



STREP

FP6-2005-IST-61-045410

MOBISERVE

**New mobile services at big events using DVB-H
broadcast and wireless network**

Deliverable:

Interactive Service Application Platform Specification

D2.2

Due date of deliverable: August 31, 2007
Actual submission date: September 25, 2007

Start date of Project: 01 September 2006

Duration: 24 months

Responsible WP: France Telecom R&D Beijing

Revision:

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

1. DOCUMENT INFO

Author

Author	Company	E-mail
ZHOU Jie	FTB	jie.zhou@orange-ftgroup.com
Jianzhong LI	FTB	ljianzhong.ext@orange-ftgroup.com
DING Wenhui	BAMC	
YI Peng	CITVC	yipeng@citvc.com
Thierry Filoche	THF	Thierry.Filoche@thomson.net
Pierre Plevent	STZ	pierre.pleven@streamezzo.org

Comment [MS1]: Please update

Documents history

Document version #	Date	Change
V0.1	24/05/07	Starting template
	31/05/07	Skeleton generation and distribution
V1	14/09/07	First version with Thomson's input

Document data

Keywords	
Editor Address data	Name: ZHOU Jie Partner: FTB Address: France Telecom R&D Beijing 2 Kexueyuannanlu, Haidian District, Beijing, 100080 China E-mail: jie.zhou@orange-ftgroup.com
Delivery date	25 September 2007

Distribution list

Date	Issue	E-mailer

Table of Content

DOCUMENT INFO.....	2
AUTHOR	2
DOCUMENTS HISTORY	2
DOCUMENT DATA.....	2
DISTRIBUTION LIST.....	2
TABLE OF CONTENT	3
1. ABBREVIATIONS AND DEFINITIONS	5
1.1 ABBREVIATIONS	5
1.2 DEFINITIONS	5
2. INTRODUCTION.....	6
2.1 STATE OF THE ART AND ISAP OBJECTIVES	6
2.2 ISAP FUNCTIONS	6
2.3 SERVICES SUPPORTED BY ISAP	7
2.4 ISAP LOGICAL ARCHITECTURE	9
3. FUNCTIONAL ARCHITECTURE.....	10
3.1 ARCHITECTURAL MODEL	10
3.2 ISAP FUNCTIONAL ENTITIES.....	10
3.2.1 ISAP PRODUCTION CENTRE	10
3.2.2 ISAP KERNEL.....	11
3.2.3 ISAP TRANSPORT PLATFORM.....	12
3.2.4 ISAP PERIPHERAL ENTITIES	12
SERVICE MANAGEMENT.....	12
CONTENT PROVISIONING & STORAGE	12
REAL TIME SOURCE ENCODING	12
4. ISAP PRODUCTION CENTER	13
4.1 STUDIO.....	15
4.1.1 STUDIO FOR CONTENT/PROGRAM PRODUCER.....	15
4.1.2 STUDIO FOR CONTENT AGGREGATOR (CHANNEL PRODUCER)	15
4.1.3 STUDIO FOR SERVICE PRODUCER	15
4.2 CONSOLE	16
4.2.1 CONSOL FUNCTIONS FOR DIFFERENT PLAYERS.....	16
4.2.2 COMMAND TYPES AND FORMATS.....	16
4 ISAP SCRIPT.....	17

5	ISAP KERNEL	17
6	OTHER HEAD END ENTITIES AND INTERFACE WITH ISAP	21
6.1	MANAGEMENT PLATFORM.....	21
6.1.1	SMART VISION INTRODUCTION AND FUNCTIONAL ENTITIES	21
6.1.2	INTEGRATION SOLUTION AND INTERFACE WITH ISAP	22
6.2	RICH MEDIA	25
6.2.1	STREAMEZZO SERVICE NODE AND LASER CAST INTRODUCTION	25
6.2.2	INTEGRATION SOLUTION AND INTERFACE WITH ISAP	25

1. Abbreviations and Definitions

1.1 Abbreviations

DVB	Digital Video Broadcasting
DVB-H	DVB-Handheld
FLUTE	File Delivery over Unidirectional Transport
RTP	Real-time Protocol
UDP	User Datagram Protocol
XML	Extended Markup Language
ESG	Electronic Service Guide
ISAP	Interactive Service Application Platform
IPDC	IP Datacast

1.2 Definitions

Integral Interactive Mobile TV service: Service where data from all channels (including broadcasting channels and bidirectional channels) can be displayed or handled when they are needed and in the way as planned; and users (or terminals) could be required to have interactions/actions at the time and in the way it's planned.

Control point: a control point is a point in the program where user interactivities should be triggered. In some cases, user feedbacks should be sent back and handled to influence the progress of the program.

Program description script: it's the script used by the ISAP Production Center: the Studio module uses it to write program description file and the Console module uses it to formulate real time commands. The script is of xml format and includes program information needed to generate ESG.

2. Introduction

2.1 State of the art and ISAP objectives

IPDC over DVB-H is designed to transport different types of content such as audio, video, texts, pictures and binary files. But actually in the IPDC world, only separated protocols are defined: RTP for AV streaming and FLUTE for file downloading. Because different types of data are transmitted separately via different protocols -- no mechanism to associate AV streams, downloading services and other interactive services/applications – watching TV is almost all about mobile TV, while the users' requirements to have real **integral interactive mobile TV services** can not be fulfilled.

Moreover, the availability of attractive programs is the key for the success of DVB-H. Today's situation is: on one hand, DVB-H is capable of largely improving users experience by providing interactive multimedia services along with or as a part of AV programs; but on the other hand, the production process for integral interactive mobile TV services are complicated and needs profound IPDC knowledge which make it a mission impossible for traditional content providers.

Having carefully considered the situations described above, the objectives of ISAP is set as follows:

- Set up an end to end framework to enable and facilitate integral interactive service/application deployment. It is an end to end framework defining the data flow and control flow between different modules for interactive service application realization. The full range of services ranging from highly interactive ones to simple TV broadcast would be addressed. Moreover, the services would be designed in a way that even with low-end terminal which possesses only the DVB-H video viewing capability, the Audio/Video part of the highly interactive services could still be enjoyed by the end-users.
- Design and implement tools to facilitate the integral interactive program production process. Traditional program producers will be enabled to produce IPDC enabled integral interactive programs with all types of data and rich media elements

2.2 ISAP Functions \

ISAP functions are as following:

- Aggregates contents from multiple sources and their related metadata in order to provide a particular service application.
- Provides the head-end application logic.
- Generates service description metadata to be used in the Electronic Service Guide
- Provides notification mechanism (including emergency notification)
- Interaction end-point for terminals to interact with service application

2.3 Services supported by ISAP

First of all, we need to make a clear understanding what we mean by interactive service.

