



Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE9.3.4

Integrated prototype of content production and distribution in push and on-demand for I-TV

Version: 1.5

Date: 17/04/2007

Responsible: EUTELSAT (revised and approved by project coordinator)

Project Number: IST-2-511299

Project Title: AXMEDIS

Deliverable Type: Public

Visible to User Groups: YES only the document and the demonstrations

Visible to Affiliated: YES only the document and the demonstrations

Visible to Public: YES only the document and the demonstrations

Deliverable Number: DE9.3.4

Contractual Date of Delivery: M30

Actual Date of Delivery: 17/04/2007

Work-Package contributing to the Deliverable: WP9.3.1, WP9.3.2

Task contributing to the Deliverable: WP9.3

Nature of the Deliverable: Documentation for promoting and presenting demonstrator

Author(s): EUTELSAT, MBI

Abstract:

This is a report on the current status of the prototype for demonstrating the content distribution via satellite, to PC and STB.

Keyword List: Distribution channel, content, satellite, i-TV, PC, STB, Programme&Publication, prototype, validation phase

Table of Contents

1	EXECUTIVE SUMMARY AND REPORT SCOPE	3
2	INTRODUCTION	3
2.1	SYSTEM OVERVIEW	4
2.1.1	Actors	5
2.1.2	Integration with AXMEDIS components.....	6
2.1.3	Passing from AXMEDIS Object to OPENSky Package on the server side	6
2.1.3.1	Content Upload.....	6
2.1.3.2	Content Packaging.....	6
2.1.3.3	Content Addressing	6
2.1.3.4	Content Sending	7
2.1.3.5	Content Transmission	7
3	PROTOTYPE FOR CONTENT DISTRIBUTION IN PUSH AND ON-DEMAND FOR I-TV	7
3.1	SENDER SIDE	7
3.1.1	Programme and Publication.....	7
3.1.2	API for content packaging and scheduling	8
3.2	RECEIVER SIDE.....	9
3.2.1	OPENSky Client GUI.....	9
3.2.2	The ActionManager Module	10
4	PROTOTYPE FOR CONTENT DISTRIBUTION TOWARDS STB.....	13
4.1	STB DESCRIPTION	13
4.1.1	Hardware	13
4.1.2	Software	14
4.2	PUSH MODEL.....	15
4.3	GRAPHICAL USER INTERFACE	15
4.4	MEDIA TYPES.....	16
4.5	CONTENT INDEX FILES.....	17
5	DEMONSTRATION AND VALIDATION	19
5.1	DEMONSTRATION VIDEO.....	19
5.2	RESEARCH GUIDELINES UPDATE	19
5.3	PLAN FOR DEMONSTRATION AND VALIDATION PHASES	19
5.3.1	STB phasing	19
5.3.2	Demonstrator Implementation.....	20
6	DEMONSTRATOR FACT SHEETS	20
6.1	DEMONSTRATOR FACT SHEET FOR SATELLITE DISTRIBUTION TO PCs	20
6.2	DEMONSTRATOR FACT SHEET FOR SATELLITE DISTRIBUTION TOWARDS STB	24
6.3	DEMONSTRATOR FACT SHEET FOR SATELLITE DISTRIBUTION TO KIOSKS.....	26
7	BIBLIOGRAPHY	30

1 Executive Summary and Report Scope

This document represents an accompanying documentation to the Prototype for content distribution in push, as defined in the WP9.3. It contains a description of the integrated prototype for content distribution via satellite data broadcast, delivered as DE9.3.4.

Actually, two prototypes are presented here, one targeting PC-based clients, the other targeting the Set-Top-Boxes. Prototypes are based on the specifications resulting from WP4 and WP5, in particular WP4.8.

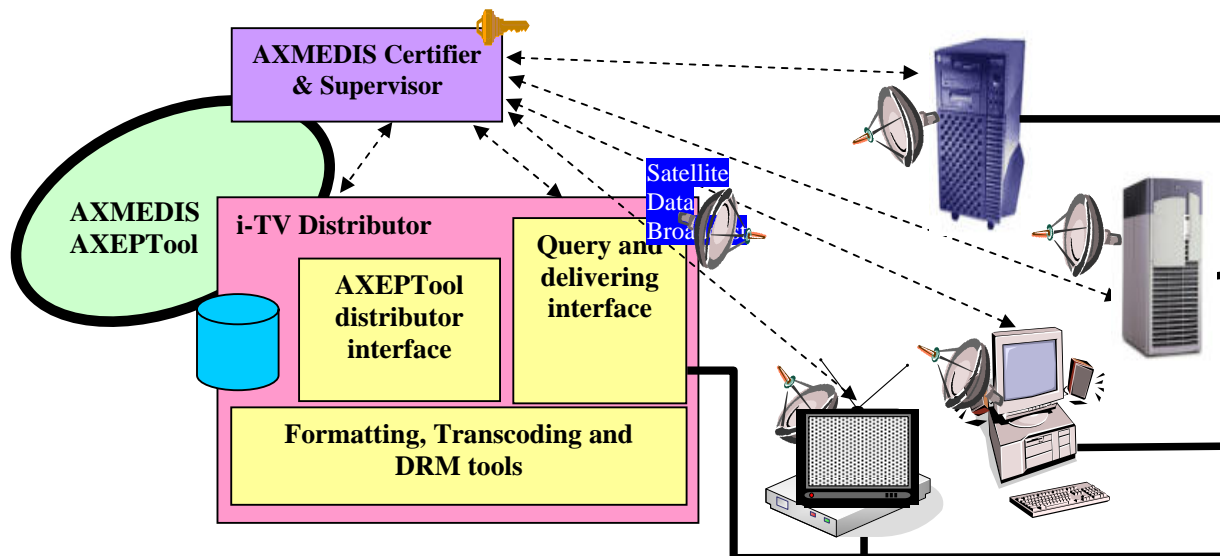
2 Introduction

Satellite Data Broadcast is a content distribution mechanism that enables the distribution of the AXMEDIS content in a very efficient manner. This technology, provided by EUTELSAT OPENSKEY platform, allows large quantities of data to be pushed via satellite directly on the user's PC without congesting local networks. The PUSH mechanism can be used, again, to renovate the catalogue of the Distributors periodically at low cost.

The first prototype for the Satellite Data Broadcast inside the AXMEDIS environment, focused on the content delivery towards PCs equipped with a DVB/IP satellite card, was called in this context i-TV.

Now a second kind of receiving device is included in the demonstration of the satellite distribution, the Set-Top-Box (STB). The STB developed for the AXMEDIS purpose is of an evolved kind including a hard disk for data storage and a guide for selecting programs to be rendered on a TV screen.

This integrated prototype supports the complete cycle performed by a new content to be distributed by the Content Provider to a large number of Content Distributors, as well as from a Distributor to the final users: this proves how the platform is suitable for both B2B and B2C cases.



AXMEDIS Distribution of i-TV Content via satellite data broadcast

The distribution process uses the PUSH transmissions of the content according to the protocols that have been previously defined.

The Distributors act as proxies for the content, making it available for the clients without the need to contact the Content Provider directly. The integrated prototype uses the Programme and Publication (P&P) module of the AXMEDIS platform to prepare the content and send it through the satellite channel. This tool interfaces with the API of the OPENSKEY system in a transparent manner to the Distributor; advantages are

evident, since the Distributor can use a single tool/interface to create its transmissions and then choose the satellite channel as well as one of the other distribution channels.

On the client side, the demonstrator focuses on the B2C distribution, with a typical scenario of transmission of contents from a Distributor to some i-TV Clients. Client stations are able to store the received data in a cache area, where additional applications can access the data and handle it, likely for filtering, statistics, etc. The other target of the distribution, the STB, is then presented in this document; the STB here documented includes the modules for rendering the AXMEDIS objects that are received via PUSH.

The prototype validates the distribution in multicast via PUSH technologies of a new multimedia content. The key elements for this system are:

- Content Provider station,
- Adaptation server,
- Satellite network infrastructure,
- Distributor station,
- i-TV client (PC equipped with a DVB/IP Satellite Card),
- Set-Top-Box (STB).

2.1 System overview

This section provides an overview of the Satellite Data Broadcast system presented per layers, from the bottom:

1. **Transport:** it is responsible of send/receive operations (open socket, create/join multicast group, packetize/reassembly files).
2. **Server/Client Communication (Guide):** it is entrusted with the communication from the server towards all connected clients. It sends the content program guide - EPG (periodically refreshed), or, alternatively, it manages the Event Signalling Channel, where the server announces all incoming transmission start dates. From the client towards the server, this level manages the packet reparation in unicast.
3. **API:** it is a very important level, since it allows Publishers to directly interact with the PUSH Technology (it is valid for both server context and client context).
4. **Web Interface:** it offers a graphical web interface to Publishers. It implements its functionalities by calling the lower API Level.
5. **AXMEDIS Interface:** this level manages the integration between AXMEDIS and OPENSky platform (i.e., this level includes the Programme&Publication Tools at the server side and the AXMEDIS Players at the client side).

