Deliverable 6.3.1

Project Title	Next-Generation Hybrid Broadcast Broadband
Project Acronym	HBB-NEXT
Call Identifier	FP7-ICT-2011-7
Starting Date	01.10.2011
End Date	31.03.2014
Contract no.	287848
Deliverable no.	6.3.1
Deliverable Name	Report on test applications for enablers of WP3/WP4/WP5
Work package	6
Nature	Report
Dissemination	Public
Author	Michael Probst (IRT)
Contributors	Björn Stockleben, Jennifer Müller, Bettina Heidkamp-Tchegloff (RBB), Janina Renz (IRT), Felix Gomez Marmol, Gines Dolera (NEC), Ivan Kotuliak, Tomas Kovacik (STUBA), Sebastian Schumann (ST), Christopher Köhnen, Marcel Raner (THM), Ray van Brandenburg (TNO), Stefan Heller (TARA)
Due Date	30.09.12
Actual Delivery Date	24.10.12

Table of Contents

Executive Summary

This document has two major parts. Chapters 2 to 4 contain a description of the test applications demonstrating the capabilities of the first prototypes of the HBB-NEXT enablers developed in Work Packages (WP) 3, 4 and 5. Chapter 5 then documents the roadmap until Milestone 3 of the project, a development plan for enablers and applications for the coming project year. This roadmap is an integral part of the project Milestone 6 which also features the initial version of HBB-NEXT software components.

The applications described in this document showcasing the initial enabler components are early testing applications and partially also end-user applications. Chapters 2 to 4 describing these applications along work packages and enablers are structured alike: Each of these sections starts off with a functional description to then detail the hardware and software set-up of the application. Chapter 2 gives an overview of the test application developed in WP 3 for the enablers Identity Management and Profile Management (ST) and Trust and Reputation Management. Chapter 3 introduces the development towards synchronisation and cloud offloading coming from WP4. Chapter 4 summarizes the input from WP5, describes the input on group recommendation and a detailed test application on the field of speaker identification as an implementation for the Multi-modal interface enabler.

An overview over the future plans concerning development and also validation and the interdependence and fine tuning between both is given within the roadmap chapter 5, part of Milestone 6. Here, each work package (2-6) describes their development plans and strategy for year 2 of the project. The conclusion in chapter 6 then summarises everything in the context of the HBB-NEXT project framework. Basically, five HBB-NEXT applications were defined which together cover all HBB-NEXT enablers and which, after the final integration at the end of 2013, will feature in the HBB-NEXT prototype platform. The inter-work package approach which is followed here in five small application teams will allow for an early integration process. Timewise, click dummy applications will be ready for user validation until February. The actual application development phase with the intermediate integration process will then start in March 2013 and last until May with a first integration meeting during the next Consortium Meeting in early March 2013 and a later one in July 2013. The integration will take place among the different enablers (intermediate versios of Milestone 7) while some enablers will also be integrated to run on the TARA set top box. The intermediate integration process will culminate in the first version of integrated software modules with the tested applications which will be available at Milestone 3, at the end of project year two.

1. Introduction

The aim of the HBB-NEXT project is to develop an open framework to enable a new kind of hybrid broadcast and broadband TV services and applications. This framework is a set of enablers which are designed and developed in the Work Packages WP3, WP4 and WP5. One of the tasks of WP6 (Task 6.3) is to showcase the capabilities of the enablers by developing applications for them, which are derived from the scenarios defined in WP2.

HBB-NEXT work is structured into three Development Cycles. Month 12 – at the time of writing this document – marked the end of Development Cycle 2 which resulted in WP3, 4, WP4 and WP5 delivering an initial version of the enabler components. This, together with the roadmap contained in chapter 5 of this document, fulfils Milestone 6 of the project. The applications described in this document showcasing the initial enabler components are early testing applications and partially also end-user applications.

The concepts, algorithms and protocols presented in this deliverable mark an essential step in the development process. Being the first set of specific testing applications they show the first set of base functionalities that will be developed within the enabler.