In the actual TV or the Internet domain, the interactivity is widely understood by the fact the viewer (or user) can choose what they want to see, to view, to go forth and back, to stop or make a rest when willing doing other things. In reality, this kind of interactivity consists of only the freedom given to the viewer (or the user) to manage the way to consume the content. Highly the degree of the freedom given, highly the interactivity is said.

Clearly, this is not what we meant for our interactive service. The interactivity that we are expecting is as follows:

- The service targeted by ISAP are broadcasted TV program centered, therefore:
 - The interactivity is related to the broadcasted video programs
 - The interactivity is triggered by the broadcasted video programs
- Three degrees of interactivity are possible
 - Direct interactivity
 - Distributive Interactivity
 - No interactivity (Service selection through ESG – Electronic Service Guide)

The Direct interactivity means that the interactions are happening between broadcasted TV program and the end users. They are initiated and controlled by TV program but upon the end users feedback, the TV program progression can be changed.

The Distributive Interactivity means that along with the TV program, the information describing the TV program is also broadcasted. The information may contain some hints (such as URL in a HTML page) from which the end users may start to visit other value added services.

The No Interactivity means only that, during its progression, the TV program does not ask feedback from end users (Direct Interactivity) nor give any description information which might help the end users to access direct to other value added services (Distributive Interactivity). In other words, it consists of the flat TV program.

The ***integral interactive mobile TV services*** can be any combination of the three degree of Interactivity.

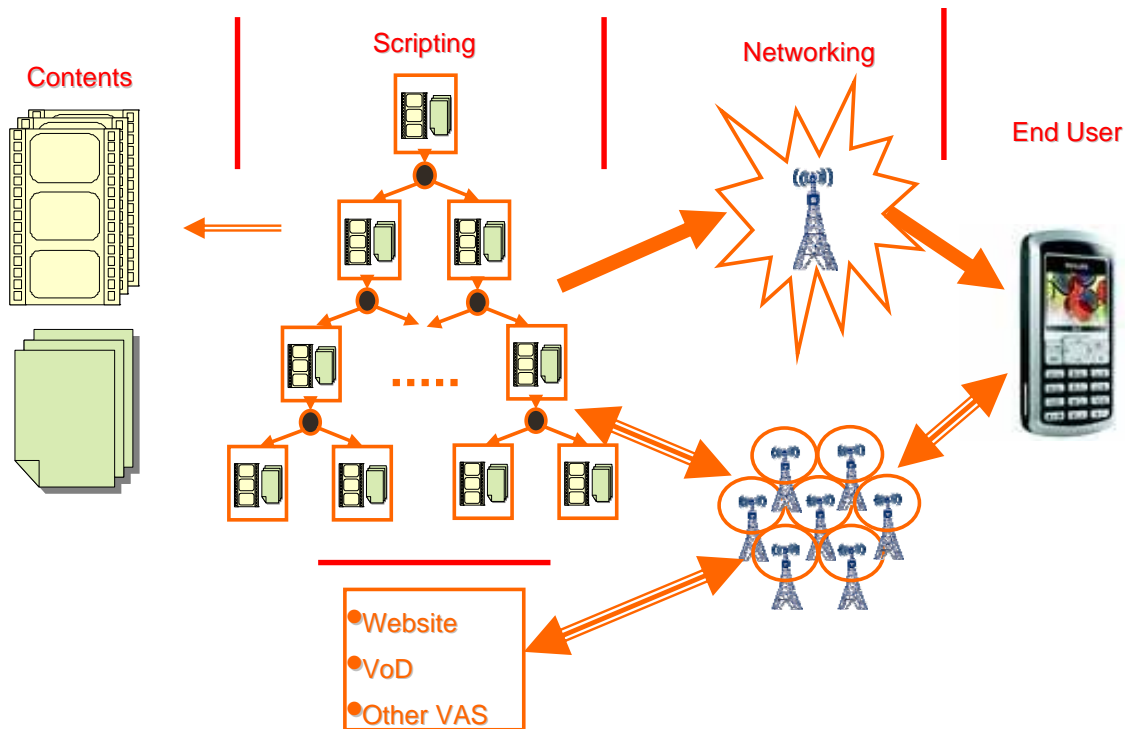


Figure 1 – The ISAP interactivities

ISAP is designed to not only support normal broadcasting services, but also integral interactive services. The following is an example of such services:

Example service scenario:

User enters TV Mobile service. A mosaic is presented to the user. The user chooses a film channel and begins watching a detective film.

During the film, user receives a message informing that a web site is available introducing the film and the film director. User clicks on the message and opens the web site to read information he's interested in.

When the first section of the film is about to finish, a message arrives telling that two pieces of the second section are available. In the first piece, police arrested the suspect number 1, and in the second piece, police arrested suspect number 2. Users can vote for the two options and the piece with most votes will be broadcasted in 2 minutes after the advertisement time.

The user votes for piece 2, but the piece favoured by most of the public is piece 1. A message arrived informing that the next piece to be broadcasted will be piece 1, but users can choose to watch piece 2 via VOD. The user clicked on the message and watched the film via VOD.

2.4 ISAP Logical Architecture

The following logical architecture describes the logical layers in the end to end platform. These layers will be implemented in different functional entities as depicted in the functional architecture.

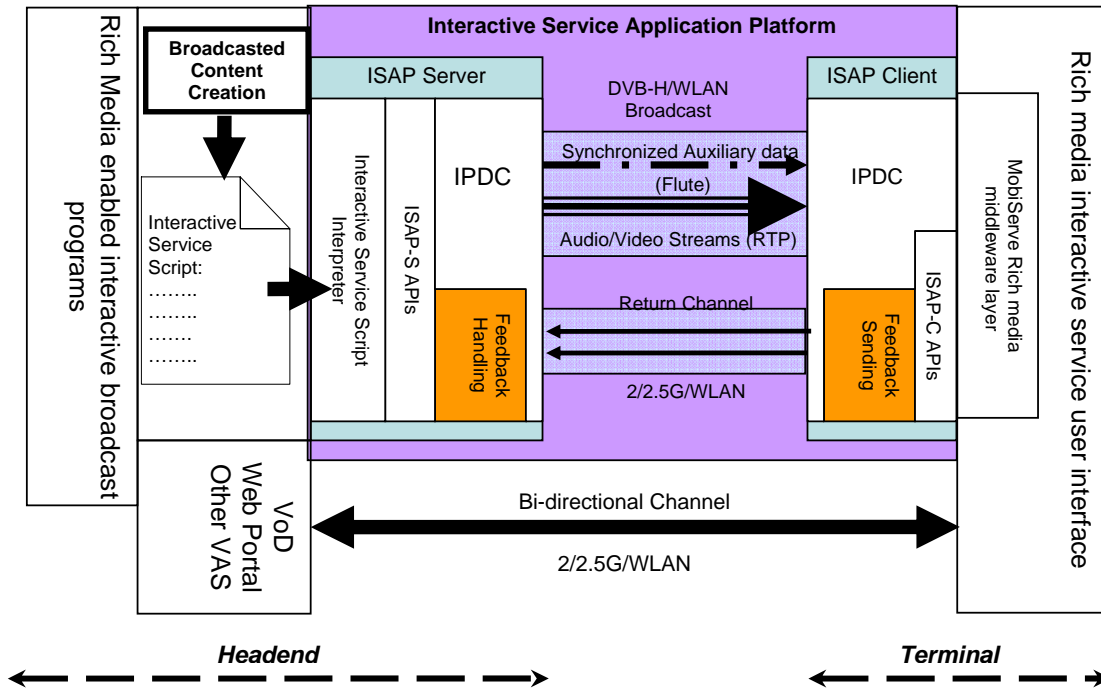


Figure 2 - ISAP Logical Architecture

3. Functional Architecture

3.1 Architectural Model

The head end functional architecture is made of mainly 4 parts (functional entities), namely:

- ISAP Production Center
- ISAP Kernel
- ISAP Transport platform
- ISAP peripheral entities

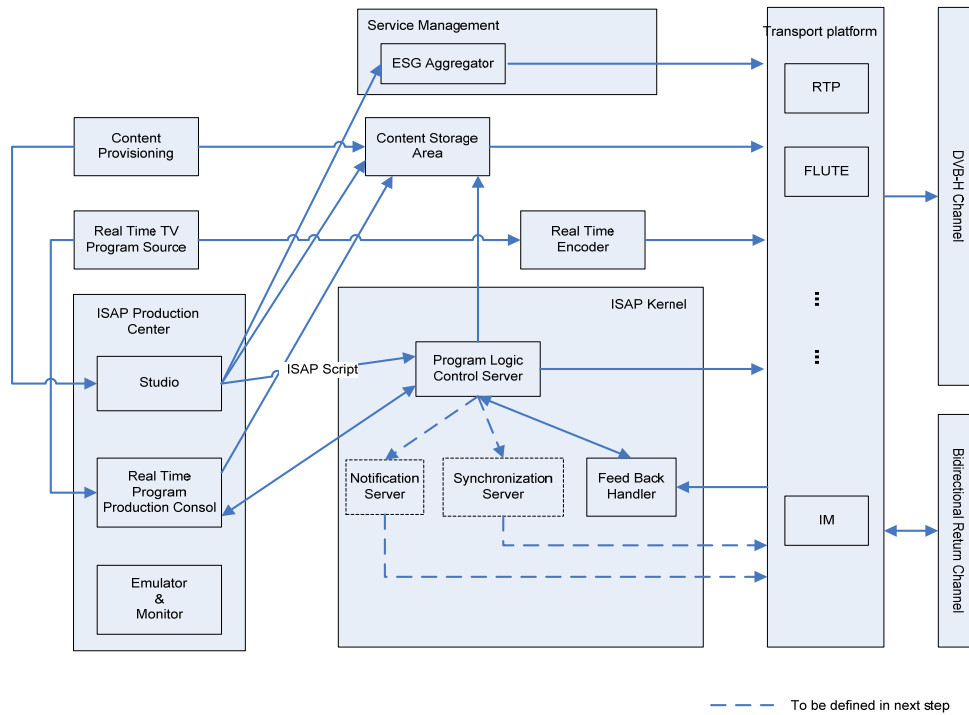


Figure3 -- ISAP Head end architecture

3.2 ISAP Functional Entities

3.2.1 ISAP Production Centre

ISAP production centre is an interactive service authoring/controlling environment provided to content/service providers and broadcasters. With the graphical interface provided by ISAP Production Centre, content/service providers and broadcasters can easily create and edit interactive programs, establish broadcast plans, and control the program when the program is being broadcasted.

ISAP Production Centre's functions are:

- Program logic defining including logics for interactivity
- Interface for program contents aggregation and edition
- Scheduling for contents sent via different channels
- Synchronization commanding
- Notification sending
- Service emulation
- Clip capturing

ISAP Production Centre includes different modules: Studio module for pre-made program authoring; Real Time Program Production Consol module for program control when program is on the air; Emulator & Monitor module for service emulation and monitoring.

3.2.2 ISAP Kernel

The ISAP Kernel is the core of ISAP. It interprets instructions from the ISAP Production Center (in the form of Program description script) and controls the over all logic of all TV programs and interactive services. It is made of 4 components.

- Program Logic Control Server

It is the control center of the ISAP Kernel. All files and commands coming from the ISAP Production Center are parsed by the program logic control server. It takes all the outputs of the ISAP Production Center and executes the instructions. It also collects user feedback results and decides the program direction according to program logic defined in the outputs of the ISAP Production Center.

- Notification Server

It is controlled by the Control Server and is in charge of sending notifications to end users.

Its functions include:

- 1) Generate notification messages from studio module and ESG demands
- 2) Decide message type and formulate the messages in a certain format
- 3) Decide distribution channel (dedicated channel & in band with AV program) and protocol

- Synchronization Server

It is controlled by the Control Server and is in charge of synchronizing TV related data to its TV program.

- Feedback Handler

It is controlled by the Control Server and is in charge of handling feedback from end users.

3.2.3 ISAP Transport platform

The ISAP Transport entity contains all transmission servers (using IP as network protocol), including unidirectional transmission servers -- RTP/UDP server, FLUTE server, etc., and bidirectional transmission servers -- Instant Messaging server, etc.

An ISAP API layer is laid on top of all transport protocols to wrap functionalities provided by transport servers. These APIs will be called by ISAP Kernel, to send out contents, embed synchronization signals, and send back feedbacks according to ISAP Kernel's demands.

3.2.4 ISAP peripheral entities

The ISAP peripheral entities are entities that are commercial products provided by partners or purchased in the market. We will use these entities as they are to complete our head end platform to offer the services.

- **Service Management**

ISAP Service Management entity is conforming to the definition of DVB-IPDC Service Management entity and is a subset of the latter. More specifically only "ESG Aggregation" functionality is included. A combination of ISAP Service Management and Smart Vision Mobility can make this subset more complete.

- **Content Provisioning & Storage**

Content provisioning provides mechanisms for third-party content providers to upload their audiovisual products to the ISAP head end. These products information (including storage location and metadata) can be further edited by the ISAP Production Center and then moved to Content Storage Area in order to be later broadcasted.

- **Real Time Source Encoding**

ISAP head end supports also reception of TV programs "on the air". These TV programs are encoded to the proper format in real time and then broadcasted.

4. ISAP Production Center

ISAP production center is an interactive service/program authoring/controlling environment provided to content/service providers and broadcasters. With the graphical interface provided by ISAP Production Center, content/service providers and broadcasters can easily create and edit interactive programs, establish broadcast plans, and control the program when the program is being broadcasted.