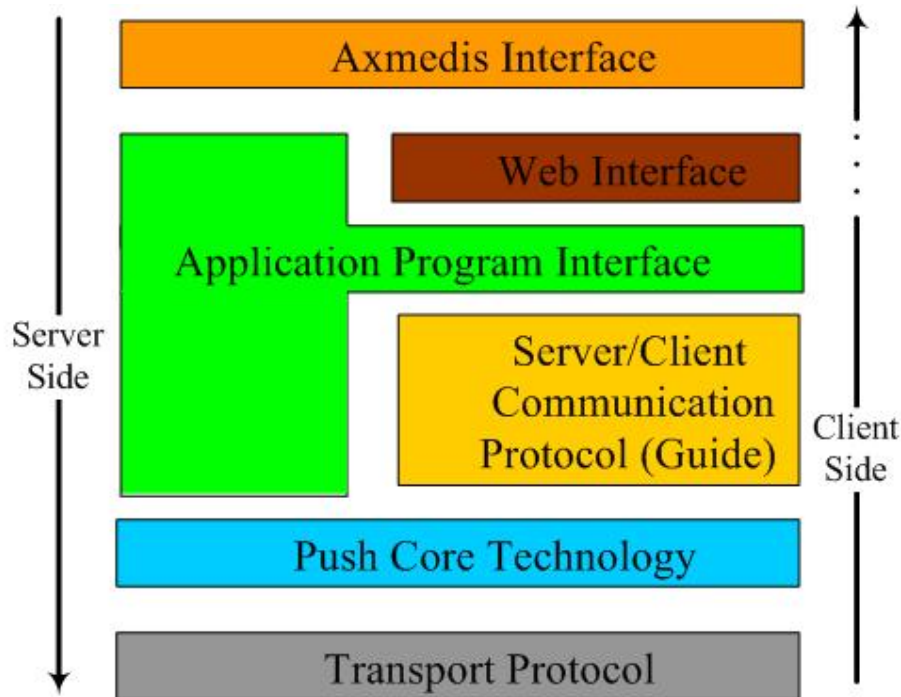


Figure 2: Satellite Data Broadcast Internal Layers

2.1.1 Actors

The entities that play a role in the distribution through via satellite data broadcasting, as well as their usage name and functions, are listed once again here for clearness:

- **Publisher** (AXMEDIS Distributor/Content Provider)
 - Create and upload the content
 - Schedule transmissions
- **Broadcaster** (EUTELSAT using OPENSKEY Platform)
 - Assign bandwidth capability and access to the platform to Publishers
 - Send content
 - Owner of technology
 - Responsible of transport level
- **Final User**
 - Receive broadcast content
- **Group**
 - Allow addressing group of users
- **Package**
 - Minimal logical unit transmission
 - Contain one or more files
- **Program/Schedule**
 - Transmit one or more package following a pre-defined sequence (Carousel)
 - Use the assigned bandwidth to its schedule
 - Define transmission details (bandwidth, time, # repetition, mode)
- **Job**
 - Transmission instance for a given package, according to the schedule

2.1.2 Integration with AXMEDIS components

The Satellite Data Broadcast gives to Publishers the total control of their micro platform of multicast file delivery. A Publisher interacts with the system either by web access or by sending XML-structured commands (treated by the API Server), in this last case by using the Programme&Publication Tools (see section 3.1.1).

Each Publisher has access to the Storage Area where he stores his uploaded content. This area is independent from the Push System, and the Publisher accesses it by means of tools like ftp.

Once content has been loaded into the Storage Area, the Publisher may manage Packages (grouping files taken from his Storage Area), create Programs (grouping Packages previously created), manage Groups of Users authorised to receive Packages, and schedule transmissions of his Programs.

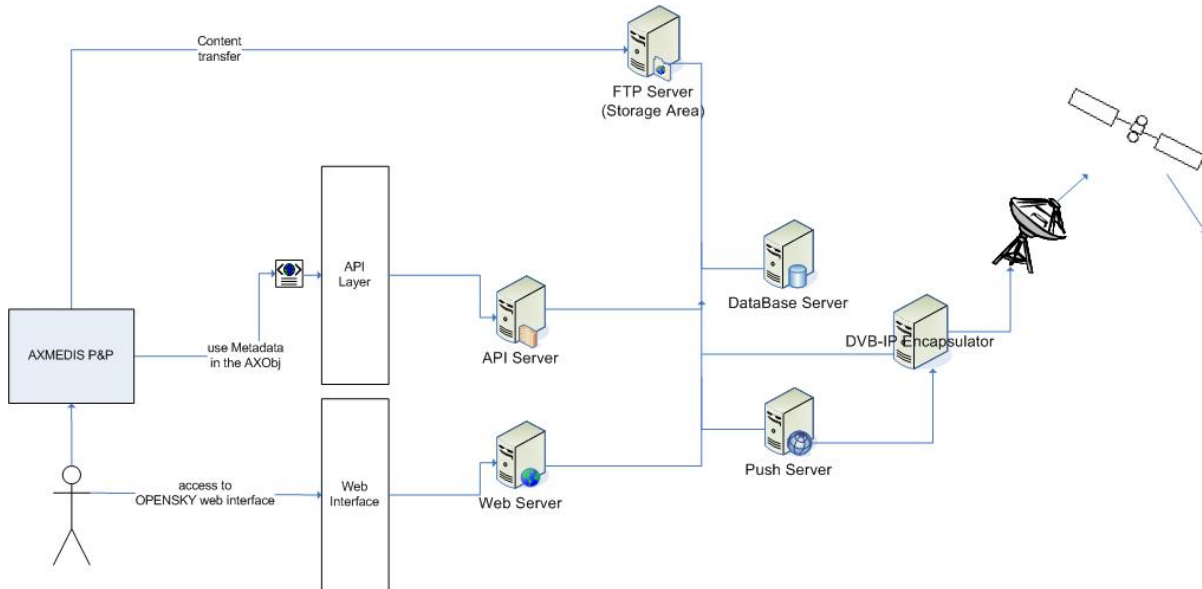


Figure 3: AXMEDIS-OPENSKY Platform Architecture

2.1.3 Passing from AXMEDIS Object to OPENSKY Package on the server side

As seen in the previous sections, the SDB platform and protocol are based on specific concepts, namely the package. This section provides a specification of the steps required for addressing content to the defined receiver, starting from the collection of content, packaging of AXMEDIS Objects into OPENSKY Packages, choosing the target, scheduling the transmission.

These operations are possible through the existing web interface of the OPENSKY platform or using a set of API (see section 3.1.2).

2.1.3.1 Content Upload

The content intended to be sent have to be uploaded on the OPENSKY platform. This can be done via simple HTTP or, more likely, via FTP account to a specific account.

2.1.3.2 Content Packaging

Once the AXMEDIS Objects are uploaded in the OPENSKY Platform, they are associated to the OPENSKY Package, defining some transmission parameters as the FEC, as well as the target group.

2.1.3.3 Content Addressing

The content can be addressed to a specific set of authorized users. Groups can be defined at the beginning, or dynamically using the OPENSKY Group Management Protocol (OGMP): adding/removing from a given group is immediately notified to the user.

2.1.3.4 Content Sending

One or more instances of a single Package can be added to the Carousel sequence in order to create the Base for the Carousel. Then a Schedule is created taking into account all possible variables:

1. Bandwidth (from 64 Kbps to 20 Mbps)
2. Start Date/ End Date of the Program (during 3 months or 1 hour)
3. Slot Length is a parameter related to the Program duration and the Periodicity. It is less than the Program duration (i.e., program_duration=3 months; slot_length=1 hour)
4. Periodicity is greater than the Slot length value. There is a correlation between Periodicity and Slot length (i.e., slot_length=1 hour, periodicity=1 day, mean that transmissions will be on air 1 hour per day)

Example:

Program_name	=	P1
bandwidth	=	512 Kbps
program_duration	=	3 months
slot_length	=	1 hour
periodicity	=	1 day

Previous values mean that during 3 months there will be an active Program P1, transmitting 1 hour per day at 512 Kbps.

2.1.3.5 Content Transmission

This module is exclusively internal to the Satellite Data Broadcast system. Details of the transmission layer are not specified here.

3 Prototype for Content distribution in push and on-demand for i-TV

The prototype focuses on the two sides of the transmission: the sender and the receiver; for the first side of transmission, the packaging of AXMEDIS Objects into OPENSKEY Packages before, and the program creation and scheduling then.

In parallel, on the receiving station, the demonstrator validates how content transmissions are presented on the Electronic Program Guide, then selected for download or even automatically downloaded, according to the transmission parameters, and stocked on a predefined Cache Area for fruition or post-reception handling.