The application development itself was done within each respective work package (3, 4 and 5) and the results in the context of Development Cycle 2 are summarised within this document, describing the functionality as well as the system requirements concerning hard- and software of the different applications. These early applications focus on testing new functionalities for a specific enabler or part of it. The development of smaller applications has some great advantages. Firstly, they could be developed and tested quickly by small teams. Secondly, a small application is less error-prone since the error-source can be spotted more easily. In the end the partners could work more independently on a specific non-integrated piece of software covering their specific field of knowledge. These applications can now be used as a base for more integrated greater applications covering more than one enabler.

The integration will happen among the enablers as well as on a new HBB-NEXT set-top-box. For this, a first integration workshop was held at TARA Systems in June 2012. The specification for the set-top-box is appended to this document.

The next steps concerning the development and integration of the applications as well as the general further project plans for the coming year are described in chapter 5, the second major part of this document. The roadmap outlined there from the perspective of each work package is an integral part of Milestone 6 of the project. The idea is to present a plan for the development stages

for Work Package 3, 4, 5 and 6 until Milestone 3, i.e. for the second project year. This will help to plan and to organise the communication for future development. However, Work Package 2 was included here too to show the time plan and interdependencies of Development Cycle 3 and User Validation Phase 2.

2. WP3 Test Applications

Within this chapter the stand-alone test applications, which utilize the enablers 'identity management' and 'trust and reputation management' are defined.

2.1. Identity Management and Profile Management

The identity management component has been implemented at a proof of concept stage for the API. The current prototype supported WP3 and WP5 in the creation of the API design in D3.2 [5] and D5.2 [7]. It is not clear yet whether it will be extended or whether a new prototype will be built.

2.1.1. Functional Description

The identity management has been implemented using Ruby on Rails [9]. The scaffolding helped to deploy both the API and a simple front-end. The front-end is shown in Figure 1, Figure 2 and Figure 3.

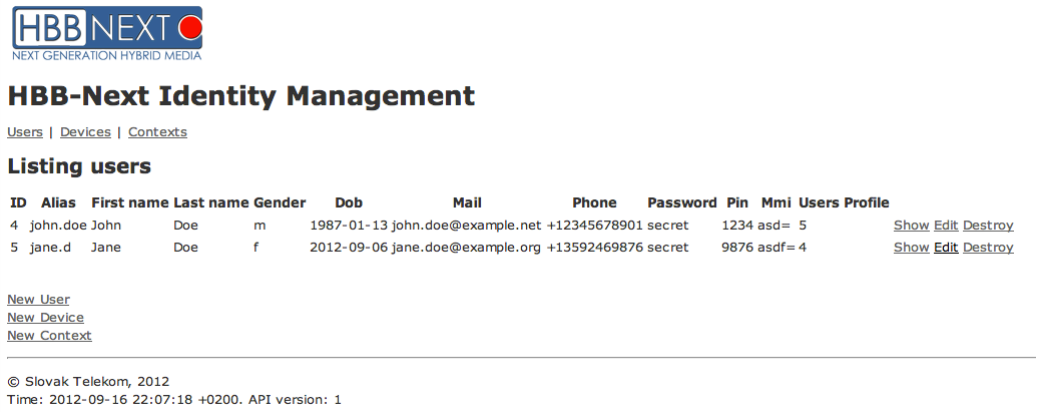

Figure 1: IdM users front-end (eq. GET /users)

Figure 2: IdM devices front-end (eq. GET /devices)