ISAP production center is the authoring/controlling environment offered to all players in the value chain including: content/program producer, content aggregator (or channel producer), service producer, and broadcast operator.

ISAP production center offer different functions for different user profiles. Functions for each user profile are listed below:

Functions offered for content/program producer:

- Interface for program contents aggregation, edition and association/synchronization
- Program logic defining including logics for interactivity
- Program uploading with metadata entering

Functions offered for content aggregator (or channel producer):

- Content/program censoring and technical check up
- Advertisement insertion
- Channel Scheduling
- Real time switch capability between different sources to deal with real time program requirement
- Real time data and notification sending capability
- Clip capturing

Functions offered for service provider:

- Data exporting for ESG generation
- User management
- Advertisement replacement and data advertisement insertion
- Advertisement statistics

ISAP Production Center includes different modules: Studio module for pre-made program authoring; Real Time Program Production Consol module for program control when program is on the air; Emulator & Monitor module for service emulation and monitoring.

- Studio (open to content/program producer, content aggregator (or channel producer) and service producer with different purview)

The Studio module is an authoring environment for non-real-time programs. It defines interactive service logic for contents. The interactive service logic indicates the interactions between TV program and end users, and is presented in the form

of Program description script, which is one input of “Program Logic Control Server” of the “ISAP Kernel”.

- Real Time Program Production Console (open to content aggregator (or channel producer) and service producer with different purview)

The Console module deals with real time TV programs that pass by the entity of “Real Time Source & Encoder”. It defines TV program related data synchronization and interactions between TV program and end users. It allows broadcaster to add prepared data and interaction to TV program while view it in real time. It also allows broadcaster to take consideration of end user feedback collected by the ISAP Kernel and to modify either related data or the live TV program itself.

- Emulator & Monitor (open to content/program producer, content aggregator (or channel producer) and service producer with different purview)

The emulator/monitor module shows to the broadcaster all the broadcasted TV programs as the end user can view, and provides means to control these TV programs.

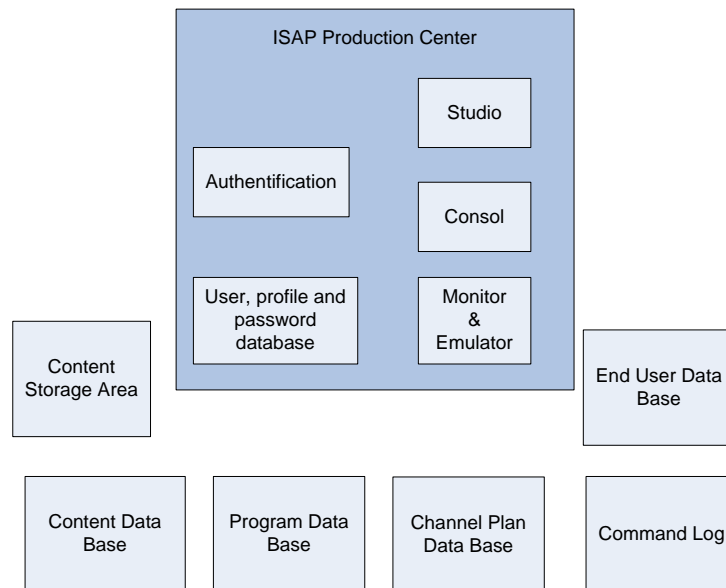


Figure4 -- ISAP Production Center

Content Storage Area: Storage area to store AV contents and data contents.

Content Data Base: Data base to store content description metadata which will include content storage information.

Program Data Base: Data base to store program description file

Command Log: Data base to store commands sent by consol

The program description file sent by the studio and the real time commands sent by the console take the same format to facilitate the interpretation by ISAP Kernel.

4.1 Studio

4.1.1 Studio for content/program producer

For content/program producer, Studio is a tool to aggregate contents for including AV content and data contents associated with the AV contents, set synchronization point between AV content and associated data contents, set program logic by setting control points, and upload program description files onto the program data base.

The working process for a content/program producer to create an integral interactive program is:

- i) Upload AV contents and data contents to Content Storage Area;
- ii) Enter content description information in content data base;
- iii) Define program by setting synchronization and control points
- iv) Save or/and send out program description file

The time base of the program descriptions files output by content/program producers always starts from 0.

The studio provide tool for setting up synchronization points. The tool consist a window to browse AV files. At the point where user wants to set up association and synchronization relation with certain data file, user can choose the data file and choose the relation type, actions should be performed by the data file at the synchronization point.

4.1.2 Studio for content aggregator (channel producer)

For content/program producer, Studio is a tool to aggregate programs made by program producers, perform program censoring and technical check up and set up schedule for the channel. In the process of setting up channel schedule, the time base of the program descriptions files produced by content/program producers will be mapped to the time base of the channel schedule.

The working process for a content aggregator / channel producer to create channel broadcast plan is:

- i) Set up the plan by choosing program description file and entering broadcast time for the program;
- ii) Create program description file and add it in the channel plan if the content aggregator / channel producer want to create some program or insert some content;

4.1.3 Studio for service producer

For service producer, Studio is a tool to aggregate channels into a service bundle, to export channel plans for ESG generation and to insert / replace advertisement if needed. At the same time, studio should provide tools for service provider functions for user management.

The entity involved in this operation is Studio module and Content Storage Area. Studio module has a graphical interface and needs no IT expertise for manipulation. By using studio module, content aggregators are able to:

- Aggregate contents from content storage area to formulate programs
- Set up schedules for different contents in one integral interactive program
- Define program logic for interactive programs (when to trigger user actions, how to handle user feedbacks, etc.)
- Establish broadcasting plan for all programs in the channel

The output of the studio module is program description file which is written by Program description script (of xml format). When the final program description file is generated, it will be sent to the program logic control server for program broadcasting on the server side and it will be sent to the ESG aggregator for ESG generation.

4.2 Console

The Console module is the environment provided to deal with real time operation requirements.

4.2.1 Consol functions for different players

Consol functions:

- 1) Real time broadcast plan defining or changing, which include:
 - Real time switch capability between different AV sources to deal with real time program requirement -- offered to program providers and channel providers
 - Real time file and notification sending capability -- offered to program providers, channel providers and service providers

For this function, the commands for broadcast plan defining or changing take the same formats as program description file which is written by Program description script (of xml format). More specifically, the same xml file generation module used by the studio is also used by the console for generating the commands.

- 2) Content / Program real time edition / creation tools
 - Clip capturing – offered to program providers
 - Feedback results browsing and real time statistical data browsing -- offered to program providers, channel providers and service providers

4.2.2 Command types and formats

The commands sent from Consol to ISAP kernel are classified into the following types:

- Program source (AV + file downloading) switching command
- Notification sending command
- File sending command with synchronization option

When the commands are generated, they will be sent to the program logic control server for program broadcasting on the server side, and at the same time, they will be sent to the ESG aggregator for ESG generation.

