG-2 and MPEG-4 TS over IP streaming;
- supports media player APIs and RTSP protocol for streaming;

- STB should be Linux/Windows-based;
- STB must be integrateable with VoD server, from which the stream will be performed;
- STB has to support hardware DVB-CSA descrambler for decrypting MPEG Transport streams.

4.1.5 Draft User Manual

To be done in the course of the project.

4.1.6 Examples of usage

End User can use AXSTB Player to:

- connect to TEO IPTV network;
- browse and see metadata of AXMEDIS video content;
- play AXMEDIS video content:
 - o stop playing video;
 - o pause video;
 - o continue playing;
 - o seek to desired point of video;
- view user information:
 - o view usage information;
 - o view billing information.

4.1.7 Integration and compilation issues

To be determined during the course of the project.

4.1.8 Configuration Parameters

Config parameter	Possible values
DB Connection string	axdbf.mdwclust:8080
AXMEDIS VOD Portal	http://mdwclust/STB/AxVOD
VOD Portal	http://mdwclust/STB/VOD
User name	User1
Password	xxxx
VOD Server	mdwclust/VOD

4.1.9 Errors reported and that may occur

To be determined during the course of the project.

Error code	Description and rationales

4.1.10 Formal description of algorithm <.....>

name	
Method	
Description	
Input parameters	
Output parameters	

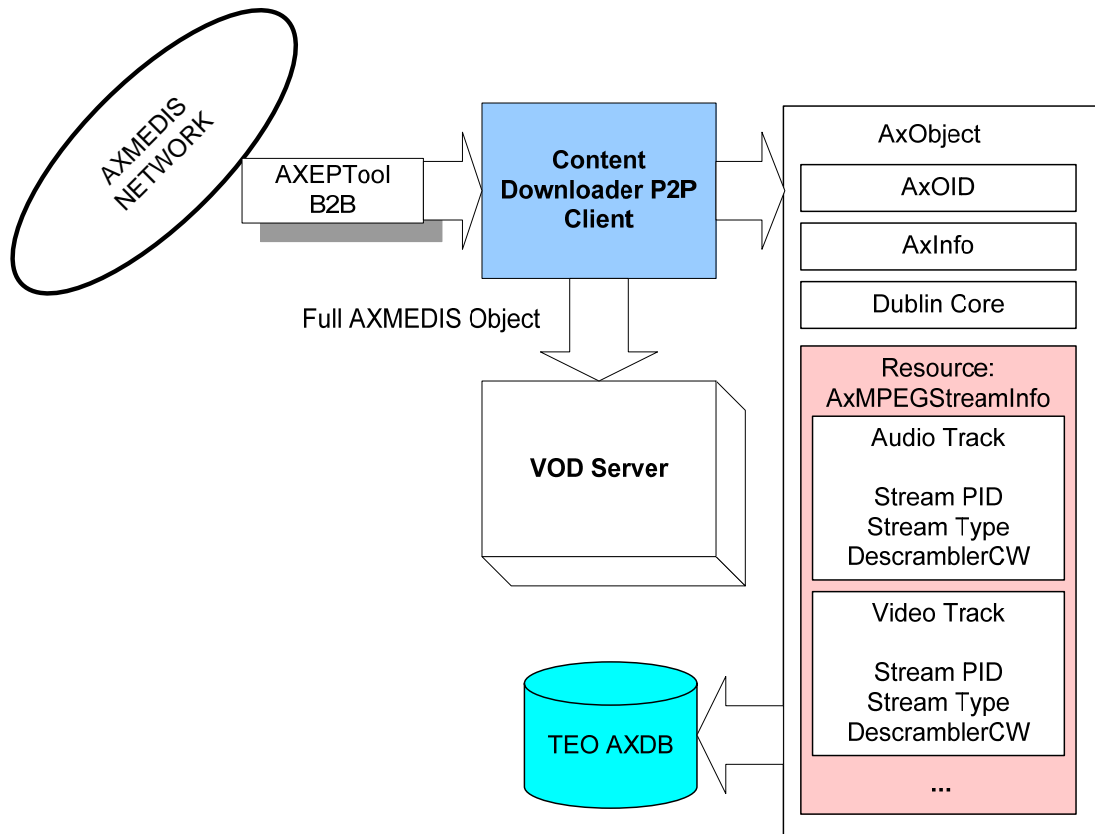
4.2 Module Content Downloader P2P Client (TEO)

Module/Tool Profile		
Content Downloader P2P Client		
Responsible Name	Jonas Ceponis	
Responsible Partner	TEO, KTU	
Status (proposed/approved)	proposed	
Implemented/not implemented	not implemented	
Status of the implementation	Specification	
Executable or Library/module (Support)	Specification	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	Windows	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/..... Not yet available	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs. Not yet available	
Reference to the AXFW location of the demonstrator executable tool for public download	Not yet available	
Address for accessing to WebServices if any, add accession information (user and Passwd) if any	Not yet available	
Test cases (present/absent)	Absent	
Test cases location	http:// Not yet available	
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- Not yet available --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
User Authentication	AXCS	

Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
AXDB		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

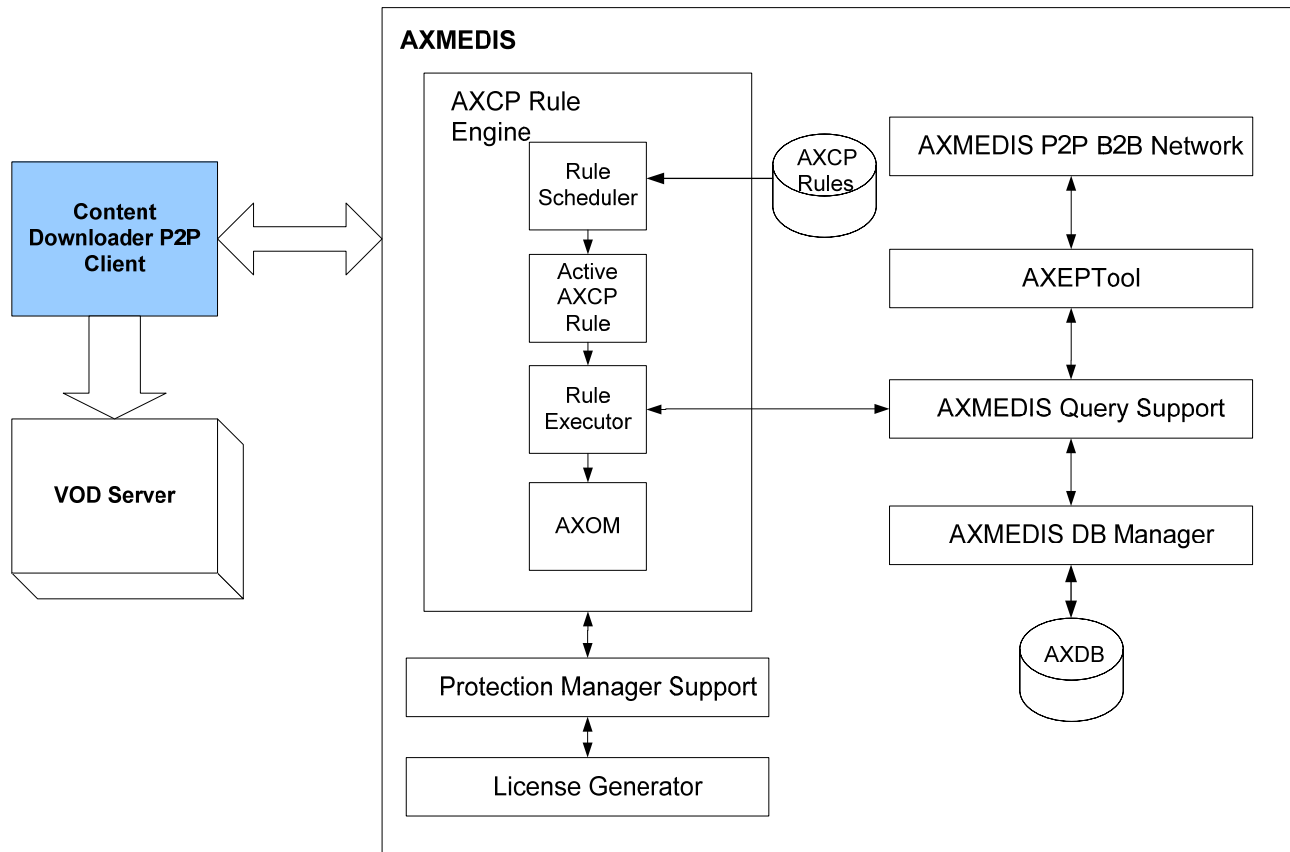
4.2.1 General Description of the Module

Content downloader P2P client will be newly developed component for placing downloaded AXMEDIS object to VOD server and updating AXMEDIS object metadata and placing metadata and stream information into AXDB. This component is part of AXMEDIS Content & User Management Portal and will be used by TEO VOD Manager for content management purposes.