3.1 Sender Side

The integrated prototype of Satellite Data Broadcast for content distribution in push replaces the previous web-based graphical interface with the Programme and Publication (P&P) tool. Using the OPENSKEY APIs, the P&P easily performs all the steps required to map AXMEDIS content into OPENSKEY packages and programs, to create and manage groups and security parameters, and to define transmission scheduling.

3.1.1 Programme and Publication

The OPENSKEY distribution system is based on the concept of packages and programs that do not have exactly a correspondent in the AXMEDIS platform. As explained in previous documents of this WP9.3, the usual procedure for this satellite distribution system asks to transform the content an OPENSKEY package, define for such a package some properties and parameters for the transmission, define target groups allowed to receive, create a program and defining its content (carousel), schedule the program for effective sending. All these steps may appear complicate and of few interest for a Content Distributor, especially when he/she has to cope with several distribution channels, any of which having its own specific constraints and concepts.

The Programme and Publication is the module of the AXMEDIS platform that aims to consolidate the multiple distribution channels under a same tool for what concerns the preparation and schedule the content.

Through the unique AXMEDIS Programme and Publication Editor interface it is possible to select from an AXDB the content to be sent, choose the channel (the EUTELSAT satellite channel in our case) and the date of delivery of the content.

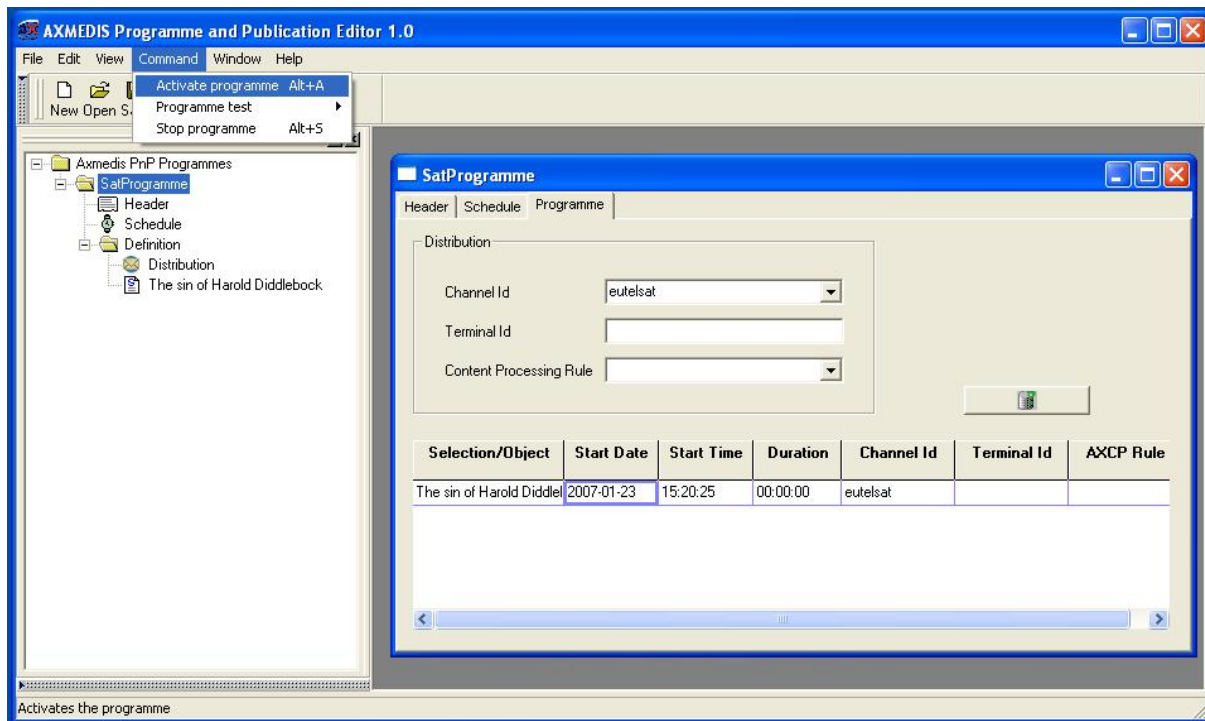


Figure 1: Programme and Publication Editor

Behind the scenes, the P&P performs the steps typical of the satellite distribution. Some assumptions have been taken to simplify the group management: at present we assume that content is addressed to a same predefined group (let's say corresponding to the collectivity of satellite users). The scheduling section is already managed by the P&P Editor, with its functionalities to activate/stop programmes.

3.1.2 API for content packaging and scheduling

The API of the EUTELSAT Push system includes functions for content packaging and program scheduling and activation. They are used in the integration of the AXMEDIS P&P tool with the Satellite Distribution Channel. With the integration, once a Distributor defines a Programme and chooses the satellite channel for the delivering, the calls to the API will prepare the corresponding packages and programs on the OPENSky platform, and finally activate the distribution.

The APIs use an XML protocol encapsulated in HTTPS protocol, and their data format is XML.

The Eutelsat Database clients send request to a server provider that asks for data the Database access server. Both the Database access server and Eutelsat Database are owned and administrated by Eutelsat S.A.

Example:

The following lines provide an example of how calling a method of Push Server API, and how to parse the response received from the API Server.

A complete set of public functions and their possible responses (successful/unsuccessful) are contained in the document *API Push Server public functions* [APS].

Definition: create_package

Function to create a package.

Params	Notes
Name	Package name
Description	Package description

Request

```

<dbrequest>
  <administrator login="my_login" password="my_password"/>
  <operation name="create_package">
    <param name="name">packagename</param>
    <param name="description">the package description</param>
  </operation>
</dbrequest>

```

Response

```

<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <operation name="create_package">
    <row>
      <code>0</code>
      <message>Package created</message>
    </row>
  </operation>
</response>

```

3.2 Receiver Side

The reception is then validated on a client station previously installed with needed material, i.e. DVB card connected to a dish.

The use of the P&P tool does not change the receiving part of the demonstration. As underlined before, the Programme and Publication is just one of the ways to interact with the satellite platform, automating all the server side operations. Thus the following description is applicable also to previous demonstrators, and in any case to any interface used to package and send content: Web Interface, APIs, P&P.

3.2.1 OPENSKY Client GUI

The reception of the content is performed by the OPENSKY OSClient application. A module called OSListener is the routine that listens for multicast transmissions and download them, when they target the considered station, but a graphical interface allows to browse the content on the carousel and to select the preferences for download, as well as to see what has already been received.

The AXObjects sent with the programme are in this way received on the enabled station, and stocked in the AXMEDIS cache area for further filtering. The pictures below show the GUI of the client application from which is possible to know which content is on air, what is currently download, and the content already received.

Name	Start Time	Duration	Size	Speed	Target
axtest1	12:07	9s	153 KB	16 KB/s	+
radio	12:07	20s	332 KB	16 KB/s	+
verdi	12:08	1m 34s	1.47 MB	16 KB/s	+
dog	12:10	36s	581 KB	16 KB/s	+

Figure 2: Program Guide with on-air content

Content Item	File Name	Size	Timestamp
dog	dog AXObj	581 KB	28/09/05 12:02
radio	radio AXObj	332 KB	28/09/05 11:59

Figure 3: List of received content

Note that this graphical application is used to better explain how the reception works. It is a merely visual aid on the demonstration. The client application for the satellite data broadcast proposed by EUTELSAT includes this GUI, but, of course, it is not essential. The core of the client application is composed by other components (likely the OSListener module) that run in background and automatically let the receiving station download and store the content, without the necessity of a human interaction (except for the initial configuration phase).

3.2.2 The ActionManager Module

The prototype answers to the necessity of caching together the received AXMEDIS Objects. On the server side, AXMEDIS Objects are wrapped for transmission inside an OPENSKY Package, and with this format they are received on the client side. As the following picture shows, the received OPENSKY package, stored on the content area of the client, contains the pk_info.os with transmission metadata and the real AXMEDIS content (one or more files indeed).

The goal is to unwrap this content from OPENSKY package, and move it to a common specific *cache area*, where all AXMEDIS Objects are stored.

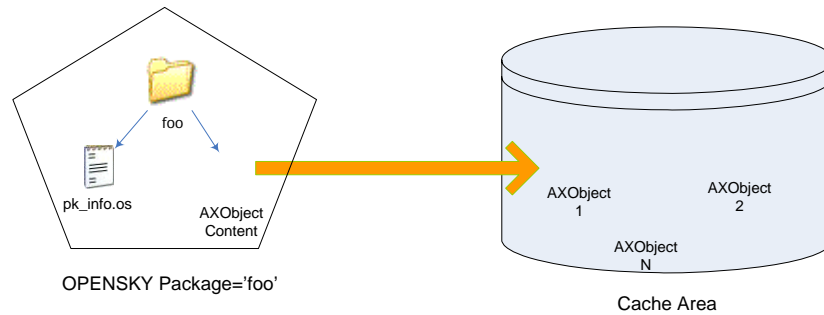


Figure 4: Action Move on a AXObject

The location of the cache area is specified in the configuration files used by the client application. Once the AXMEDIS Objects are stored in a common area, they are available for direct fruition using the Axmedis Player, or can be accessed there for further content filtering and other post-reception handling.