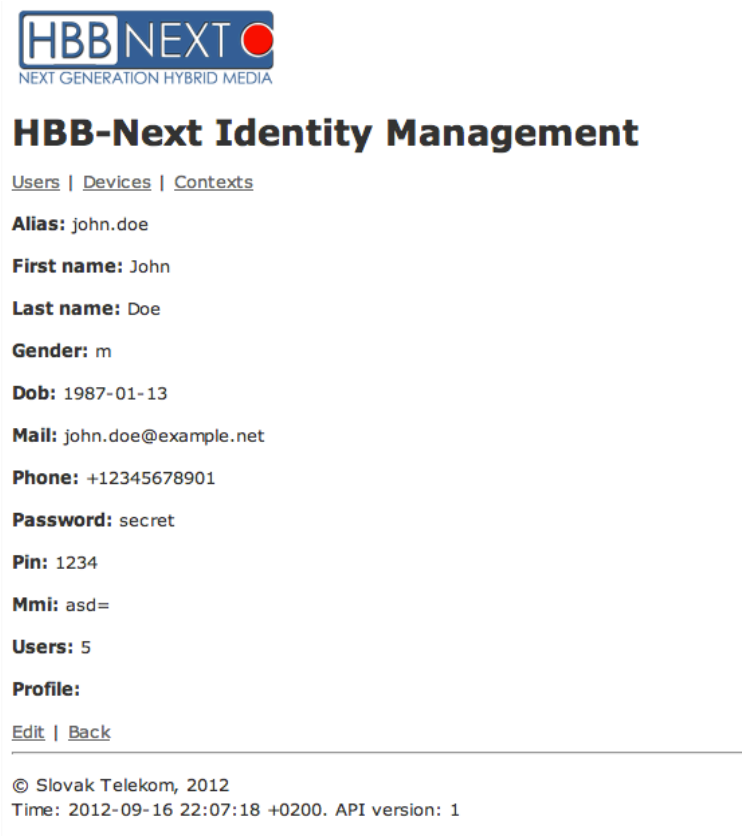

Figure 3: IdM user front-end for a particular user (eq. GET /users/4)

The API request to a request for the same user is produced in several output formats.

The following is a request to the same user represented in JSON:


```
{"alias":"john.doe","created_at":"2012-09-06T15:16:02Z","dob":"1987-01-
13","first_name":"John","gender":"m","id":4,"last_name":"Doe","mail":"john.
doe@example.net","mmi":"asd=","password":"secret","phone":"+12345678901","p
in":1234,"profile":null,"updated_at":"2012-09-06T15:27:36Z","users_id":5}
```

(GET /users/4.json)

The following is a request to the same user represented in XML:


```
<user>
<alias>john.doe</alias>
<created-at type="datetime">2012-09-06T15:16:02Z</created-at>
<dob type="date">1987-01-13</dob>
<first-name>John</first-name>
<gender>m</gender>
<id type="integer">4</id>
<last-name>Doe</last-name>
<mail>john.doe@example.net</mail>
<mmi>asd=</mmi>
<password>secret</password>
<phone>+12345678901</phone>
<pin type="integer">1234</pin>
<profile type="integer" nil="true"/>
<updated-at type="datetime">2012-09-06T15:27:36Z</updated-at>
<users-id type="integer">5</users-id>
</user>
```

(GET /users/4.xml)

The software is processing the data in the data module only once. It is only the output format that differs according to the requested type.

2.1.2. Hardware Setup

The enabler has been implemented on normal consumer PC hardware. As the implementation shall only proof the design of the API, there are no particular requirements for the system.

The implementation used the scripting language Ruby on Rails which has no further implications on the hardware (e.g. architecture).

The test application has been implemented on the following equipment:

- Apple MacBook Pro
- 2,3 GHz Intel Core i5
- 8GB RAM

- Mac OS X v10.8

2.1.3. Software Setup

The prototype has been implemented using Ruby on Rails. The easiest approach seemed to be building up on a scaffold.

The implementation modified mainly controllers, models, database and views.

Further implementation details will be provided in future versions of this document.

2.2. Trust and Reputation Management

2.2.1. Functional Description

The test application for checking the initial functionality of the trust and reputation management enabler (designed and implemented as part of the WP3 work) consists of a web browser based app-store showing the computed reputation assigned to each HBB-NEXT application before installing and consuming it.

Such a test application shows how the trust and reputation module aims to assist end-users selecting the most trustworthy and reliable HBB-NEXT application on the market. To this end, every HBB-NEXT application offered to the end-users through the app-store has an associated reputation score. Yet, such computed reputation score for each HBB-NEXT application is not just the average of the received feedbacks or recommendations from previous users, but rather the result of a sophisticated mechanism.

Every end-user receives a customized reputation score for each HBB-NEXT application, depending on their personal preferences. That is to say, the opinions and feedbacks coming from users with similar preferences to user *A* have a higher weight when computing the customized reputation values for the HBB-NEXT applications to be shown to user A.