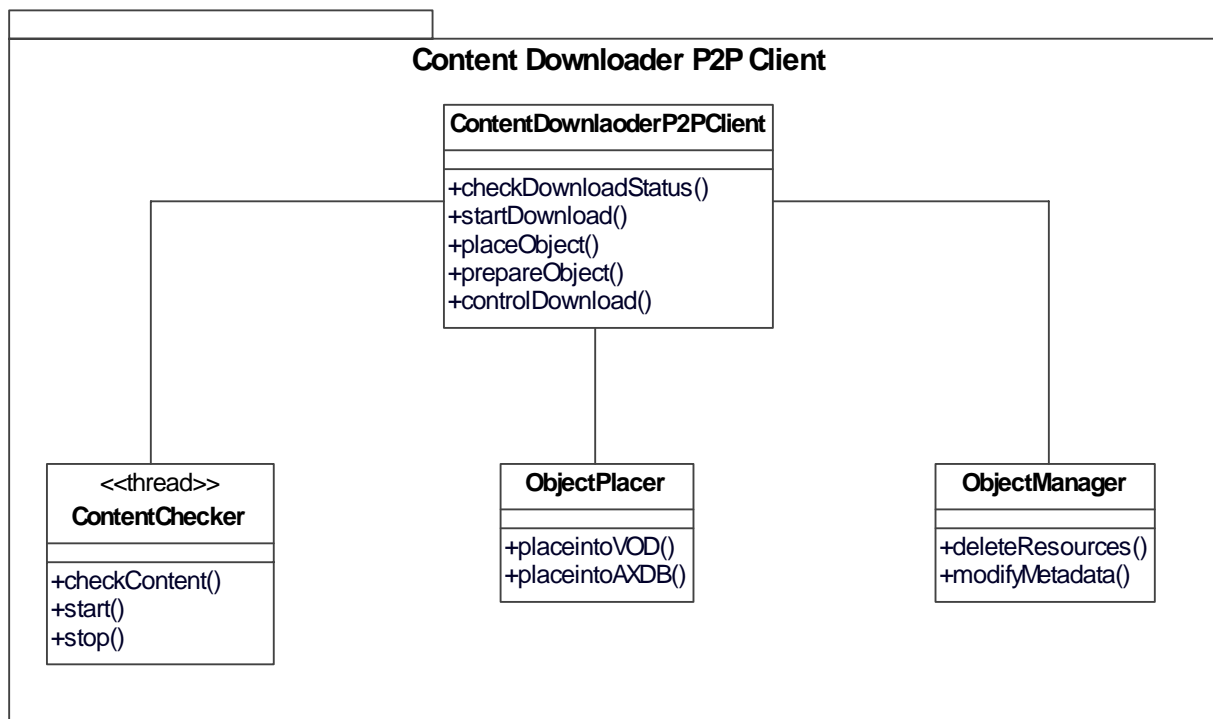


Content Downloader P2P Client will use AXEPTol for automatically/manually data downloading from AXMEDIS P2P B2B Network. Data retrieving from AXMEDIS P2P B2B Network is performed manually by TEO VOD Manager. In content management system TEO VOD Manager can download AXMEDIS object from global AXMEDIS network. In this case, AXMEDIS query is used for searching for the required object. When the object is found, it can be downloaded by using AXEPTool. AXMEDIS query and AXEPTool are tools which will be used as they are. Downloaded object will be saved in local AXDB and VOD Server.

Data retrieving from AXMEDIS P2P B2B Network in TEO Demonstrator setup can be performed automatically using AXCP Rule Engine. In this case it will be created AXCP rules and executed by Rule Scheduler and Rule Executor for searching objects in global AXMEDIS network. Rules will contain conditions for searching objects, which are created especially for TEO Demonstrator.



4.2.2 Module Design in terms of Classes



Content Downloader P2P Client module will be used for automatic download of AXMEDIS objects from B2B network. ContentChecker class will be responsible for automatic checking AXMEDIS objects intended for use in TEO demonstrator. Operation `checkContent()` uses information from configuration file for identifying content provider. During demonstration only VRS will be used as content provider.

ObjectManager class will be responsible for managing downloaded AXMEDIS objects. ObjectPlacer class will be used for placing AXMEDIS objects into VOD server and AXDB.

ContentDownloaderP2PClient class will manage checking and downloading new available content intended for TEO, preparing and placing objects into VOD Server and AXDB

4.2.3 User interface description

This module does not have user interface.

4.2.4 Technical and Installation information

Content downloader P2P client can be implemented in server where is Microsoft IIS HTTP server with ASP.NET 2.0 extensions, Microsoft SQL server 2005 x64, Microsoft .NET framework 2.0 x64.

4.2.5 Draft User Manual

To be done in the course of the project.

4.2.6 Examples of usage

TEO VOD Manager uses Content downloader P2P client for placing full AXMEDIS Object into VOD Server and updating AXMEDIS object metadata and placing metadata and stream information into AXDB.

4.2.7 Integration and compilation issues

To be determined during the course of the project.

4.2.8 Configuration Parameters

Config parameter	Possible values

4.2.9 Errors reported and that may occur

To be determined during the course of the project.

Error code	Description and rationales

4.2.10 Formal description of algorithm <.....>

name	
Method	
Description	
Input parameters	
Output parameters	

4.3 Module VOD Access Service (TEO)

Module/Tool Profile		
VOD Access Service		
Responsible Name	Jonas Ceponis	
Responsible Partner	TEO, KTU	
Status (proposed/approved)	proposed	
Implemented/not implemented	not implemented	
Status of the implementation	Specification	
Executable or Library/module (Support)	Specification	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	Windows	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/..... Not yet available	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs. Not yet available	
Reference to the AXFW location of the demonstrator executable tool for public download	Not yet available	
Address for accessing to WebServices if any, add accession information (user and Passwd) if any	Not yet available	
Test cases (present/absent)	Absent	
Test cases location	http:// Not yet available	
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- Not yet available --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
User Authentication	AXCS	

Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
AXDB		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

4.3.1 General Description of the Module

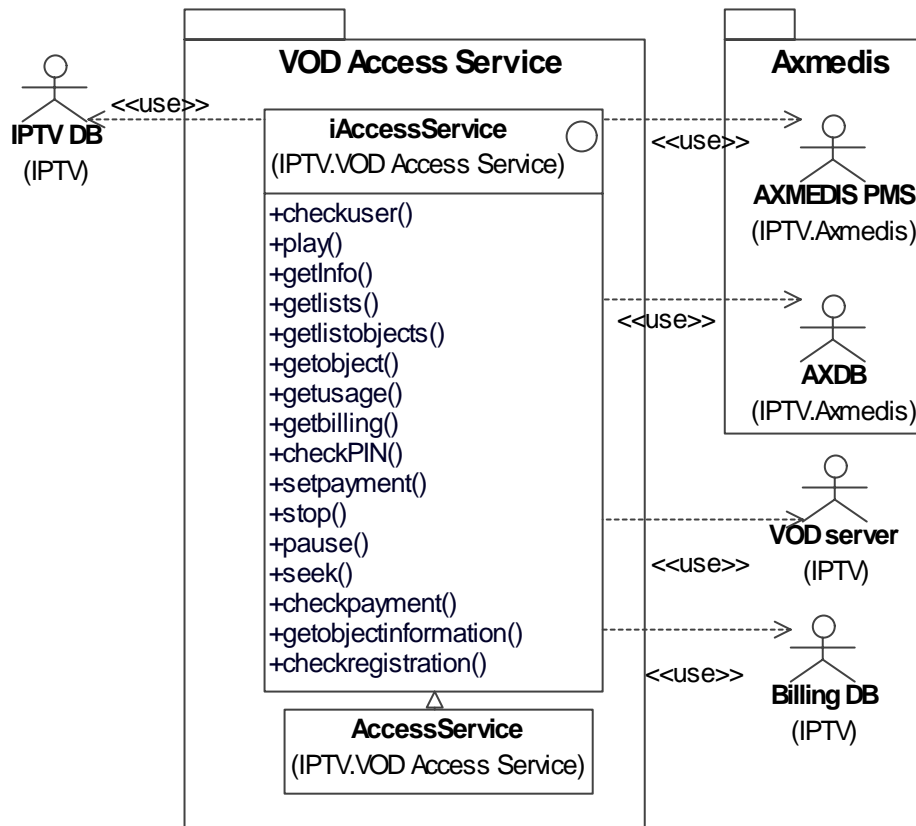
VOD Access Service package is responsible for proper End User AXSTB Player connection with AXMEDIS content. This service will act in middleware server.