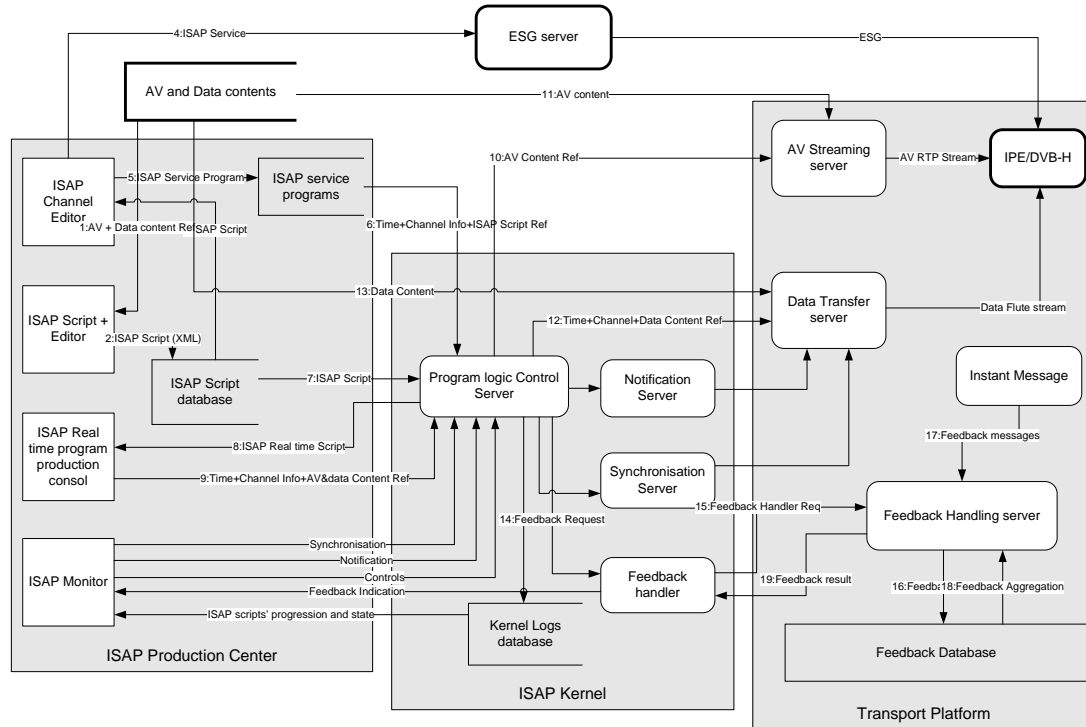


Figure5 -- ISAP Data Flow

4 ISAP Script

The communication between ISAP production centre and ISAP kernel is through program description files and real time commands. Both of them are written in ISAP script.

ISAP script will be translated by the Program Logic Control Server module in ISAP kernel to control the progress of programs.

At the same time, program description files and real time commands will be sent to the ESG aggregator for ESG generation.

For the reasons above, the ISAP script takes much the same syntax as ESG and adds its own elements for managing the end user interactivities.

5 ISAP Kernel

The ISAP Kernel is the core of ISAP. It interprets instructions from the ISAP Production Center (in the form of Program description script) and controls the over all logic of all TV programs and interactive services. It is made of 4 components:

- Program Logic Control Server

It is the control center of the ISAP Kernel. All files and commands coming from the ISAP Production Center are parsed by the program logic control server. It takes all the outputs of the ISAP Production Center and executes the instructions. It also collects user feedback results and decides the program direction according to program logic defined in the outputs of the ISAP Production Center.

- Notification Server

It is controlled by the Control Server and is in charge of sending notifications to end users.

Its functions include:

- 1) Generate notification messages from studio module and ESG demands
- 2) Decide message type and formulate the messages in a certain format
- 3) Decide distribution channel (dedicated channel & in band with AV program) and protocol

- Synchronization Server

It is controlled by the Control Server and is in charge of synchronizing TV related data to its TV program.

- Feedback Handler

It is controlled by the Control Server and is in charge of handling feedback from end users.

In the first version of ISAP implementation, the notification server and synchronization server will not be implemented, and they will be specified in the next version of specification.