The test application also allows end-users to provide their feedback and recommendation about a previously downloaded and consumed HBB-NEXT application. The test application takes into account the capabilities of the device of the end-user. Therefore, the way of providing opinions depends on the device in use. Those opinions are used, in turn, to compute subsequent reputation scores.

Finally, Figure 4 shows the flow diagram of the test application developed in order to show the initial functionality of the trust and reputation enabler.

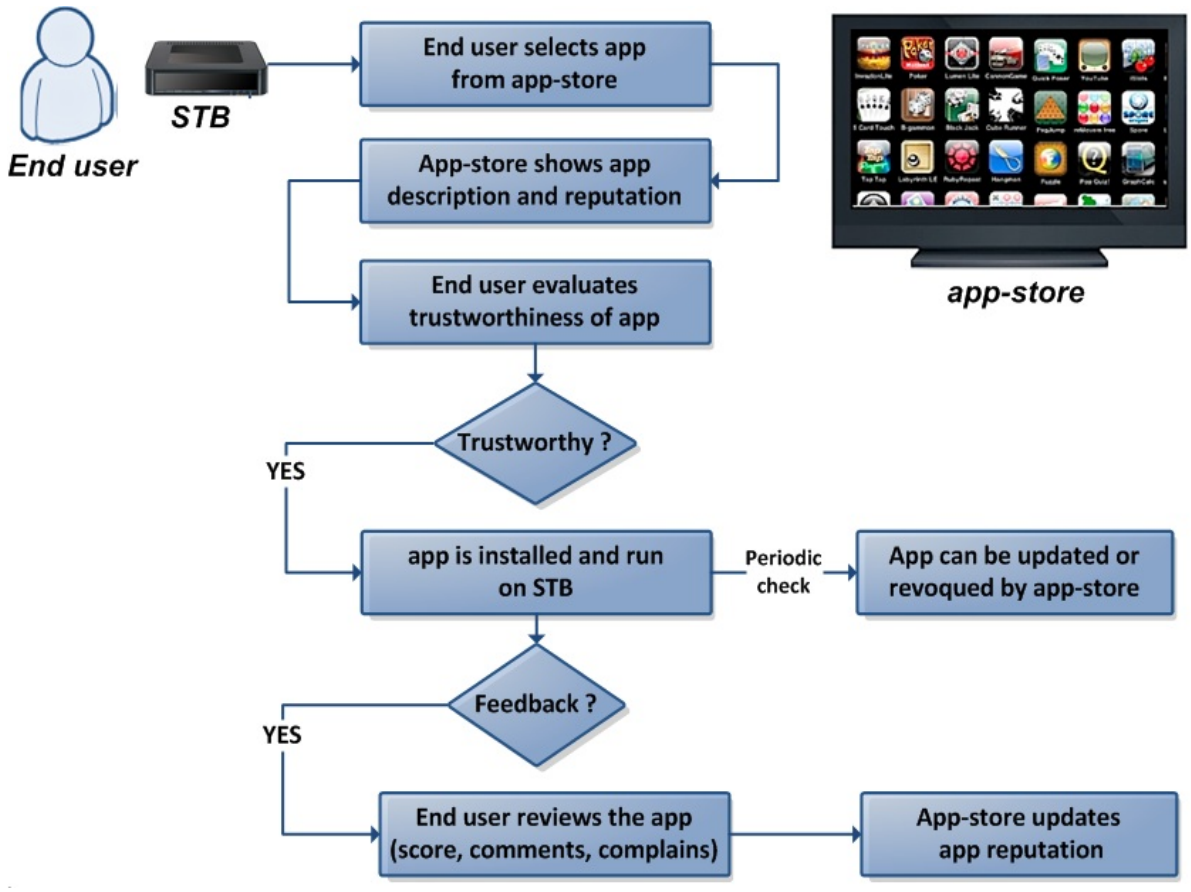

Figure 4: Interaction between the end-user and the app-store