In AXSTB Player package Player interface uses iAccessService interface from VOD Access Service package. iAccessService interface communicates with VOD server, Billing DB, IPTV DB, AXDB, and AXMEDIS PMS.

4.3.2 Module Design in terms of Classes

VOD Access Service class diagram is presented below. iAXSTBPlayer interface use iAccessService interface from VOD Access Service package. iAccessService interface

communicates with VOD server, Billing DB (using BillingWS web service described in section 5.2.1), IPTV DB, AXDB, and AXMEDIS PMS.



4.3.2.1 VOD Access Service web service WSDL specification

Initial WSDL specification of VOD Access Service web service is presented below. This specification will be completed during implementation stage of the module.

```

<wsdl:definitions
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://localhost/VODAccessService/VODAccessService.asmx/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://localhost/VODAccessService/VODAccessService.asmx/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
  </wsdl:types>
  <wsdl:message name="getObject">
    <wsdl:part name="ObjectID" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="getInfo">
    <wsdl:part name="Info" type="xsd:string" />
  </wsdl:message>
  </wsdl:definitions>
    
```

```

</wsdl:message>
<wsdl:portType name="GetPortType">
  <wsdl:operation name="getObject">
    <wsdl:input message="tns:getObject" />
    <wsdl:output message="tns:getObject" />
  </wsdl:operation>
  <wsdl:operation name="getInfo">
    <wsdl:input message="tns:getObject" />
    <wsdl:output message="tns:getInfo" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="PlayerPortType">
  <wsdl:operation name="play">
    <wsdl:input message="tns:getObject" />
  </wsdl:operation>
  <wsdl:operation name="stop">
    <wsdl:input message="tns:getObject" />
  </wsdl:operation>
  <wsdl:operation name="pause">
    <wsdl:input message="tns:getObject" />
  </wsdl:operation>
  <wsdl:operation name="seek">
    <wsdl:input message="tns:getObject" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="GetPortBinding" type="tns:GetPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="getObject">
    <soap12:operation
soapAction="http://localhost/VODAccessService/VODAccessService.asmx/getObject"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getInfo">
    <soap12:operation
soapAction="http://localhost/VODAccessService/VODAccessService.asmx/getInfo"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```

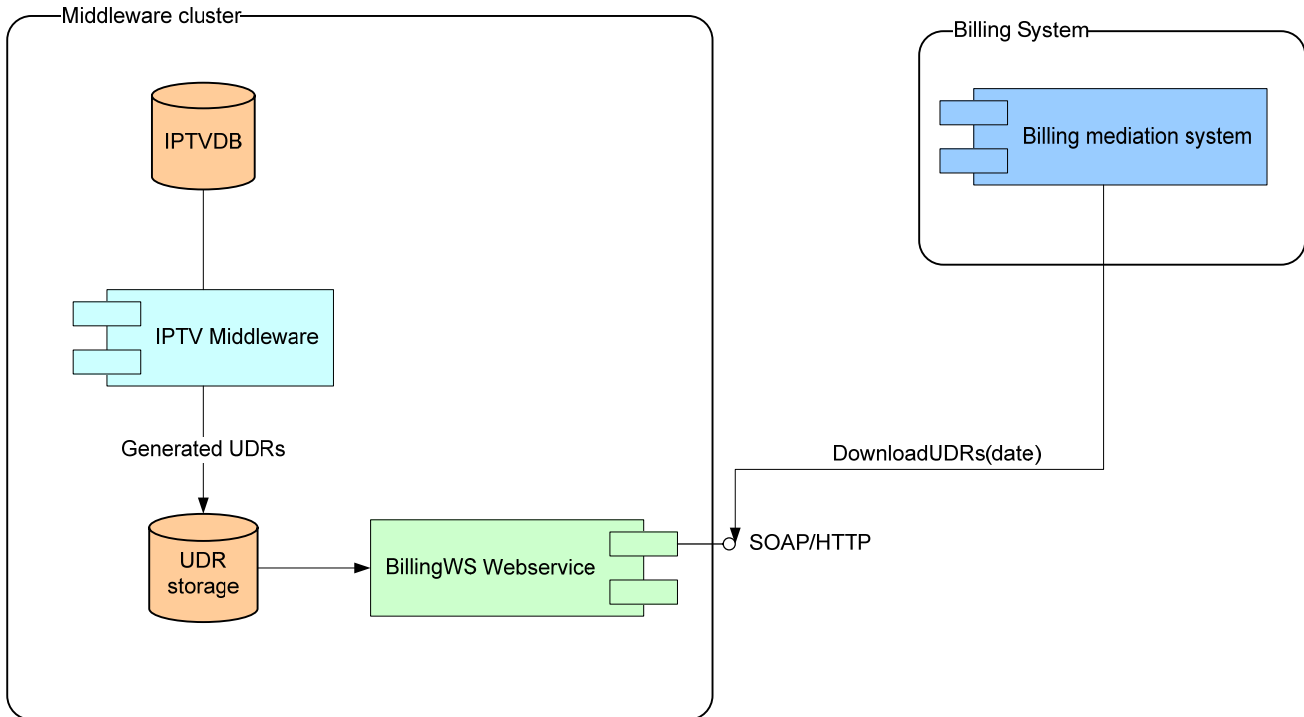
<wsdl:binding name="PlayerPortBinding" type="tns:PlayerPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="play">
    <soap12:operation
soapAction="http://localhost/VODAccessService/VODAccessService.asmx/play"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="stop">
    <soap12:operation
soapAction="http://localhost/VODAccessService/VODAccessService.asmx/stop"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="pause">
    <soap12:operation
soapAction="http://localhost/VODAccessService/VODAccessService.asmx/pause"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="seek">
    <soap12:operation
soapAction="http://localhost/VODAccessService/VODAccessService.asmx/seek"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="VODAccessService">
  <wsdl:documentation>Connect STB with DB's and VOD server</wsdl:documentation>
  <wsdl:port name="ObjectPort" binding="tns:GetPortBinding">
    <soap:address location="http://localhost/VODAccessService/VODAccessService.asmx/" />
  </wsdl:port>
  <wsdl:port name="PlayerPort" binding="tns:PlayerPortBinding">
    <soap:address location="http://localhost/VODAccessService/VODAccessService.asmx/" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

4.3.2.2 BillingWS web service specification

VOD Access Service communicates with BillingDB using BillingWS web server. In this section BillingWS web service is described. IPTV Middleware system gathers information about specific user actions, for which user can be charged. For example, ordering of a paid channel, subscription/unsubscription of a set-top box, etc.

Below is the Billing system and Middleware system integration scheme.



User status – active services are saved in IPTVDB database. When the user buys/refuses the service, UDR (Usage Data Records) database traces this event by filling UDR record. UDR record contains the exact date and time, event type and other information related to the event. The BillingWS webservice is used at IPTV middleware side for integration with the billing system. The BillingWS webservice provides a method called DownloadUDRs(date) which is invoked by TEO Billing mediation system. This method retrieves the usage data records (URSs) for the specified period. The Billing system invokes this service once a day, showing the date and time from which all the UDR records must be sent.

1. BillingWS web service address `http://<internal address>/BillingWS/udr.asmx`
2. BillingWS has one service: `DownloadUDRs(startTime)`
3. `startTime` parameter shows date and time. This parameter indicates for web service that it needs to send all UDR records, which were made no earlier than shown `startTime`. Normally `startTime` parameter will be date and time of the last Billing system's `DownloadUDRs` operation
4. `DownloadUDRs` service returns the list of UDR records with the following structure of one UDR record:

Parameter	Format	Description
EventTime	YYYY.MM.DD hh:mm:ss	Exact event type
ServiceID	Integer	Service ID – identifies the subscriber at TEO CRM system.

EventData	VarChar(255)	ASCII code set to 255 symbols. Describes event. The format of this parameter is defined in paragraph „EventData parameter Format“.
ValidFrom	YYYY.MM.DD hh:mm:ss (optional field)	Shows the date and time from which user started using service. From this date the user must be charged for the service. In case of AXMEDIS content usage this field is mandatory.
ValidTo	YYYY.MM.DD hh:mm:ss (optional field)	Shows the date and time till which user is using service. From this date Billing system must stop charging the user for the service. This parameter also could be null. In this case service will be provided for indefinitely, paying for it periodically as shown by <i>PricingStrategy</i> parameter.
PricingStrategy	Char(2)	Two symbols code, showing user's paying for service strategy. Possible parameter combinations in the regular IPTV subscription service: MO – subscription fee, charged every month for using the service. DA - subscription fee, charged every day for using the service. VI – onetime fee for service. Service is activated for user for one time (session) and is charged on the same day. It is used for pay-per-view, video-on-demand type services. Special variations of onetime services pertaining to AXMEDIS demonstrator: V1 – AXMEDIS free content usage with adverts. V2 – AXMEDIS paid content with 2 adverts. V3 – AXMEDIS paid content without adverts. V4 – special version AXMEDIS content
Price	Decimal	Service price Lt (Litas, national currency of Lithuania)

Web Service can be called in three ways: using HTTP POST method and parameter, SOAP 1.1 protocol, SOAP 1.2 protocol. The samples of requests and responses from all three ways are presented below:

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

POST /BillingWS/udr.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://<internal address>/BillingWS/udr.asmx/DownloadUDRs"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DownloadUDRs xmlns="http://<internal address>/BillingWS/udr.asmx/">
      <startTime>dateTime</startTime>
    </DownloadUDRs>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DownloadUDRsResponse xmlns="http://<internal address>/BillingWS/udr.asmx/">
      <DownloadUDRsResult>
        <UsageDataRecord>
          <EventTime>string</EventTime>
          <PaslaugosID>int</PaslaugosID>
          <EventData>string</EventData>
          <ValidFrom>string</ValidFrom>
          <ValidTo>string</ValidTo>
          <PricingStrategy>string</PricingStrategy>
          <Price>decimal</Price>
        </UsageDataRecord>
        <UsageDataRecord>
          <EventTime>string</EventTime>
          <PaslaugosID>int</PaslaugosID>
          <EventData>string</EventData>
          <ValidFrom>string</ValidFrom>
          <ValidTo>string</ValidTo>
          <PricingStrategy>string</PricingStrategy>
          <Price>decimal</Price>
        </UsageDataRecord>
      </DownloadUDRsResult>
    </DownloadUDRsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

POST /BillingWS/udr.asmx HTTP/1.1

Host: localhost

AXMEDIS Project

Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DownloadUDRs xmlns="http://<internal address>/BillingWS/udr.asmx/">
      <startTime>dateTime</startTime>
    </DownloadUDRs>
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DownloadUDRsResponse xmlns="http://<internal address>/BillingWS/udr.asmx/">
      <DownloadUDRsResult>
        <UsageDataRecord>
          <EventTime>string</EventTime>
          <PaslaugosID>int</PaslaugosID>
          <EventData>string</EventData>
          <ValidFrom>string</ValidFrom>
          <ValidTo>string</ValidTo>
          <PricingStrategy>string</PricingStrategy>
          <Price>decimal</Price>
        </UsageDataRecord>
        <UsageDataRecord>
          <EventTime>string</EventTime>
          <PaslaugosID>int</PaslaugosID>
          <EventData>string</EventData>
          <ValidFrom>string</ValidFrom>
          <ValidTo>string</ValidTo>
          <PricingStrategy>string</PricingStrategy>
          <Price>decimal</Price>
        </UsageDataRecord>
      </DownloadUDRsResult>
    </DownloadUDRsResponse>
  </soap12:Body>
</soap12:Envelope>
```

HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to be replaced with actual values.

POST /BillingWS/udr.asmx/DownloadUDRs HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

Content-Length: length

startTime=string

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfUsageDataRecord xmlns="http://<internal address>/BillingWS/udr.asmx/">
  <UsageDataRecord>
    <EventTime>string</EventTime>
    <PaslaugosID>int</PaslaugosID>
    <EventData>string</EventData>
    <ValidFrom>string</ValidFrom>
    <ValidTo>string</ValidTo>
    <PricingStrategy>string</PricingStrategy>
    <Price>decimal</Price>
  </UsageDataRecord>
  <UsageDataRecord>
    <EventTime>string</EventTime>
    <PaslaugosID>int</PaslaugosID>
    <EventData>string</EventData>
    <ValidFrom>string</ValidFrom>
    <ValidTo>string</ValidTo>
    <PricingStrategy>string</PricingStrategy>
    <Price>decimal</Price>
  </UsageDataRecord>
</ArrayOfUsageDataRecord>
```

EventData text parameter transfers information about event type and its details. This parameter consists of fixed part and changeable part, which depends on event type.

General EventData format:

I	i	I	To 252 symbols
---	---	---	----------------

Service Metadata

type

i – integer.

Service type – three numeric symbols code, identifying charged service type, e.g.: 000 – buying of AXMEDIS video content on IPTV.

Metadata describes the data about particular service. Below, the format of this field is described for AXMEDIS service:

Service 010 – AXMEDIS content viewing

This event type describes the fact of AXMEDIS content usage. When the user on the screen chooses the film and confirms the purchase, MW generates new type UDR with the service type 010 – AXMEDIS content usage. The format of EventData parameter is presented below:

0	0	2	i	i	i	i	i	i	I	Up to 245 symbols
---	---	---	---	---	---	---	---	---	---	-------------------

Service type TEO internal film ID AXMEDIS Object ID (AXOID)

i – integer.

The unused positions of ID numbers are changed with spaces and later, when used in information system, must be removed.

Film ID – is MW internal film ID. This ID, together with film titles, possible values are not described in UDR specification, because they vary in time. This information must be stored in Billing system and presented as data for claims analysis.

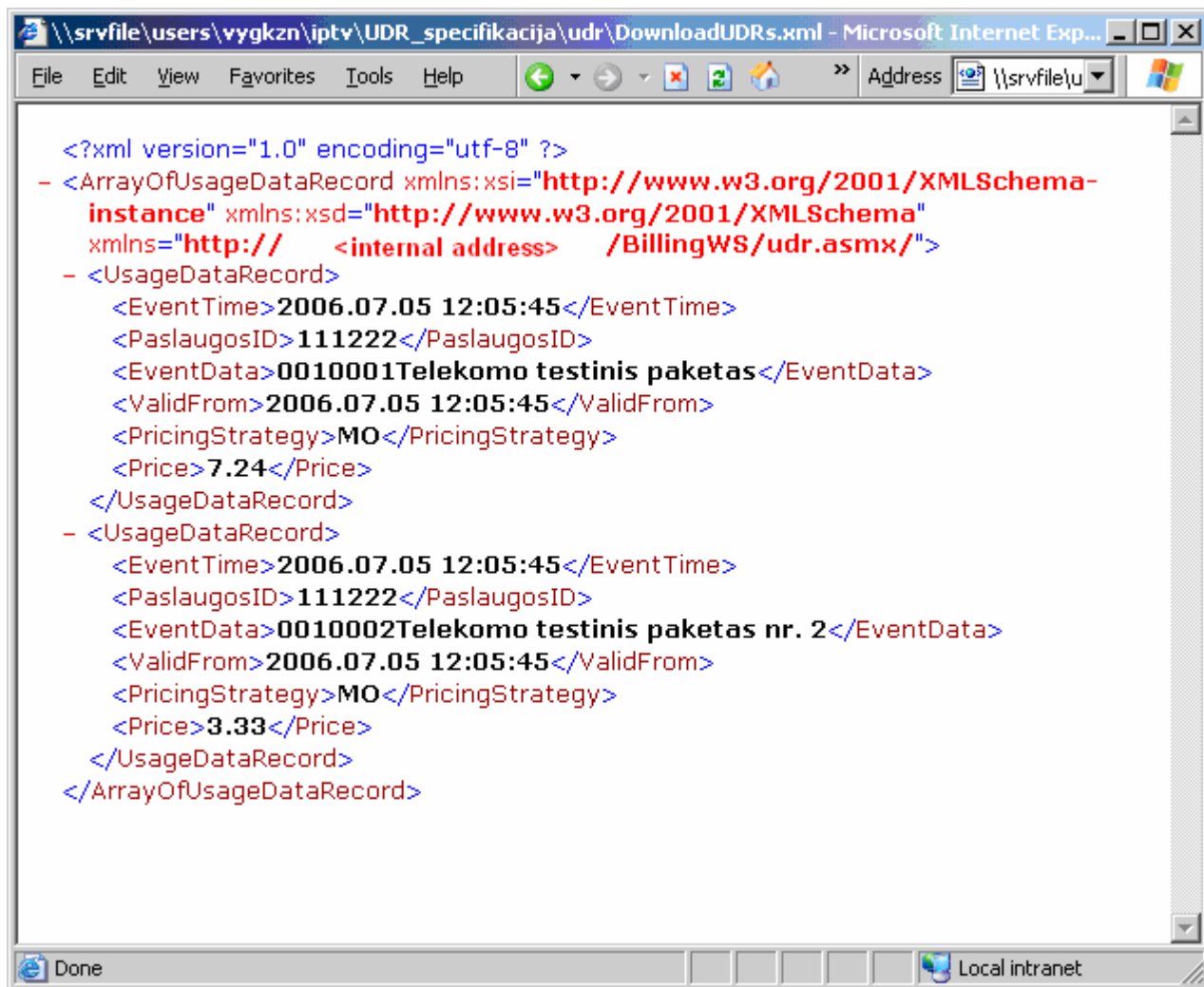
In AXMEDIS buying event date/time is saved in ValidFrom, which shows when user started using the service. Also date/time ValidTo is saved, which shows till when the service must be provided. MW also generates package PricingStrategy and Pricing parameters that must be ignored at this stage.

UDR mandatory fields table:

Parameter	Obligation	Comments
EventTime	Mandatory	
PaslaugosID	Mandatory	
EventData	Mandatory	
ValidFrom	Mandatory	
ValidTo	Mandatory	
PricingStrategy	Mandatory	
Price	Mandatory	

AXMEDIS content view service is provided in the period ValidFrom..ValidTo. UDRs with the same Film ID that appears during this period must be ignored and not charged repeatedly.

Below there is testing web service typical XML response to request:



BillingWS web service WSDL specification

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions                                xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"                                xmlns:tns="http://<internal
address>/BillingWS/udr.asmx/"                                xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"    targetNamespace="http://<internal
address>/BillingWS/udr.asmx/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema                elementFormDefault="qualified"                targetNamespace="http://<internal
address>/BillingWS/udr.asmx/">
      <s:element name="DownloadUDRs">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="startTime" type="s:dateTime" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </wsdl:types>
  </wsdl:definitions>