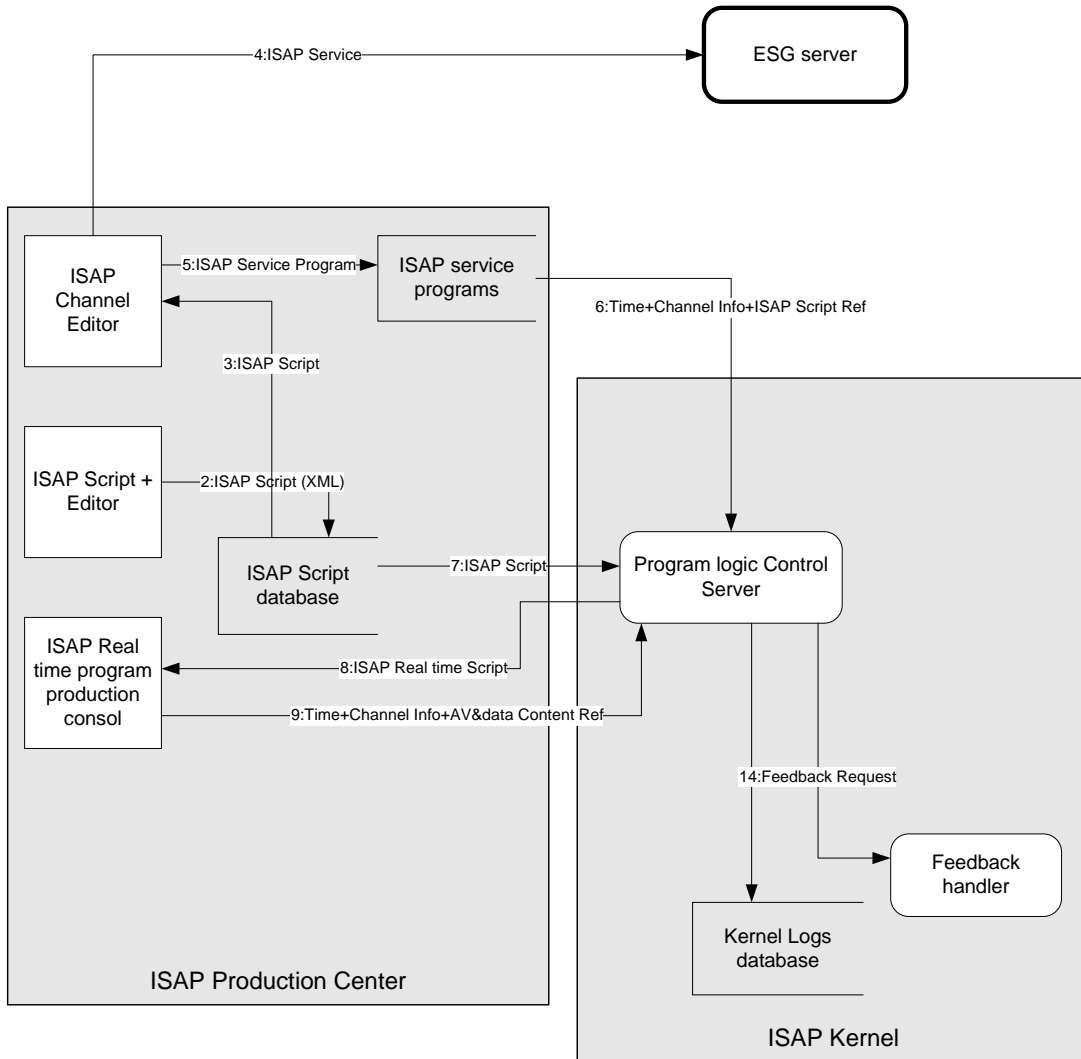


Figure 6 -- Dataflow of the ISAP Kernel

Data flow number	Label	Description	Data structure
2	ISAP Script (XML)	Program description files	XML files
3	ISAP Script	Program description files to be chosen by channels	XML files
4	ISAP Service	Program description files for ESG aggregator for ESG generation	XML files
5	ISAP Service Program	Program description files aggregated for a channel	XML files
6	Time + Channel Info + ISAP Script Ref	Program description files aggregated for a channel	XML files
7	ISAP Script	Program description files	XML files
8	ISAP Real Time Script	Program description files in real time	XML files
9	Time + Channel Info + AV & Content Data Ref	Program description files in real time	XML files
14	Feedback Request	Registration information for feedback handling	<ul style="list-style-type: none"> - Feedback ID: integer - Type: String - Start Time: date - End time: date - Validation: hh:mi:ss

6 Other head end entities and Interface with ISAP

6.1 management platform

6.1.1 Smart Vision introduction and functional entities

SmartVision Mobility (SVM) is Thomson service management system based on open standards, designed to manage multimedia services (audio, video, data) to be delivered over DVB-H networks on mobile devices (mobile phones, PDA, laptops,...).

SmartVision Mobility relies on Thomson DVB-H broadcast head-end system, SmartCast Mobility, to deliver live audiovisual content on the broadcast bearer.

SmartVision Mobility includes a powerful and flexible Electronic Service Guide (ESG) management and delivery. It integrates a program guide provisioning application that aggregates service description metadata used in the ESG and generates an overall ESG. It provides for a push delivery server to broadcast efficiently the ESG along with other contents (news, pictures, HTML pages,...).

SmartVision Mobility also interfaces with CAS/DRM systems for content and service protection in order to allow customers to consume and access services managed within the service platform.

SmartVision Mobility includes an administration web portal to manage services and bouquets, to declare products that define the purchase information related to services and bouquets, to manage the ESG generation and delivery, to administrate customers with their subscription, to manage servers that deal with services delivery, CAS/DRM systems and messaging servers.

SmartVision Mobility provides open interfaces with the mobile operator systems to deal with users' requests (services subscription, interactive services, logs collection...), the mobile operator billing, rating and provisioning systems, the mobile operator messaging servers.

SmartVision Mobility includes a user web/wap portal that can be shared across several DVB-H mobile phone terminals and accessible from the mobile operator network (UMTS, HSPDA, GPRS...).

In the framework of MOBISERVE, SmartVision will especially manage:

- Content provisioning (program guide description, audio/video descriptions (SDP))
- End-User management
- Electronic Service Guide accumulation and generation

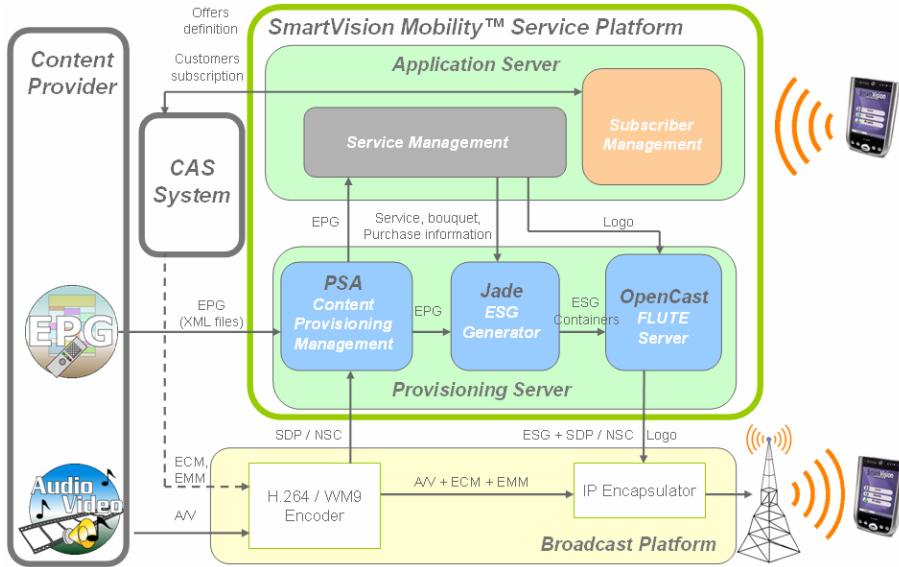


Figure 6 SmartVision architecture

6.1.2 Integration solution and Interface with ISAP

SmartVision platform interfaces at two levels with the interactive service application platform :

- With the ISAP Production center for ESG handling
- With the ISAP Kernel for ISAP commands broadcast

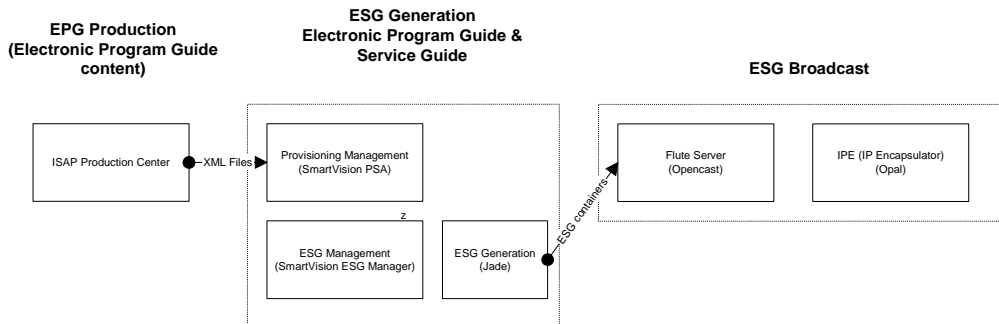


Figure 6 SmartVision interface with ISAP Production Center (EPG)



Figure 6 SmartVision interface with ISAP Kernel (ISAP commands)


Currently, the solution envisaged to integrate ISAP kernel is at Flute Server level: ISAP commands are provided via a file used as an input for the Thomson Flute server. This solution offers a good integration as it reduces number of modules;


Thomson flute server interface (Opencast) :


Opencast uses notion of “services” to distribute the content.