The module in charge of this operation is the Action Manager. It is a module of the OPENSky OSListener, that is responsible for handling post-reception actions at the reception of new contents.

The agent entrusted of the content download, the Job, is also responsible to fire the corresponding actions.

Once the content reception is completed without errors, the Job class checks if any action is associated to the received package, and, if it the case, it instantiates the Action Manager classes responsible of these actions and invokes the execution of the actions.

The Job class implements this part of post-reception as follows:

1. Parse the package information field from the guide
2. Try to instantiate the appropriate Action Manager for each action parsed (the post-reception action may not always have an appropriate Action Manager)
3. Retrieve the action properties and store them within the Action Manager (destination folder, email address, ...)
4. Try to execute the action on the content
5. If action succeeds, execute following actions, otherwise just stop processing anymore actions.

One of the actions associated to the content is the one that move the received content on a specified folder (MoveAction), as in the case of the AXMEDIS Objects, where the ActionManager module is automatically called when an OPENSky package identified as an *axmedis* type is downloaded.

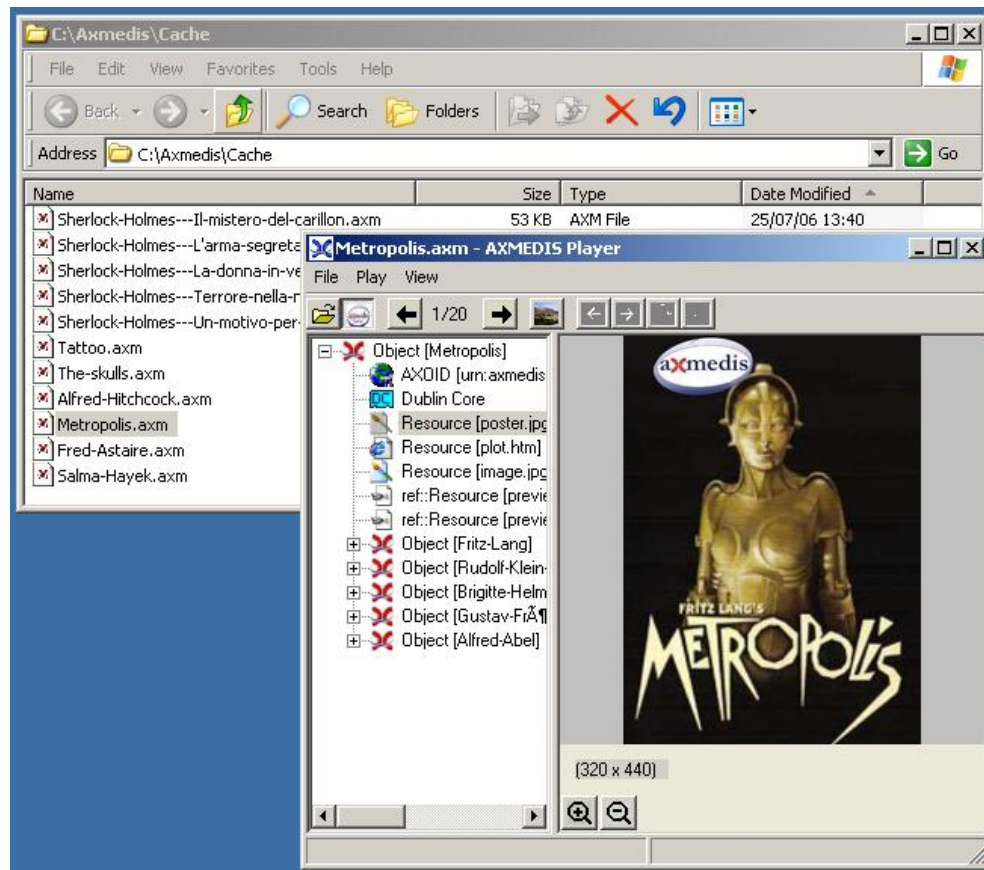


Figure 8: Fruition of cached AXMEDIS objects

4 Prototype for Content distribution towards STB

This prototype is dedicated to the distribution of AXMEDIS content to the AXMEDIS STB client device. The OPENSKEY server side of the distribution remains virtually unchanged, or with minimal adaptations, among the various client devices. In this chapter we will thus focus particularly on the client side.

The AXMEDIS STB platform, for this demonstrator, is based on a specific STB (provided by MBI) including AXMEDIS technology which is able to receive multimedia content (AXMEDIS, MPEG-21, MPEG-4 content) via satellite and make it accessible to the users via a typical consumer STB user interface. The STB system is made up of hardware and software elements: within the STB activity the existing software components are adapted to allow the integrated system to implement the functions needed for the fruition of the different AXMEDIS media. The STB hardware components are purchased from external manufacturers. The key activity, in the STB context, is the integration of the AXMEDIS Object Model (AXOM).

4.1 STB description

The Set-Top-Box (STB) is a x86 based CPU Set-Top device with integrated chipset for hardware video decoding and DVB satellite front-end. The software is based on Linux Operating System. In the figure below a view of the STB:



Figure 9: The STB

4.1.1 Hardware

The STB is currently based on the following hardware specification:

- VIA C3 CPU running at 600Mhz
- Sigma Designs EM8471 hardware MPEG-1/2/4 decoder
- OSD support with 256 colors for scrolling text on TV
- Outputs: SCART, S -video, C-video, SP/DIF, L/R audio with RCA jacks
- Ethernet 10/100 Lan port
- USB port
- DVB-S interface
- 256 MB DDR 266 RAM
- 80 GB Hard Disk
- IR Remote Control
- V.92 PSTN Modem

- External switching power supply, 110/240, 80W peak
- Optional IR Remote keyboard

4.1.2 Software

The STB runs the Linux Debian distribution version 3.1. The Graphical User Interface is based on the X-Window System (v.4.3.0.1) running on the On Screen Display (OSD) provided by the Sigma video chipset. On top of the Linux layer MBI have developed their middleware and their GUI as the following picture.

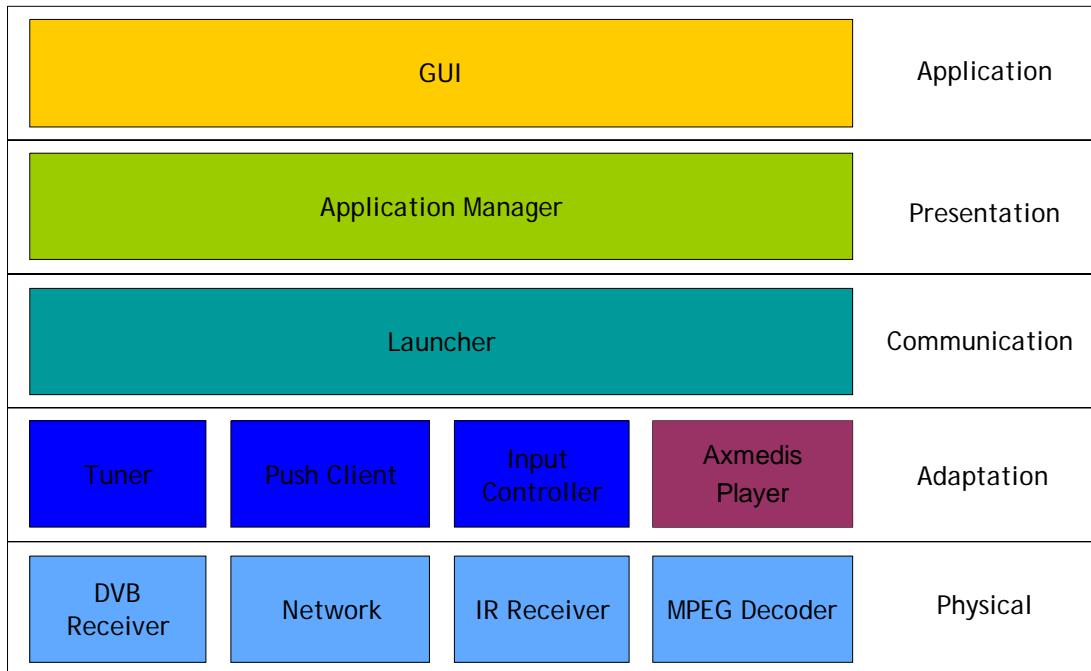


Figure 10: MBI STB software stack

From the Adaptation to the Presentation layers there is a set of modules to provide all the functions of the STB. Each module is highly independent from each other, the interface paradigm is based on asynchronous messages travelling on TCP/IP channels. The general structure is highly hierarchical: each module is linked to one parent module and possibly one or more children: In such way each module knows how to communicate with its peers, but has no knowledge of their implementation. The use of standard protocols (TCP) for the interconnection of the modules, makes it possible to have them implemented in different languages (i.e. C, C++, Java, Python, etc).