```

```

<s:element name="DownloadUDRsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="DownloadUDRsResult"
type="tns:ArrayOfUsageDataRecord" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ArrayOfUsageDataRecord">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="UsageDataRecord"
type="tns:UsageDataRecord" />
  </s:sequence>
</s:complexType>
<s:complexType name="UsageDataRecord">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="EventTime" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="PaslaugosID" type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="EventData" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="ValidFrom" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="ValidTo" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="PricingStrategy" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Price" type="s:decimal" />
  </s:sequence>
</s:complexType>
</s:schema>
</wsdl:types>
<wsdl:message name="DownloadUDRsSoapIn">
  <wsdl:part name="parameters" element="tns:DownloadUDRs" />
</wsdl:message>
<wsdl:message name="DownloadUDRsSoapOut">
  <wsdl:part name="parameters" element="tns:DownloadUDRsResponse" />
</wsdl:message>
<wsdl:portType name="UDRSoap">
  <wsdl:operation name="DownloadUDRs">
    <wsdl:input message="tns:DownloadUDRsSoapIn" />
    <wsdl:output message="tns:DownloadUDRsSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="UDRSoap" type="tns:UDRSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="DownloadUDRs">
    <soap:operation soapAction="http://<internal address>/BillingWS/udr.asmx/DownloadUDRs"
style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>