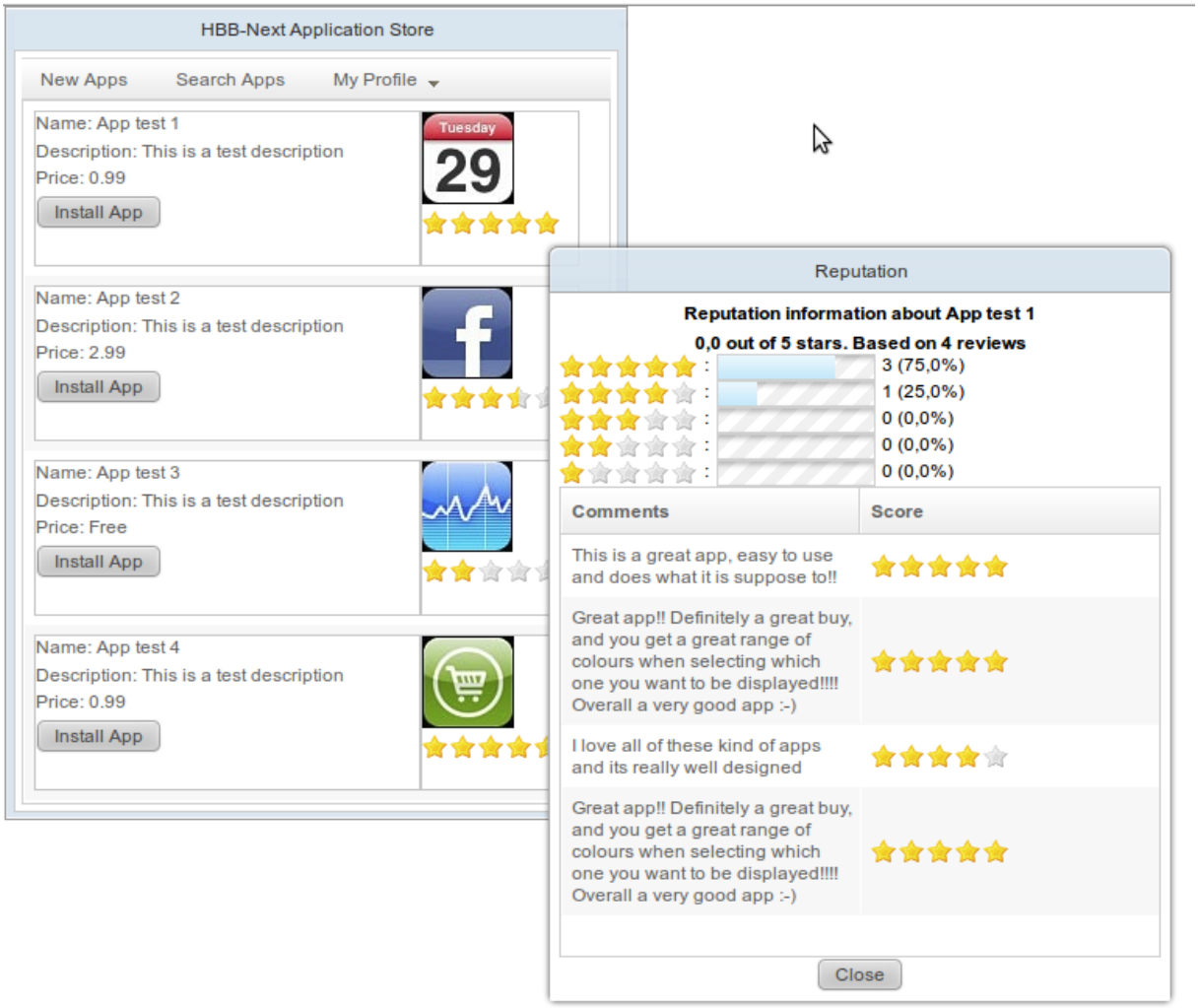

Figure 5: Screenshot of the test application for the trust and reputation enabler

In short, the script for the demo will consist of the following steps:

1. An end-user accesses the app-store from a PC, laptop, smartphone, tablet or PDA.

2. To be able to download HBB-NEXT applications the end-user needs to be logged in the system. Since he/she does not have an account for this platform yet, he/she decides either to create a new one, or use other alternative mechanisms like providing his/her OpenID account [2] [4]. In any case, the preferences of the user regarding HBB-NEXT applications should be reflected.