At the data level the DVB Receiver allows the STB to tune to a satellite transponder and receive data from selected DVB Pids. The DVB software driver also presents a network interface (dvb-net) for the incoming IP data encapsulated in DVB streams via Multi Protocol Encapsulation (MPE). The Network interfaces of the STB transmit/receive IP data via Ethernet, Pstn modem and dvb-net (receive only). With the Infrared Receiver driver the STB receives keys from a Remote Control or IR keyboard, while the MPEG decoder module allows the hardware accelerated decoding of the video from both file or stream in the formats specified further down in [4.4].

The roles of each module will be briefly described here:

launcher	routes of the different messages among the modules. Controls power status modes such as STB stand-by, on and off
inputcontroller	produces Dintv input messages from input devices such as remote control or keyboard
Ax player	middleware module integrating the AXOM library. It feeds the Sigma Designs MPEG2/4 hardware decoder with the clear data from the AXOM once the protection processing, including the interaction with the PMS, has been done
tuner	responsible for tuning and demuxing the various sources, e.g. dvb front end, and making the elementary contents accessible
appsmanager	manages the user side applications at various levels: start/stop, message dispatching, focus and input handling; it's the only interface to the rest of the system for user applications
gui	resident system-type application that provides the bottom-screen banner and lists for content navigation, content description, tracks selection, etc.

4.2 Push model

The AXMEDIS STB, for this demonstrator, acts as a slave device of the Push service. The OPENSKEY Listener (OSListener) runs as a background task on the STB and receives the Push packages that are targeted to the STB device it is running on. The reception of a package is enabled if the device belongs to a Group to which the package is delivered. A device may belong to multiple Groups. The STB can react automatically upon reception of a package, decompress it if necessary, and present the new contents to the user. It is also possible to clear a content from the STB, by sending an empty package with the same name of the one that delivered the content. This results in a powerful means of handling the media list of the STBs, in defined Groups, entirely from the Distributor, or Service Provider side.

4.3 Graphical User Interface

The STB does not show the GUI available for PC-based device, as described in [3.2.1], which is typical for a PC user, but would be completely inadequate for a STB user. The STB Push model as described above, implies that the STB retains its typical consumer-electronics user interface made up of menus, lists and banners, while the Push reception activity runs in background. The figure below shows an example of list that allows the user to navigate the contents via remote control.



Figure 11: Example of content guide for STB

4.4 Media types

The STB platform is based on a hardware accelerated decoding of the various media. This implies that some limitations exist with regard to the media types that the STB can handle. The following is a list of media types, in terms of codecs and container formats, currently supported by the MBI STB.

Video types	Notes
MPEG-4	MPEG-4 Advanced Simple profile Level 5 (without ¼ pixel and Global Motion Compensation). Rectangular shape video decoding up to CCIR 601 resolution (720x480 or 720x576), support for B pictures.
MPEG-2	Main Profile @ Main Level.
MPEG-1	System.

Audio types	Notes
MP3	MPEG-1 Audio Layer III
MPEG1	MPEG-1 Audio Layer I and II
MPEG-4	MPEG-4 Audio High Quality Profile Level 2
AC3	Dolby Digital down mixed to 2 channels

Container types	MIME type	Resource extension	file	Notes
MPEG-4	video/mp4	mp4		ISO/IEC 14496-14.
AVI	video/mp4	avi		Allowed FourCC codes are: DIVX, DX50, MP4V, XVID.
MPEG-2 Transport	video/MP2T	m2t, m4t		ISO/IEC 13818-1, file extension according to video codec used: m2t for MPEG2 video, m4t for MPEG4 video.
MPEG-2 Program	video/MP2P	m2p, m4p		ISO/IEC 13818-1, file extension according to video codec used: m2p for MPEG2 video, m4p for MPEG4

			video.
VOB	video/MP2P	vob	Only MPEG2 codec.
MPEG-1	video/mp1s	mpg, m1s	ISO/IEC 11172-1.
MP3	audio/mpeg	mp3	ISO/IEC 11172-3.
PCM Audio	audio/PCMA	wav	Uncompressed wav.

4.5 Content index files

The STB creates a list of media that the user can navigate by processing one or more special index XML files named contents*.xml (being * any sequence of characters allowed in a file name). The STB parses an index files tree starting from the “root” contents.xml file. Those index files can be automatically detected and parsed when they are received via the OPENSky Push. In this way the STB is able to keep the navigation list up to date against the contents delivered via the satellite distribution.

Following is the XML schema used for the contents*.xml files.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="tuner">
    <xs:annotation>
      <xs:documentation>Contents and sources</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="contents">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
              <xs:element name="content">
                <xs:complexType>
                  <xs:sequence minOccurs="0" maxOccurs="unbounded">
                    <xs:element ref="parameter"/>
                  </xs:sequence>
                  <xs:attribute name="name" type="xs:string" use="required"/>
                  <xs:attribute name="id" type="xs:integer" use="required"/>
                  <xs:attribute name="delivery" use="required">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="LIST"/>
                        <xs:enumeration value="FILE"/>
                        <xs:enumeration value="RTSP"/>
                        <xs:enumeration value="DVB"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:attribute>
                  <xs:attribute name="type" type="xs:string" use="required"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="sources" minOccurs="0">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="source">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
              <xs:element ref="parameter"/>
            </xs:sequence>
            <xs:attribute name="name" type="xs:string" use="required"/>
            <xs:attribute name="type" use="required">
```

```

        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="transport"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="others" minOccurs="0">
  <xs:complexType>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="other">
        <xs:complexType>
          <xs:attribute name="url" type="xs:anyURI" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="parameter">
  <xs:complexType>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="value" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```

In the following lines an example of content.xml file.

```

<?xml version="1.0"?>
<tuner>
  <contents>
    <content name="Ax MP4 clip" id="101" delivery="FILE" type="M4P">
      <parameter name="file" value="file_mp4.axm"/>
      <parameter name="encryption" value="axmedis"/>
      <parameter name="description_it" value="Questo e' un file axm contenente video MPEG-4."/>
      <parameter name="description_en" value="This is an axm file containing MPEG-4 video."/>
    </content>
    <content name="Ax MP2-TS clip" id="102" delivery="FILE" type="M2T">
      <parameter name="file" value="file_ts.axm"/>
      <parameter name="encryption" value="axmedis"/>
      <parameter name="description_it" value="Questo e' un file axm contenente video MPEG-2 TS."/>
      <parameter name="description_en" value="This is an axm file containing MPEG-2 TS video."/>
    </content>
    <content name="AX DIVX clip" id="103" delivery="FILE" type="M4P">
      <parameter name="file" value="file_divx.axm"/>
      <parameter name="encryption" value="axmedis"/>
      <parameter name="description_it" value="Questo e' un file axm contenente video DIVX."/>
      <parameter name="description_en" value="This is an axm file containing DIVX video."/>
    </content>
    <content name="AX MP2-PS clip" id="104" delivery="FILE" type="M2P">
      <parameter name="file" value="file_ps.axm"/>
      <parameter name="encryption" value="axmedis"/>
      <parameter name="description_it" value="Questo e' un file axm contenente video MPEG-2 PS."/>
      <parameter name="description_en" value="This is an axm file containing MPEG-2 PS video."/>
    </content>
  </contents>
</sources>
</sources>

```

```
<!-- Cascaded xml content files -->
<others>
</others>
</tuner>
```

5 Demonstration and validation

5.1 Demonstration Video

Video released on July 2006 provides an overview of the satellite distribution as a stand alone application, without the use of the Programme and Publication tool.

The video presents the Content Distribution via Satellite data broadcast process, on both server/distributor side and client/user side, using a PC-based receiver.

The Publisher logs on the web interface and prepare the packages corresponding to the content to be delivered; then he/she defines the program, that means a catalogue of packages distributed, and its transmission parameters.

The video shows also in real time what happens when the program is on air. On the client side, the guide with the available content is live updated; the user can select the packages he/she's interested in from the list, and the download automatically starts at the scheduled time.

The received content is then available, thanks to the Action Manager, on the folder designed as cache area, and ready for fruition.

5.2 Research Guidelines Update

In the early phases of the project, the Consortium had planned to produce a simulator for the push/pull balance of data broadcasting delivering. Nowadays, the activity has been abandoned since it assumed less relevance for the market and for EUTELSAT.

The WP9.3 follows the guidelines of the WP4.8, and in particular the T4.8.1; this task has been reviewed, defining that the satellite distribution has to target two kinds of receivers: the traditional PC-based client and the STB. This demonstrator reflects the above updates on the R&D activities guidelines.