```

```

</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="UDRSoap12" type="tns:UDRSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="DownloadUDRs">
    <soap12:operation
      address="/BillingWS/udr.asmx/DownloadUDRs" style="document" />
      soapAction="http://<internal
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="UDR">
  <wsdl:port name="UDRSoap" binding="tns:UDRSoap">
    <soap:address location="http://localhost/BillingWS/udr.asmx" />
  </wsdl:port>
  <wsdl:port name="UDRSoap12" binding="tns:UDRSoap12">
    <soap12:address location="http://localhost/BillingWS/udr.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

4.3.3 User interface description

This module does not have user interface.

4.3.4 Technical and Installation information

VOD Access Service can be implemented in server where is Microsoft IIS HTTP server with ASP.NET 2.0 extensions, Microsoft SQL server 2005 x64, Microsoft .NET framework 2.0 x64.

4.3.5 Draft User Manual

To be done in the course of the project.

4.3.6 Examples of usage

VOD Access Service package is responsible for proper End User AXSTB Player connection with AXMEDIS content. This service will act in middleware server.

4.3.7 Integration and compilation issues

To be determined during the course of the project.

4.3.8 Configuration Parameters

Config parameter	Possible values
DB Connection string	axdbf.mdwclust:8080
AXMEDIS VOD Portal	http://mdwclust/Kreatel/AxVOD
VOD Portal	http://mdwclust/Kreatel/VOD
VOD Server	mdwclust/VOD

4.3.9 Errors reported and that may occur

To be determined during the course of the project.

Error code	Description and rationales

4.3.10 Formal description of algorithm <.....>

name	
Method	
Description	
Input parameters	
Output parameters	

4.4 Module TEO Content & User Management Portal (TEO)

Module/Tool Profile		
TEO Content & User Management Portal		
Responsible Name	Jonas Ceponis	
Responsible Partner	TEO, KTU	
Status (proposed/approved)	proposed	
Implemented/not implemented	not implemented	
Status of the implementation	Specification	
Executable or Library/module (Support)	Specification	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	Windows	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/..... Not yet available	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs. Not yet available	
Reference to the AXFW location of the demonstrator executable tool for public download	Not yet available	
Address for accessing to WebServices if any, add accession information (user and Passwd) if any	Not yet available	
Test cases (present/absent)	Absent	
Test cases location	http:// Not yet available	
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- Not yet available --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
User Authentication	AXCS	

Formats Used	Shared with	format name or reference to a section
AXMEDIS STB streaming		
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
AXDB		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

4.4.1 General Description of the Module

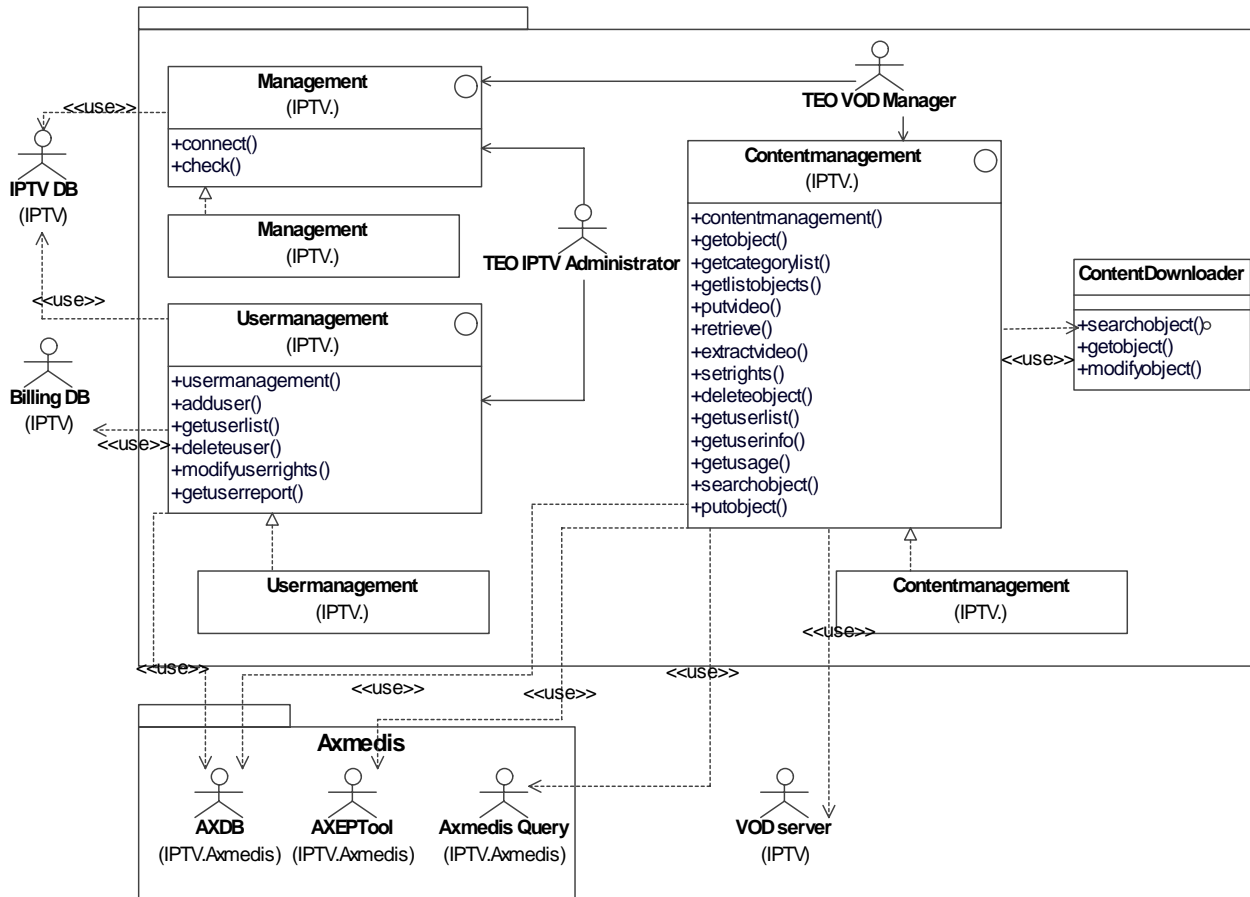
TEO Content & User Management Portal package has all tools for managing AXMEDIS content in local DB. This package is also responsible for TEO AXMEDIS user management. TEO Content & User Management Portal will be used by TEO VOD Manager. TEO Content & User Management Portal module will be developed on IPTV Content Management Portal platform.

4.4.2 Module Design in terms of Classes

Three interfaces are specified in TEO Content & User Management Portal package: Management, iUserManagement and iContentManagement. Management interface is used to connect to TEO Content & User Management Portal. In order to perform this operation, Management interface communicates with IPTV DB. iUserManagement interface is responsible for user adding, deleting,

modifying and reporting operations. iUserManagement interface communicates with IPTV DB, Billing DB and AXDB. iContentManagement interface is used for content downloading, preparing, deleting and reporting. This interface communicates with AXDB, AXEPTool, AXMEDIS Query and VOD Server.

TEO Content & User Management Portal class diagram:



4.4.3 User interface description

Initial TEO Content & User Management Portal user interfaces are presented in sections 3.2.1-3.3.6. In these sections user interface prototypes are presented. These prototypes describe only the information and its position in window, which is presented to TEO VOD Manager and TEO IPTV Administrator.

4.4.4 Technical and Installation information

TEO Content & User Management Portal can be implemented in a dedicated server in Microsoft IIS HTTP server with ASP.NET 2.0 extensions, Microsoft SQL server 2005 x64, Microsoft .NET framework 2.0 x64.

4.4.5 Draft User Manual

To be done in the course of the project.

4.4.6 Examples of usage

TEO VOD Manager using TEO Content & User Management Portal can:

- Connect to TEO Content & User Management Portal;
- Manage TEO AXMEDIS content:
 - o Download AXMEDIS content from global AXMEDIS DB, using AXEPTool and place into VOD Server and AXDB;
 - o Delete content from local TEO AXDB;

- Make and view reports about usage of AXMEDIS content in TEO IPTV network;
- Manage TEO AXMEDIS users:
 - Add user to TEO AXMEDIS DB;
 - Delete user from TEO AXMEDIS DB;
 - Set user rights to use AXMEDIS content in TEO IPTV network;
 - Make and view reports about user actions in TEO AXMEDIS DB.

4.4.7 Integration and compilation issues

To be determined during the course of the project.

4.4.8 Configuration Parameters

Config parameter	Possible values
DB Connection string	axdbf.mdwclust:8080
AXMEDIS VOD Portal	http://mdwclust/STB/AxVOD
VOD Portal	http://mdwclust/STB/VOD
VOD Server	mdwclust/VOD

4.4.9 Errors reported and that may occur

To be determined during the course of the project.

Error code	Description and rationales

4.4.10 Formal description of algorithm <.....>

name	
Method	
Description	
Input parameters	
Output parameters	

4.5 Plug-in MPEGTSAadaptationPlugin (TEO)

Module/Tool Profile		
MPEGTSAadaptationPlugin		
Responsible Name	Jonas Ceponis	
Responsible Partner	TEO, KTU	
Status (proposed/approved)	proposed	
Implemented/not implemented	not implemented	
Status of the implementation	Specification	
Executable or Library/module (Support)	Specification	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	Windows	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/..... Not yet available	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs. Not yet available	
Reference to the AXFW location of the demonstrator executable tool for public download	Not yet available	
Address for accessing to WebServices if any, add accession information (user and Passwd) if any	Not yet available	
Test cases (present/absent)	Absent	
Test cases location	http:// Not yet available	
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- Not yet available --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)

Formats Used	Shared with	format name or reference to a section
MPEG-2 TS		
AXMEDIS STB streaming		
DVB-CSA		
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
AXDB		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

4.5.1 General Description of the Module

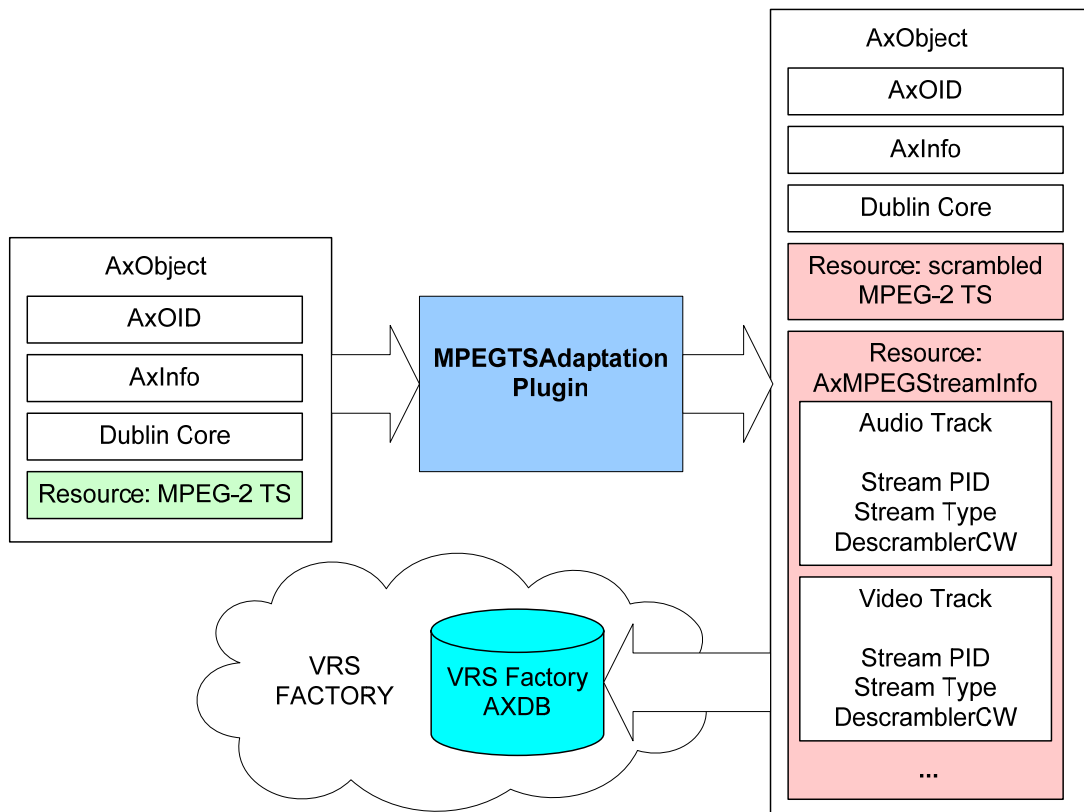
In AXELTEO architecture VRS prepares videoclips – AXMEDIS objects that contain metadata with resources. The resources are MPEG-2 Transport Stream video/audio content. At first the produced content has to be adapted to streaming requirements at VRS Factory by using AXMEDIS Editor or AXCP. For this purpose the AXCP MPEGTSAdaptationPlugin will be developed. This plugin performs two tasks:

- Scrambles MPEG-2 Transport Stream (TS) resource using DVB-CSA algorithm so that content is ready for streaming over IP and protected. The MPEGTSAdaptationPlugin must encrypt all video and audio channels of stream and leave unencrypted frames containing Program Clock Reference (PCR) which is critical for streaming control. Each scrambled frame of transport stream, has to be marked as scrambled according to ISO13818-1 standard.

- The second task of plugin is to collect information about all scrambled elementary MPEG-2 TS streams (ES). This task associates each ES with DVB-CSA Control Word (CW) which was used for scrambling the ES. This information is returned by the plugin as a new RESOURCE which is appended to the transformed AXMEDIS Object.

After adaption the transformed AXMEDIS Object consists of:

3. Scrambled MPEG-2 Transport Stream;
4. Information about scrambled MPEG-2 TS elementary streams and DVB-CSA CWs in resource AxMPEGStreamInfo.

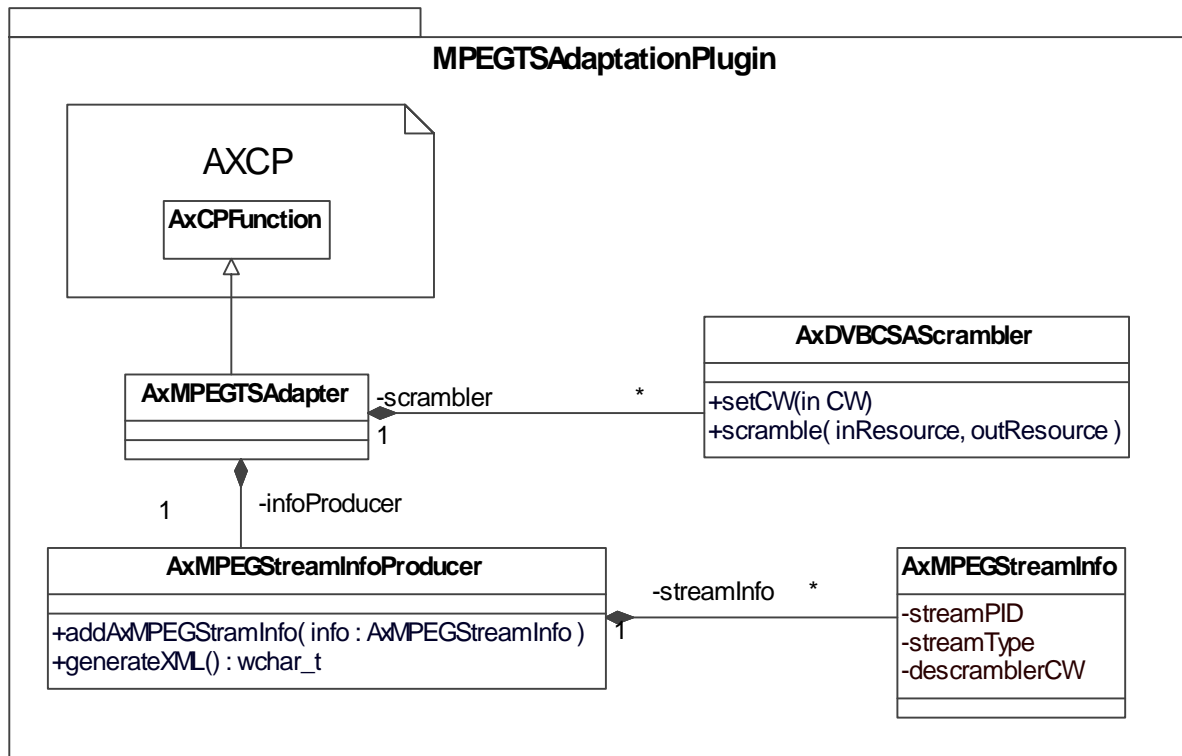


This transformed AXMEDIS Object is protected by using AXOM Protection Processor and put to VRS AXDB.

MPEGTSAdaptationPlugin will be implemented in VRS Factory or used directly by AXEDITOR.

4.5.2 Module Design in terms of Classes

The picture below shows what has to be developed in MPEGTSAdaptationPlugin in detail.

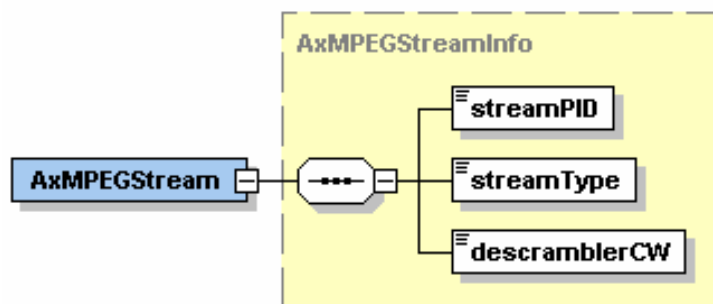


AxMPEGTSAdapter – is an *AxCPFunction* instance which coordinates the scrambling and Stream Information extraction tasks.

AxDCBCSAScrambler – is a class which implements DVB-CSA scrambling algorithm and applies it to the resource.

AxMPEGStreamInfoProducer class is responsible for collection of MPEG-TS elementary stream information and recording of descrambler CWs in class *AxMPEGStreamInfo* instances.

AxMPEGStreamInfo format:



```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:complexType name="AxMPEGStreamInfo">
<xs:sequence>
<xs:element name="streamPID" type="xs:string"/>
<xs:element name="streamType" type="xs:string"/>
<xs:element name="descramblerCW" type="xs:string"/>

```

```

</xs:sequence>
</xs:complexType>
<xs:element name="AxMPEGStream" type="AxMPEGStreamInfo"/>
</xs:schema>

```

4.5.3 User interface description

This module will have user interface similar to other content processing plug-ins in AXMEDIS Editor.

4.5.4 Technical and Installation information

To be determined during the course of the project.

4.5.5 Draft User Manual

To be done in the course of the project.

4.5.6 Examples of usage

Scrambling MPEG-2 Transport Stream (TS) resource by using DVB-CSA algorithm. After that content is ready for streaming over IP and protected. The MPEGTSAadaptationPlugin must encrypt all video and audio channels of stream and leave unencrypted frames containing Program Clock Reference (PCR) which is critical for streaming control. Each scrambled frame of transport stream has to be marked as scrambled according to ISO13818-1 standard.

MPEGTSAadaptationPlugin plug-in is used for collecting information about all scrambled elementary MPEG-2 TS streams (ES). This task associates each ES with DVB-CSA Control Word (CW) which was used for scrambling the ES. This information is returned by the plug-in as a new RESOURCE which is appended to the transformed AXMEDIS Object.

4.5.7 Integration and compilation issues

To be determined during the course of the project.

4.5.8 Configuration Parameters

To be determined during the course of the project.

Config parameter	Possible values

4.5.9 Errors reported and that may occur

To be determined during the course of the project.

Error code	Description and rationales

4.5.10 Formal description of algorithm <.....>

name	
Method	
Description	
Input parameters	
Output parameters	

4.6 Elion portal integration module ELAXAPI_WS

Module/Tool Profile	
ELAXAPI_WS	
Responsible Name	Andrius Lokotaš
Responsible Partner	ELION
Status (proposed/approved)	proposed
Implemented/not implemented	not implemented
Status of the implementation	Specification
Executable or Library/module (Support)	Specification
Single Thread or Multithread	Multithread
Language of Development	C#
Platforms supported	Windows
Reference to the AXFW location of the source code demonstrator	N/A
Reference to the AXFW location of the demonstrator executable tool for internal download	N/A
Reference to the AXFW location of the demonstrator executable tool for public download	N/A
Address for accessing to Web Services if any, add accession information (user and Password) if any	Not yet available
Test cases (present/absent)	Absent
Test cases location	Not yet available
Usage of the AXMEDIS configuration manager (yes/no)	no
Usage of the AXMEDIS Error Manager (yes/no)	no
Major Problems not solved	-- Not yet available

Major pending requirements	--	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
User authentication	AXCS	
Licensing	PMS	
Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
AXCACHE		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

4.6.1 General description of the module

ELAXAPI_WS module is single point interfacing module that will be created to provide all usage related services to serve as plug-in for making the presentation as straight forward as possible on heavy load hot.ee portal and to avoid any possible software conflicts.

4.6.2 Module design in terms of classes

Elaxapi_ws	Elaxapi_ws::AXListItem
+IsEnabled() +RegisterUser() +GetList() +GetLicense() +ValidLicense()	+Link:String +AXOID:String +Description:String +Price:Double

ELAXAPI_WS is primary wrapper class containing all functions and methods needed to interface with all used AXMEDIS tools (PMS, AXCS) and also to provide content listings to hot.ee browsing page in a form of an array of AXLISTITEM class that can be parsed into usable format for end user presentation.

4.6.3 Module database design

Module uses Microsoft SQL 2005 Express engine for local data storage, the initial database structure is as follows;

transactionLog	publishedObjects	epassUserBind
	PK <u>AXOID</u> filename description price	PK <u>epassUser</u> AXUID
epassUser AXOID Price TransactionDate		

4.6.4 Module WDSL specification

Initial WDSL specification is as follows;

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="https://axmedis.neti.tv/elaxapi_ws/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="https://axmedis.neti.tv/elaxapi_ws/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="https://axmedis.neti.tv/elaxapi_ws/">
      <s:element name="IsEnabled">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="epassUser" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="IsEnabledResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="IsEnabledResult" type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="RegisterUser">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="epassUser" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>