3. Either using his/her previous OpenID account or the recently created one, the end-user logs into the system.

4. The end-user browses through the app-store checking the customized reputation scores for the HBB-NEXT applications and maybe some comments left by previous users.

5. The end-user chooses one trustworthy HBB-NEXT application and decides to download and install/consume it.

6. Once the end-user has downloaded the HBB-NEXT application, he/she provides a feedback regarding his/her satisfaction with such application.

7. The end-user checks again the new reputation score of the downloaded HBB-NEXT application and observes how his/her feedback has influenced/altered the previous reputation score of such application.

8. The end-user logs out from the platform.

9. A different end-user logs into the system. This end-user has different preferences than the previous end-user.

10. This new end-user checks the customized (different) reputation scores for the HBB-NEXT applications, specifically for the HBB-NEXT application the previous user consumed and rated.

2.2.2. Hardware Setup

Figure 6: Hardware setup of the test application for the trust and reputation management enabler

Conceptually, Figure 6 shows the hardware deployment required to show the functionality of the trust and reputation enabler. Nevertheless, only the following hardware elements are needed in order to show this first version of the test application for the trust and reputation management enabler:

- One laptop acting as app-store server
- One tablet to allow the end-user to access the app-store
- Wireless access point to communicate both the laptop and the tablet

As of the specific requirements for each one of these elements, we have:

- Laptop
  - Wi-Fi supporting IEEE802.11g
  - At least 384 MB of RAM (512 MB is recommended)
- Tablet
  - Wi-Fi supporting IEEE802.11g
- Wireless access point
  - Support of IEEE802.11g
  - Static IP for client support

2.2.3. Software Setup

On the devices the following software is installed:

- Laptop
  - Java Runtime Environment (JRE) version 6 or later [3].
  - Any application container supporting Servlet 2.3 or later (e.g. Tomcat 6 [1]).
- Tablet PC
  - Web browser Firefox 3+, Internet Explorer 6+, Chrome10+, Safari3+, Opera9+ or compatible

2.2.4. Open issues for the final demo

Finally, there are still some open questions that will need to be faced in future versions of the test application, and fully addressed for the final version:

- How to actually link and communicate the tablet with the TV
- How and where to develop the real app-store
- How to develop HBB-NEXT applications for the HBB-NEXT app-store in a “standardised“ way (i.e. to be valid in any HBB-NEXT deployment)
- How to integrate the reputation module within these applications in a “standardised“ way (i.e. to be valid in any HBB-NEXT deployment)

- How to compute reputation depending on the device in use and other system conditions
- How to preserve the privacy of end-users when providing their feedback and avoiding this way app stores to trace their suggestions

3. WP4 Test Applications

This chapter summarises the two test applications by the WP4 partners which utilize the enablers A/V Content Synchronisation and Cloud Service Offloading.

TNO defines a technical approach to show the synchronisation of IP and broadcast. While on the other hand RBB and IRT develop a user centred test application using IP and broadcast synchronisation mechanism. The Cloud Service Offloading Application shows how a video processing application can be implemented in the cloud.

3.1. Synchronization of IP video with Broadcast Service

The aim for this test application is to present frame-accurate synchronization of multiple audio-visual media streams delivered over heterogeneous networks and transport protocols. The application, which is part of TNO's multi-platform synchronization test-bed, will be demonstrated at the 2012 MediaSync Workshop [10] in Berlin, organized by TNO in collaboration with several other FP7 projects.

Apart from the client-side application currently being shown, the underlying elements, which have been based partially on the multi-platform open-source GStreamer framework [11], has been developed specifically to allow for easy integration with NEC's Cloud Offloading framework currently being developed in WP4. As a result, the newly developed synchronization elements can also be used in a cloud-based environment, allowing for synchronization to occur in the cloud, and resulting in a single media stream being sent to the end-device.