5.3 Plan for demonstration and validation phases

This section focuses on the next steps to have a definitive demonstrator of the satellite delivery channel in the AXMEDIS platform, covering the whole cycle of distribution and fruition of content.

The distribution process will not differ substantially between PCs and STB devices. In fact the only difference will be that all the contents for the STB devices will be flagged as “auto-listen”, so that the OSListener on the STB will save them without any end-user intervention.

Typically the STB clients will have some limitations in the capability to render the various kinds of contents, as shown in previous section [4.1]. This means that only a subset of all possible contents will be playable on the STBs.

We underline here that some slight differences are also foreseen concerning the use of the satellite channel and how the P&P interacts with the satellite platform. The prototype presented in this deliverable is based on punctual transmissions, that means that the AXMEDIS P&P is able to connect to and interface with the operational EUTELSAT Satellite Data Distribution Platform in order to perform all the steps necessary to translate each of its distribution programme in a Carousel Program. In the aim of the final demonstrator, we assume that the Carousel Program for Satellite will be always running, and the P&P will be able to add and remove content from the Carousel. This new behaviour demands indeed only a subset of the steps currently performed by the prototype.

5.3.1 STB phasing

Concerning the demonstration towards STB, the modular approach is one of the key factors that will drive the integration activities. New capabilities will be added progressively and smoothly integrated; by the end of the activity the following use cases will be validated:

- M29: Playout of AXMEDIS objects from STB local storage.
- M32: Play of AXMEDIS protected contents with local protection information. One way distribution of contents to a closed group of STBs via OPENSKY push service.
- M34: Play of AXMEDIS DRM-ed contents with access to PMS. Automatic generation of content index XML files.

The playout of AXMEDIS objects on the STB is available by means of the integrated AXMEDIS Object Manager (AXOM). According to MBI's middleware a new STB module, called AXPlayer, has been developed in order to satisfy the requirements of a DRM-capable media player. In particular the new AXPlayer integrates the AXOM and the hardware-dependent video libraries in order to handle the media data entirely in memory and without dynamically linked libraries.

5.3.2 Demonstrator Implementation

This section gives a description of the implementation of the demonstrator for the validation phase, as it is planned to be. Changes may apply in the following months to this provisional plan.

The Satellite Distribution Channel targets the two receiving devices described in this document, i.e. the PC-based clients and the STBs provided by MBI.

In both cases the Distribution is performed in multicast via Push Protocol, through the EUTELSAT's OPENSKY platform. EUTELSAT will provide the satellite distribution channels; a distribution channel includes satellite capacity as well as all the facilities on the Satellite Distribution platform to support the transmissions.

The Programme&Publication (P&P) of UNIVLEEDS is the tool that joins the AXMEDIS platform with the satellite distribution. The P&P Engines, connected with the AXEPTOOL P2P network, allows choosing content to be sent via satellite, and provides all the necessary steps to translate the transmission program on the EUTELSAT platform.

The selection on the P&P Editor of the satellite channel for the distribution of content brings the P&P Engine to automatically put that content in the sequence of the satellite carousel: the content will be transmitted at next repetition of the carousel, and then removed, by the P&P Engine, at the expiration date defined.

First distribution channel to be activated will be used by AXMEDIS partners for validation of the distribution chain and for demonstration to third parties: partners will be able to schedule content for distribution and receive it on their clients. From several locations it will be possible to put content on the satellite carousel for broadcasting, and to receive this content.

The content added in the carousel can be addressed to users by group management: groups will identify the partners and/or the type of client (i.e. STB/PC, ...); the content used in this demonstrator will thus have to satisfy the needs of the receivers to validate the distribution.

In the same way, besides this general distribution channel, another channel can be activated and dedicated to ILABS for distribution towards Kiosks. It will work, for what concerns the satellite distribution part, with the same architecture of P&P Engines to send + EUTELSAT Satellite Data Broadcast via Push for transport + PC with satellite reception and OPENSKY OSClient to receive. Since the satellite distribution channel would work as explained in this document, we do not give here more details on this other potential channel, and we remand to the documentation on the distribution via Kiosk on PDA (refer to WP9.6 of the AXMEDIS project).

6 Demonstrator Fact Sheets

6.1 Demonstrator Fact Sheet for Satellite Distribution to PCs

- **Main purpose**

The demonstrator aims at validating the distribution of content to PCs via Satellite Data Broadcast. The use of satellite for data distribution has the incontestable advantage of reaching any geographical area with minimal infrastructure, and the possibility to broadcast data to many users at the same time. Content produced with AXMEDIS content production tools is scheduled for satellite transmission via Programme&Publication, adding it in the carousel and broadcast to defined group of users. Distribution

via satellite to PCs can be done in Push, forcing the reception of content to the receivers, or proposing to the users the catalogue (guide) of contents and let them select what to download. Content downloaded is then regularly fruited with the AXMEDIS Player.

- **Review of the architecture integration with AXMEDIS**

- Content acquisition/ production/ processing/ protecting are not in the scope of this demonstrator;
- Users and tools registration is optional: content reception does not require any registration, but registration is needed for full fruition of the content;
- Content distribution operated via Programme&Publication Tool:
Programme&Publication connected with the AXEPTool P2P network and configured to access the satellite distribution channel: it retrieves content from P2P area and puts it in distribution on the satellite carousel.

Scenario for Satellite distribution:

- 1 P&P programme to be activated from the P&P Editor
- 2 The Engine parses and checks if the P&P Programme is already running
 - 2.1 Engine searches active programmes
 - 2.2 The matching programme is removed if found
- 3 if AXCP rule is associated with the AXMEDIS Object
 - 3.1 send request to AXCP with Object ID and AXCP Rule ID (Rule must be in the Rule Scheduler of the AXCP area)
 - 3.2 receive new object ID from the AXCP area
 - 3.3 update the programme with new object ID
- 4 calculate distribution time based on distribution schedule and distribution server profile and the programme is scheduled for distribution
- 5 at their scheduled distribution time
 - 5.1 request Objects from AXDB, local file or AXEPTool
 - 5.2 Receive objects
 - 5.3 Send to Distribution Servers

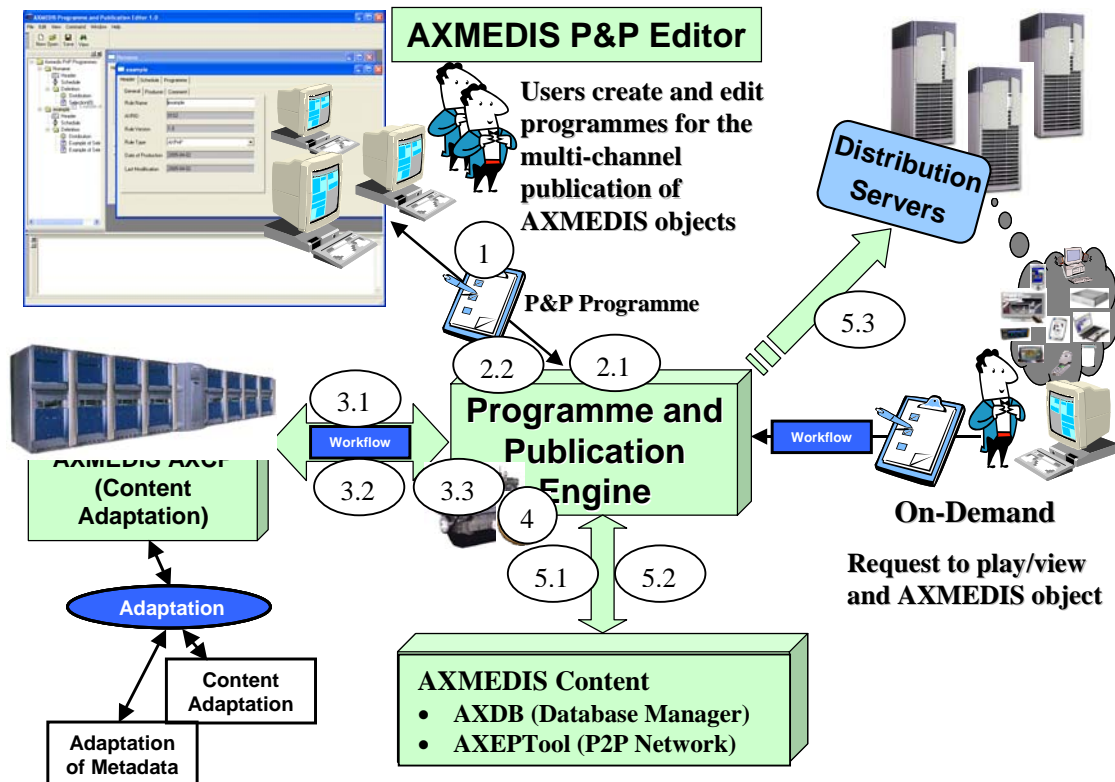


Figure 12: Overview of the P&P Programme Area for Distribution with Scenario for Satellite distribution