```

```

<s:element name="RegisterUserResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RegisterUserResult" type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetList">
  <s:complexType />
</s:element>
<s:element name="GetListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetListResult" type="tns:ArrayOfAXListItem"
/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ArrayOfAXListItem">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="AXListItem" nillable="true"
type="tns:AXListItem" />
  </s:sequence>
</s:complexType>
<s:complexType name="AXListItem">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Link" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="AXOID" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Price" type="s:float" />
  </s:sequence>
</s:complexType>
</s:schema>
</wsdl:types>
<wsdl:message name="IsEnabledSoapIn">
  <wsdl:part name="parameters" element="tns:IsEnabled" />
</wsdl:message>
<wsdl:message name="IsEnabledSoapOut">
  <wsdl:part name="parameters" element="tns:IsEnabledResponse" />
</wsdl:message>
<wsdl:message name="RegisterUserSoapIn">
  <wsdl:part name="parameters" element="tns:RegisterUser" />
</wsdl:message>
<wsdl:message name="RegisterUserSoapOut">
  <wsdl:part name="parameters" element="tns:RegisterUserResponse" />
</wsdl:message>
<wsdl:message name="GetListSoapIn">
  <wsdl:part name="parameters" element="tns:GetList" />
</wsdl:message>
<wsdl:message name="GetListSoapOut">
  <wsdl:part name="parameters" element="tns:GetListResponse" />
</wsdl:message>
<wsdl:portType name="elaxapi_wsSoap">
  <wsdl:operation name="IsEnabled">
    <wsdl:input message="tns:IsEnabledSoapIn" />

```

```

    <wsdl:output message="tns:IsEnabledSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="RegisterUser">
    <wsdl:input message="tns:RegisterUserSoapIn" />
    <wsdl:output message="tns:RegisterUserSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetList">
    <wsdl:input message="tns:GetListSoapIn" />
    <wsdl:output message="tns:GetListSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="elaxapi_wsSoap" type="tns:elaxapi_wsSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="IsEnabled">
    <soap:operation soapAction="https://axmedis.neti.tv/elaxapi_ws/IsEnabled" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RegisterUser">
    <soap:operation soapAction="https://axmedis.neti.tv/elaxapi_ws/RegisterUser" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetList">
    <soap:operation soapAction="https://axmedis.neti.tv/elaxapi_ws/GetList" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="elaxapi_wsSoap12" type="tns:elaxapi_wsSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="IsEnabled">
    <soap12:operation soapAction="https://axmedis.neti.tv/elaxapi_ws/IsEnabled" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RegisterUser">
    <soap12:operation soapAction="https://axmedis.neti.tv/elaxapi_ws/RegisterUser" style="document" />
    <wsdl:input>

```

```

    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetList">
  <soap12:operation soapAction="https://axmedis.neti.tv/elaxapi_ws/GetList" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="elaxapi_ws">
  <wsdl:port name="elaxapi_wsSoap" binding="tns:elaxapi_wsSoap">
    <soap:address location="http://localhost:50652/axmedis/elaxapi_ws.asmx" />
  </wsdl:port>
  <wsdl:port name="elaxapi_wsSoap12" binding="tns:elaxapi_wsSoap12">
    <soap12:address location="http://localhost:50652/axmedis/elaxapi_ws.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

4.6.5 User interface description

Module is web service and has no user interface

4.6.6 Technical and installation information

To be determined during the course of the project.

4.6.7 Draft user manual

To be done in the course of the project.

4.6.8 Example of usage

ELAXAPI_WS module is responsible for providing single interface for performing all AXMEDIS related operations to hot.ee portal.

4.6.9 Integration and compilation issues

To be determined during the course of the project.

4.6.10 Configuration Parameters

To be determined during the course of the project.

Configuration parameter	Possible values

4.6.11 Errors reported and may occur

To be determined during the course of the project.

Error code	Description and rationales

4.6.12 Formal description of algorithm <.....>

name	
Method	
Description	
Input parameters	
Output parameters	

5 Bibliography

[1] AXMEDIS project website, www.axmedis.org

6 Glossary

A

API, Application Programming Interface

A set of definitions of the ways one piece of computer software communicates with another.

ASP, Active Server Pages

A web-scripting language by Microsoft.

B

B2B, Business to Business

Refers to one business communicating with or selling to another.

B2C, Business to Consumer

Refers to a business communicating with or selling to an individual rather than a company.

C

CM, Content Management

Designs the set of processes and technologies supporting the evolutionary life cycle of digital information.

CMS, Content Management System

A software tool designed to help content managers create, manage, and publish their content.

CRM, Customer Relationship Management

An information industry term for methodologies, software, and Internet capabilities that help an enterprise manage customer relationships in an organized way.

D

DAM, Digital Asset Management

DRM, Digital Rights Management

A system for protecting the copyrights of digital content.

F

FTP, File Transfer Protocol

A protocol used to transfer files over a TCP/IP network.

G

GNU, Gnu's Not UNIX

A project sponsored by the Free Software Foundation that develops and maintains a complete software environment.

GPL, GNU General Public License

The license that accompanies the GNU software from the Free Software Foundation.

GUI, Graphical User Interface

A computer terminal interface based on graphics instead of text.

H

HTTP, HyperText Transport Protocol

The communications protocol used to connect to servers on the World Wide Web.

HTTPS, HyperText Transport Protocol Secure

Version of the HTTP protocol that enables secured transmission of Web pages.

HTML, HyperText Markup Language

The document format language used on the World Wide Web.

I

IP, Internet Protocol

The network layer protocol in the TCP/IP communications protocol suite.

J

J2EE, Java 2 Enterprise Edition

A version of Java for developing and deploying enterprise applications.

L

LDAP, Lightweight Directory Access Protocol

A protocol used to access a directory listing.

N

.NET

Microsoft's framework for Web services and component software.

Q

QoS, Quality of Service

S

SSL, Secure Sockets Layer

A protocol developed by Netscape for transmitting private documents on the Internet.

SOAP, Simple Object Access Protocol

A message-based protocol based on XML for accessing services on the Web.

STB, Set-Top-Box

AXMEDIS Project

U

UDDI, Universal Description Discovery and Integration

An industry initiative for a universal business registry of Web services.

UML, Unified Modeling Language

An object-oriented analysis and design language.

URI, Uniform Resource Identifier

The addressing technology for identifying resources on the Internet.

URL, Uniform Resource Locator

A type of uniform resource identifier (URI) that uses an HTTP connection.

V

VoD, Video on Demand

W

WCM, Web Content Management

WSDL, Web Services Description Language

An XML-based language (Developed by Microsoft and IBM) for defining Web services.

X

XHTML, eXtensible HyperText Markup Language

A reformulation of HTML markup language with the same expressive possibilities but a stricter syntax.

XML, eXtensible Markup Language

A metalanguage used to design markups languages.