Each Opencast service can contain several files but only one file is sent at a time. The service defines the content's distribution features such as compression, encryption, reliability and the receivers to be addressed. It also defines the multicast IP address and port number used to transmit the files over the network.

The OpenCast system offers four kinds of services that differ in the way their files are sent.

 **Standard:** This is the basic kind of service and does not have any specific features. The files and folders to be sent are clearly defined and can be distributed in a loop (indefinitely or n times).

 **Carousel:** The files and sub-folders are sent in loop mode (it is exactly the same as a *Standard* service). The complementary feature regarding the standard service is that a “carousel” description file describing the list of content files is also delivered in a loop so that receivers can select the files they wish to receive.

 **HotFolder:** As soon as a file is dropped into a defined folder, it is automatically sent. Sub-folders and their content are also sent.

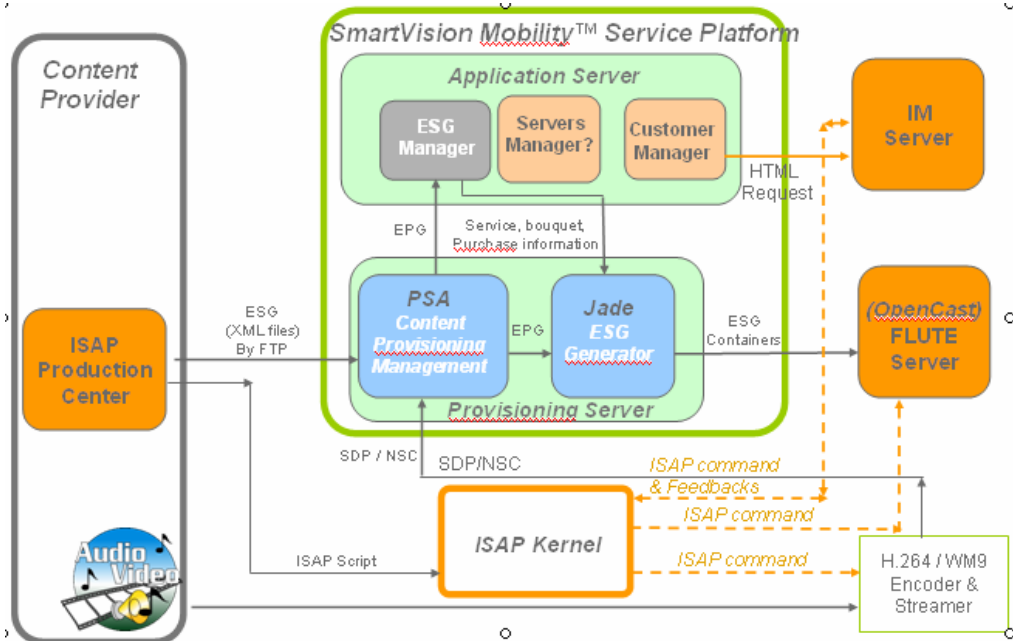
 **Mirroring:** Each folder and its contents are reflected, or mirrored on each receiver as they are at the Server end. The mirror file lists all the files and sub-folders making up the mirror folder. It is transmitted as soon as a change occurs in the mirror folder, in other words, whenever a file or a folder is modified, deleted or added.

NOTE. Carousel and Mirroring service kinds are not available in FLUTE/CBMS in order to respect full interoperability with other tiers FLUTE/CBMS receivers.

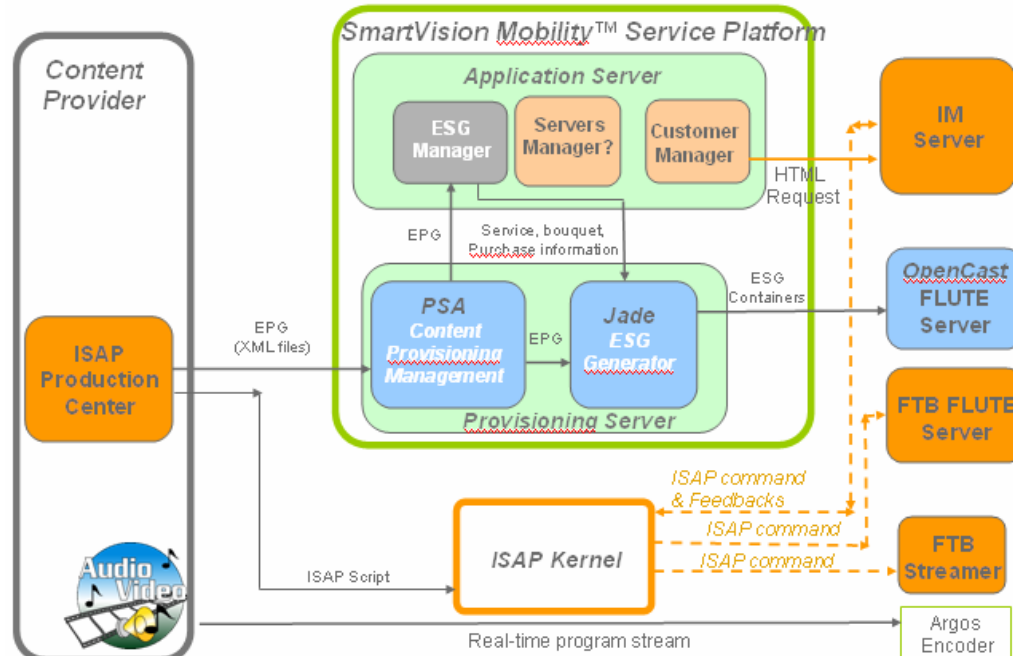
To interface with ISAP kernel, best adapted services are “Standard” and “HotFolder”.