- **Description of the effective installation**

- EUTELSAT Satellite Data Broadcast platform:
Transport via Push Protocol - EUTELSAT is responsible of the distribution channels, from satellite capacity reservation to operations on its platform.
Satellite bandwidth provided on EUTELSAT's satellite ATLANTIC BIRD™ 1 positioned at 12.5°West, transponder C11 (short AB1-C11).
Satellite bandwidth of 512kbps 24/7 for 12 months, from M36 to M48, shared between demonstrators of distribution to PC-based clients and STBs. The same carousel will broadcast content to the two devices.
- Content Distribution Nodes (P&P Engines):
One or more Programme&Publication Engines connected with the AXEPTool P2P network and configured to access the satellite distribution channel. Content is taken from the P2P network and added in the EUTELSAT Carousel for PCs by querying and adding AXMEDIS objects to a P&P Programme using the P&P Editor tool and selecting the Eutelsat distribution channel. The P&P Programme is then activated into the P&P Engine to process and distribute the objects to the specified distribution channel. Content distribution is provided by partners of the project.
Communication between P&P Engines and satellite platform is via FTP and HTTPS protocols. Access to the satellite platform is restricted to authorized users. P&P Engines must be authorized by EUTELSAT.
- Distribution infrastructure for PC clients:
Satellite reception satellite ATLANTIC BIRD™ 1 positioned at 12.5°West, transponder C11 (short AB1-C11) i.e. satellite dish and DVB-S adapter for PC.

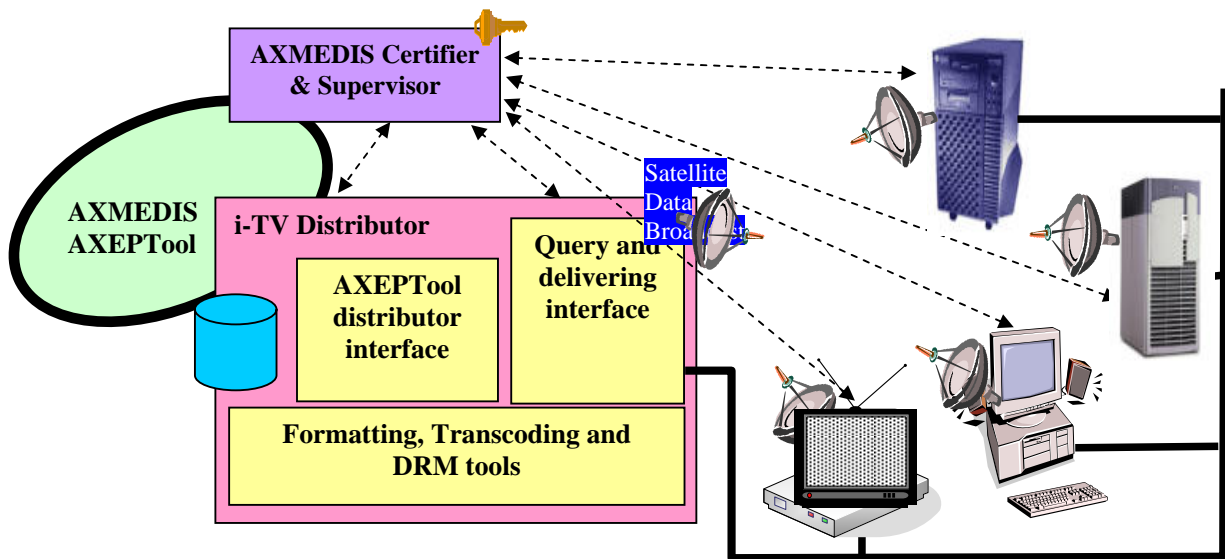
- Streaming/downloads:
No streaming; Content downloaded via Push Transport over DVB/IP.
- Players needed: AXMEDIS Player for PC.
- **AXMEDIS tools**
 - AXMEDIS P2P usage NO
 - AXCP usage NO
 - Workflow tools usage NO
 - Programme& Publication usage YES
Programme&Publication Engines configured to access the satellite distribution channels
 - PMS/AXCS usage NO
 - AXMEDIS database usage NO
- **Target Market**
 - First phase of the validation is limited to a restricted group of users, mainly partners of the project for validation and promotion.
 - The set up of a distribution channel targeting final users (B2C) will be considered later on. Satellite broadcast can reach number and number of users with a single transmission, so the targeted market is more or less a Mass market, but it is also suitable for large communities like branches of a corporate company.
- **Description of the business model**
 - EUTELSAT does not distribute the content directly, but provide the distribution channel to other producers/distributors. License kind and topology is thus left to these distributors.
- **Description of content**
 - Suggested at least 20-50 different content objects, to ensure a refresh of the carousel.
 - Content size limited to 1GByte for single file, in order to optimize the distribution reducing the risk of lost packets.
 - Format: any content / resource fitting the PC device are ok: partners participating in the demonstration can propose several models to validate the distribution.
 - Type of Content: to open the distribution to a mass market, audio and video content is suggested, to attract the potential users.
- **Final Users/Clients**
 - First phase: between 5 and 10 users.
 - They are partners of the project and promotional users for dissemination.
 - Registration of the user is optional: content reception does not require any registration, but registration is needed for full fruition of the content.
- **Partners involved and roles**
 - EUTELSAT:
responsible of the distribution channels, from satellite capacity reservation to operations on its platform.

- UNIVLEEDS:
responsible of the P&P tools (P&P Engine and P&P Editor) and its interface with the EUTELSAT platform (Eutelsat distribution plugin);
manage one of the P&P Engines to access the distribution channel.
- EUTELSAT, DSI, ANY:
equipped for satellite reception can validate the distribution channel.
- ANSC, AFI, BBC, ANY Content Partner:
Are asked to provide content for satellite distribution to PC, with respect of the constraints of the satellite channel (see above).

6.2 Demonstrator Fact Sheet for Satellite Distribution towards STB

- **Main purpose**

The demonstrator aims at validating the distribution of content to STBs via Satellite Data Broadcast. The use of satellite for data distribution has the incontestable advantage of reaching any geographical area with minimal infrastructure, and the possibility to broadcast data to many users at the same time. Content produced with AXMEDIS content production tools is scheduled for satellite transmission via Programme&Publication, adding it in the carousel and broadcast to defined group of users. Distribution via satellite to STBs is done in Push, sending the content to the hard-disk of the terminals, and then the user can select from the program guide the available content. Content downloaded is then regularly fruited with the AXMEDIS STB Player.



AXMEDIS Distribution of i-TV Content via satellite data broadcast

- **Review of the architecture integration with AXMEDIS**

- Content acquisition/production/processing/protecting are not in the scope of this demonstrator;
- Users and tools registration is optional: content reception does not require any registration, but registration is needed for full fruition of the content;
- Content distribution operated via Programme&Publication Tool:
Programme&Publication connected with the AXEPTool P2P network and configured to access the satellite distribution channel: it retrieves content from P2P area and puts it in distribution on the satellite carousel.
- **Description of the effective installation**
 - EUTELSAT Satellite Data Broadcast platform:
Transport via Push Protocol - EUTELSAT is responsible of the distribution channels, from satellite capacity reservation to operations on its platform.
Satellite bandwidth provided on EUTELSAT's satellite ATLANTIC BIRD™ 1 positioned at 12.5°West, transponder C11 (short AB1-C11).
Satellite bandwidth of 512kbps 24/7 for 12 months, from M36 to M48, shared between demonstrators of distribution to PC-based clients and STBs. The same carousel will broadcast content targeting the two different devices
 - Content Distribution (P&P Engines):
One or more Programme&Publication Engines are connected with the AXEPTool P2P network and/or AXDB and configured to access the satellite distribution channel. Content is queried on the P2P network using the query tool in the P&P Editor, and added to a P&P Programme for distribution. The EUTELSAT Carousel for STB distribution is selected in the P&P Editor and the programme is activated to the P&P Engine. Content for distribution is provided by partners of the project.
Communication between P&P Engine and satellite platform is via FTP and HTTPS protocols. Access to the satellite platform is restricted to authorized users. P&P nodes must be authorized by EUTELSAT.
 - Distribution infrastructure for STB:
Satellite reception satellite ATLANTIC BIRD™ 1 positioned at 12.5°West, transponder C11 (short AB1-C11) and STB provided by MBI.
 - Streaming/downloads:
No streaming; Content downloaded via Push Transport over DVB/IP.
 - Players needed: STB Player.
- **AXMEDIS tools**
 - AXMEDIS P2P usage NO
 - AXCP usage NO
 - Workflow tools usage NO
 - Programme& Publication usage YES
Programme&Publication Engines configured to access the satellite distribution channels and specified satellite distribution groups.
PMS/AXCS usage NO
 - AXMEDIS database usage NO
- **Target Market**