As described, the current test application focuses on providing frame-accurate synchronization of multiple heterogeneous media sources on a single device. Although strict frame-accurateness is not important for most of the HBB-NEXT use, there are some more-advanced use cases imaginable in which frame-accurateness is a vital requirement. An example of such a use case is multi-camera picture-in-picture and tiled streaming, where a video stream is divided into multiple tiles, which are streamed independently from each other and recombined on the client-side. By developing the synchronization framework from the ground up to support frame-accurateness, the chosen synchronization methods can be guaranteed to be ready for such advanced use cases in the near future.

Another important aspect of the current application is its ability to work with a number of different streaming protocols for the broadband streams. As part of HbbTV 1.5 [12], MPEG DASH [13] has been designated as the main streaming protocol for the delivery of broadband video content in the

HbbTV context. However, since the standardization of MPEG DASH has only just been completed and support is not yet included in major browsers and mobile operating systems, the role it will end up having in the larger internet streaming context remains to be seen, and there are other streaming protocols vying for the same top spot. One of these protocols is Apple's HTTP Live Streaming (HLS) [14], which is the only streaming protocol supported by Apple devices and with support in recent versions of Android. For this reason, the test-bed has been designed in a way that allows for the synchronization of DVB streams with both MPEG DASH as well as HLS streams.

For more details on the exact synchronization methods used, the reader is referred to D4.2 [6].

3.1.1. Functional Description

The test application allows for different media (audio, video and/or subtitles) from different sources (local, in the form of file or DVB, and remote, in the form of MPEG DASH, HLS or subtitle files) to be displayed concurrently in a synchronized manner.

An example use case where this is useful is one in which the main football match is being broadcasted over the DVB network, while additional camera feeds, such as specific angles, are being sent over the IP network. These additional viewpoints can then be selected by the user to be shown in a picture-in-picture like manner over the main DVB broadcast stream. Another use case is one in which audio tracks aimed at a small audience, such as might be relevant for a small minority of the, to be received over the IP network, thereby reducing the bandwidth required on the DVB network.

In the screenshot Figure*7* shown below, the user is given the ability to select a visual sign language interpreter video as a picture-in-picture video. Using a simple interface the user is also presented with the ability to change the size and position of that video and adjust its transparency. It should be noted that the user interface visible in the screenshot is just for testing purposes. In the future, when actual user testing will be performed, a more elaborate and user-friendly interface will be developed.

Figure 7:

To showcase the accuracy of the synchronization test-bed and highlight its frame-accurateness, the above screenshot shows a setup in which the left and right side of the main video are received as two separate videos and synchronized on the client side. As the screenshot clearly shows, this effect is completely invisible to the user, and thus shows the fact that the synchronization between the two halves is frame accurate.

In order to show what happens when the synchronization is not working properly and what effect even a single frame of de-synchronization has, the following screenshot shows the same setup, but with the synchronization method deliberately set to create a single frame of de-synchronization between the two halves.

Figure 8: Visualization of De-Synchronization

3

The test application has been developed based on the open-source and multi-platform GStreamer framework, to allow for easy integration with the NEC Cloud Offloading framework in the future. The resulting software application can therefore be run on any platform supporting GStreamer.

The demonstrator that is shown at the 2012 MediaSync Workshop will consist of a laptop running the test application and connected to a USB DVB-S2 receiver, a DVB server broadcasting the main video stream, and an Apache server hosting the various MPEG DASH and HLS audio/video streams.

For more details regarding the exact hardware and software implementation and setup, the reader is referred to D4.2 [6].

3.2. Synchronization of IP delivered Subtitles with Broadcast Service

An accessibility services application was chosen in order to test and demonstrate HBB-NEXT synchronization features. While available in broadcast-only services for years, the support of both subtitle and sign language video synchronization in hybrid networks is still a challenge. As well the HBB-NEXT consortium believes that accessibility is a key feature for public acceptance of an open standards platform, making this kind of service a perfect test case from technical as well as from a political perspective. The user experience has been explored extensively in the D2.3.1 user studies.

A first working prototype of the accessibility services application has been implemented upon the result of these user studies and a well-chosen set of system and service requirements, coming from

D2.2. The prototype allows access to both sign language video and subtitles, which can be configured in size and style by the user according to her preferences (see Figure 9). The settings are stored persistently in an individual user profile. The findings of the user experience tests in D2.3.1 suggest that the application is perceived as a major advancement compared to existing broadcast-only, non-customizable accessibility services.