In summary, two solutions are envisaged for the integration of S.V. and ISAP:

- Solution1: Opencast server be used for ESG and data files broadcasting (the standard mode or "hot folder mode" interface will be used in the communication between ISAP kernel and Opencast server); Thomson encoder & streamer be used for both real time and pre-made TV programs
- Solution2: Opencast server be used for ESG broadcasting, and data files broadcasting will be done by FTB flute server; Thomson encoder & streamer be used for real time TV programs and FTB streamer be used for pre-made TV programs



Integration solution 1 architecture



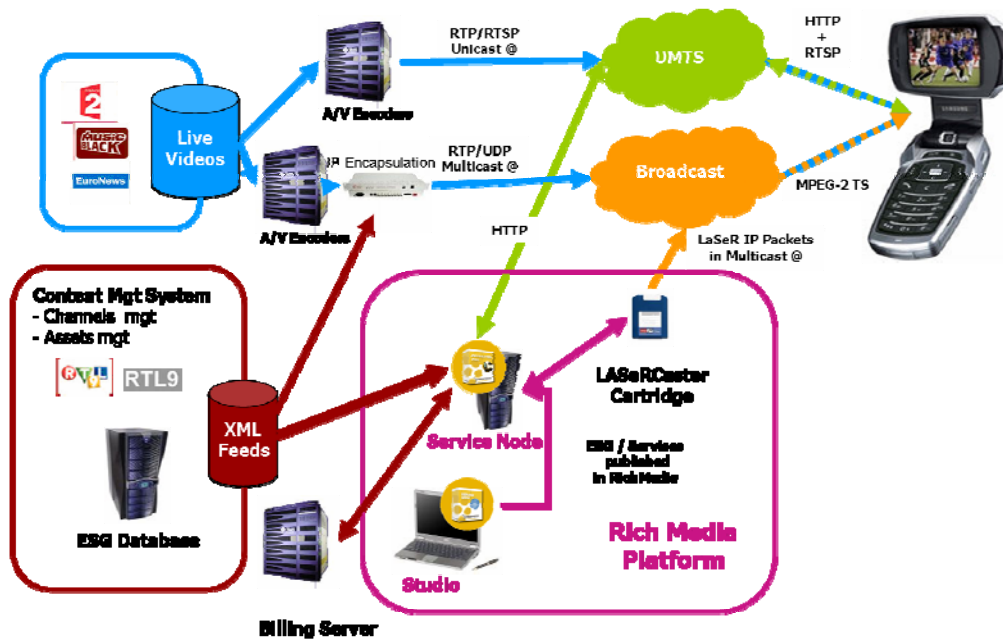
Integration solution 2 architecture

6.2 Rich media

6.2.1 Streamezzo Service Node and Laser Cast introduction

Diagram below shows complete flow of Rich media Production, the mechanisms at head end where Unicast Rich media is streamed by “Laser service node” and where Broadcast Rich Media is streamed inside the broadcast signal after multiplexing

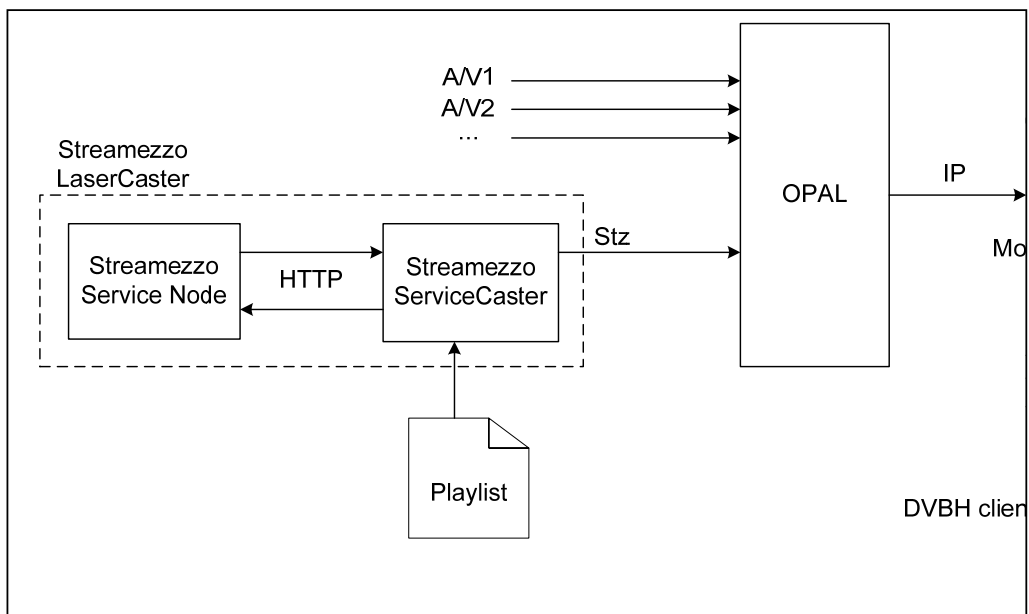
The terminal receives both type of content and the Rich Media engine, tightly coupled with audio and video codec, will render the final scene for the user.



6.2.2 Integration solution and Interface with ISAP

In order to implement the Rich Media solutions from the head end side is to supply a server that runs a LaserCaster and a Service Node and to integrate it in the platform in order to provide to the IPE (Thomson OPAL) a RichMedia input. This input will be multiplexed with the DVB data in order to display on the client wanted pieces of information.

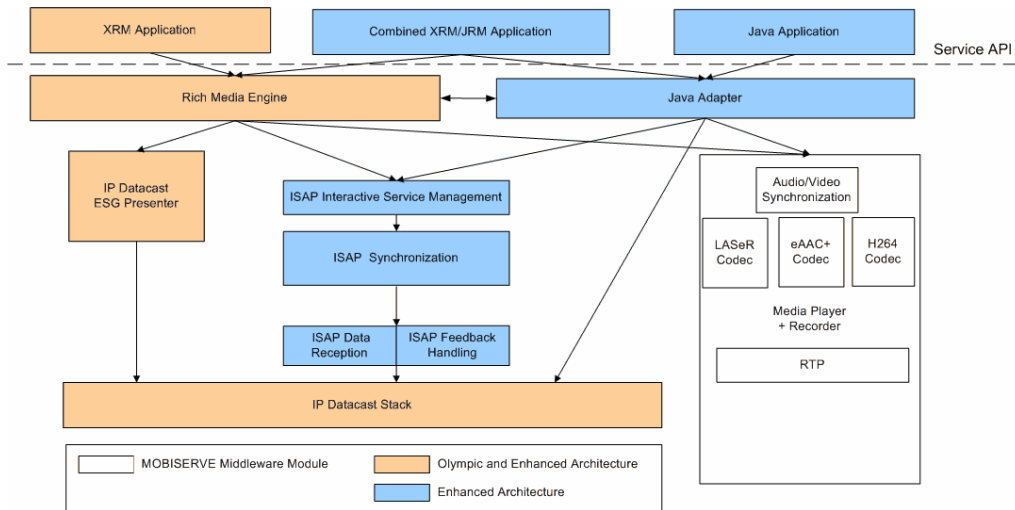
Here is a simple schema of the implemented architecture.



Interface with ISAP Production Center is performed through Playlists inputs

On the terminal side the integration architecture is as below:

In this integration solution, the ISAP and Rich Media flow independently



Interfaces between Rich media engine and ISAP Interactive Service Management will be managed with the same syntax like the IP Datacast ESG Presenter.