- First phase of the validation is limited to a restricted group of users, mainly partners of the project for validation and promotion.
- The set up of a distribution channel targeting business users (B2B) will be considered later on. The targeted market could be large communities like for example the branches of a corporate company for a business-tv type of service.
- **Description of the business model**
 - EUTELSAT does not distribute the content directly, but provide the distribution channel to other producers/distributors. License kind and topology is thus left to these distributors.
- **Description of content**
 - Suggested 10-20 different content objects, to ensure a refresh of the carousel.
 - Content size limited to 1GByte for single file, in order to optimize the distribution reducing the risk of lost packets.
 - Format: any MPEG2 and MPEG4 audio/video content conforming to Section 4.4, with a single embedded media resource.
- **Final Users/Clients**
 - First phase: between 3 and 8.
 - They are partners of the project and promotional users for dissemination.
 - Registration of the user is optional: content reception does not require any registration, but registration is needed for full fruition of the content.
- **Partners involved and roles**
 - EUTELSAT:
responsible of the distribution channels, from satellite capacity reservation to operations on its platform.
 - UNIVLEEDS:
responsible of the P&P tool and its interface with the EUTELSAT platform;
manage one of the P&P Engines to access the distribution channel.
 - MBI:
responsible of the STB for satellite reception.
 - XIM, ANSC, AFI, BBC, ANY Content Partner:
are asked to provide content for satellite distribution to STB, with respect of the constraints of the satellite channel (see above).

6.3 Demonstrator Fact Sheet for Satellite Distribution to Kiosks

- **Main purpose**

The demonstrator aims at validating the distribution of content to Kiosks via Satellite Data Broadcast. The use of satellite for data distribution has the incontestable advantage of reaching any geographical area with minimal infrastructure, and the possibility to broadcast data to many users at the same time.

At the Factory side, a set of contents is selected and added to a list called “catalogue”; this catalogue (together with related contents) can be scheduled for satellite transmission via Programme&Publication,

adding it in the carousel and broadcasting to defined group of Kiosks. The catalogue downloaded at the Kiosks will be able to be browsed by the users in order to access to related contents.

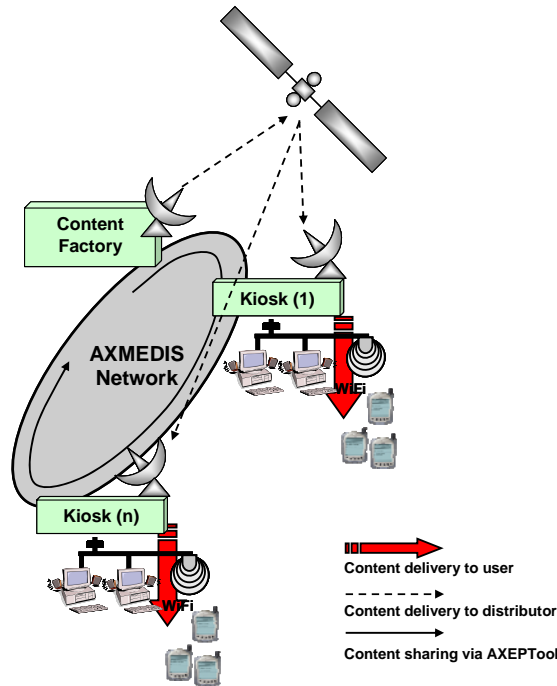


Figure 13: Sending and receiving on Kiosks through Satellite

- **Review of the architecture integration with AXMEDIS**
 - Content acquisition/ production/ processing/ protecting are not in the scope of this demonstrator;
 - Users and tools registration is optional: content sending and reception does not require any registration, but registration is needed for full fruition of the content;
 - Content distribution operated via Programme&Publication Tool: Programme&Publication connected with the AXEPTool P2P network and configured to access the satellite distribution channel: it retrieves content from P2P area and puts it in distribution on the satellite carousel.
- **Description of the effective installation**
 - EUTELSAT Satellite Data Broadcast platform:
Transport via Push Protocol - EUTELSAT is responsible of the distribution channels, from satellite capacity reservation to operations on its platform.
Satellite bandwidth provided on EUTELSAT's satellite ATLANTIC BIRD™ 1 positioned at 12.5°West, transponder C11 (short AB1-C11).
Satellite bandwidth of 512kbps 24/7 for 12 months, from M36 to M48, dedicated to the distribution to Kiosks.
 - Content Distribution Nodes (P&P Engines):
One or more Programme&Publication Engines connected with the AXEPTool P2P network and configured to access the satellite distribution channel. Content is taken from the P2P network and added in the EUTELSAT Carousel. Content distribution is provided by partners of the project.

Communication between P&P Engines and satellite platform is via FTP and HTTPS protocols. Access to the satellite platform is restricted to authorized users. P&P Engines must be authorized by EUTELSAT.

- Distribution infrastructure for PC clients:
Satellite reception satellite ATLANTIC BIRD™ 1 positioned at 12.5°West, transponder C11 (short AB1-C11) i.e. satellite dish and DVB-S adapter for PC.
- Streaming/downloads:
No streaming; Content downloaded via Push Transport over DVB/IP.
- Players needed: AXMEDIS Player for PC, PDA and Mobile Phone.
- **AXMEDIS tools**
 - AXMEDIS P2P usage NO
 - AXCP usage NO
 - Workflow tools usage NO
 - Programme& Publication usage YES
Programme&Publication Engines configured to access the satellite distribution channels
 - PMS/AXCS usage NO
 - AXMEDIS database usage NO
- **Target Market**
 - First phase of the validation is limited to a restricted group of users, mainly partners of the project for validation and promotion.
 - The set up of a distribution channel targeting final users (B2C) will be considered later on. Satellite broadcast can reach number and number of users with a single transmission, so the targeted market is more or less a Mass market, but it is also suitable for large communities like branches of a corporate company.
- **Description of the business model**
 - After the end of the project this scenario may be easily turned into a real case. In this case the business model is the “subscription” one as the owner of the kiosks factory will have to subscribe a service contract with EUTELSAT for bandwidth acquisition and usage.
- **Description of content**
 - Content size limited to 1GByte for single file, in order to optimize the distribution reducing the risk of lost packets.
 - Format: content will be in HTML, PDF, Text, SMIL and possibly Video and Flash embedded into AXMEDIS objects.
 - Type of Content: Art, Cultural Heritage and Tourism related.
- **Final Users/Clients**
 - First phase: 1 Kiosk at ANSC.
- **Partners involved and roles**

DE9.3.4 – Integrated prototype of content production and distribution in push

- EUTELSAT:
responsible of the distribution channels, from satellite capacity reservation to operations on its platform.
- UNIVLEEDS:
responsible of the P&P tool and its interface with the EUTELSAT platform;
manage one of the P&P Engines to access the distribution channel.
- ILABS/ANSC:
equipped for transmission/satellite reception can validate the distribution channel.
- ANSC, AFI, BBC, ANY Content Partner:
Are asked to provide content for satellite distribution to PC, with respect of the constraints of the satellite channel (see above).

7 Bibliography

DE9.3.1 Specification of content production and distribution in push and on-demand for i-TV
[axmedis-de9-3-1-spec-content-distribution-via-satellite-data-broadcast-v1-1.pdf](http://www.axmedis.org/documenti/view_documenti.php?doc_id=1292)
(http://www.axmedis.org/documenti/view_documenti.php?doc_id=1292)

DE9.3.2 Mock up of content production and distribution in push and on-demand for i-TV
[axmedis-de9-3-2-mockup-content-distribution-via-satellite-data-broadcast-v1-0.pdf](http://www.axmedis.org/documenti/view_documenti.php?doc_id=1919)
(http://www.axmedis.org/documenti/view_documenti.php?doc_id=1919)

DE9.3.3 First prototype of content production and distribution in push and on-demand for i-TV
[axmedis-de9-3-3-first-prot-content-distribution-via-satellite-data-broadcast-v1-0.pdf](http://www.axmedis.org/documenti/view_documenti.php?doc_id=2310)
(http://www.axmedis.org/documenti/view_documenti.php?doc_id=2310)

DE3.1.2.2.11 Specification of AXMEDIS Programme and Publication Tools
[axmedis-de3-1-2-2-11-spec-of-ax-progr-andpub-tool-v1-6.pdf](http://www.axmedis.org/documenti/view_documenti.php?doc_id=1885)
(http://www.axmedis.org/documenti/view_documenti.php?doc_id=1885)

API Push Server public functions [APS], Eutelsat SA

Video on [Satellite data broadcast and push](http://www.axmedis.org/documenti/view_documenti.php?doc_id=2175) (July 2006)
(http://www.axmedis.org/documenti/view_documenti.php?doc_id=2175)

Video on [Programme and Publication Area](http://www.axmedis.org/documenti/view_documenti.php?doc_id=2146) (July 2006)
(http://www.axmedis.org/documenti/view_documenti.php?doc_id=2146)