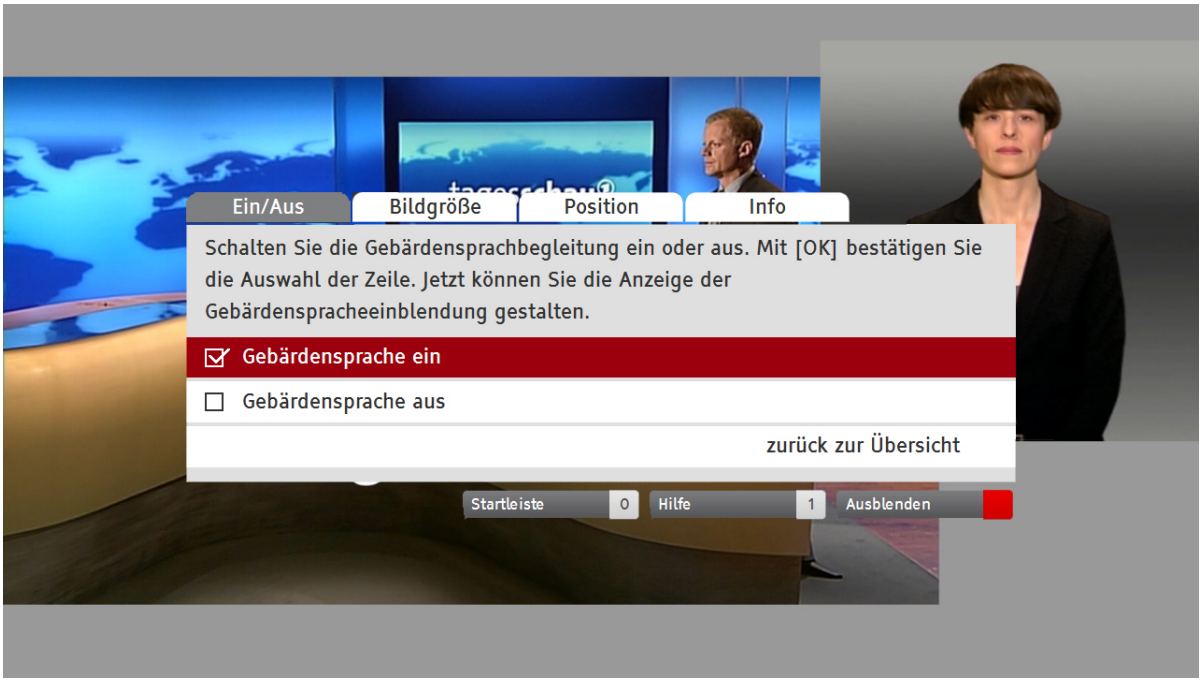

Figure 9: IFA Showcase – sign language interpreter

WP4 in close cooperation with WP6 and WP2 has developed a first proof of concept showcase for IFA 2012[1] exhibition in Berlin. The showcase demonstrated the possibilities for application development combining HbbTV with WP4 enablers. The actual details of the showcase and about the presentation at the fair are described in D4.2 [6] and D7.3.1 [15].

3.2.1. Functional Description

The accessibility services are configured and rendered on the screen via an HbbTV based application. This application is signalled along with the TV service bouquet of a broadcaster and available via an application launch bar. The on-screen configuration menu offers subtitles and sign language video but is designed to integrate further synchronized services like original language subtitles or audio description in the future. The subtitles can be altered in size, background style and position. The sign language interpreter can be altered in size and position. All settings are stored

[1] http://www.ifa-berlin.de

persistently so that they are available automatically when the user tunes into the bouquet / program.

Figure 10: the user settings menu of the accessibility app

Figure 11 shows the hardware to setup the demo application. It requires three PCs or servers, the first one hosting the broadcast server to generate a broadcast HF signal to feed the set-top-box. The second one is the hosting HbbTV application server (e.g. Apache HTTP server). At the receiving end a third PC as DVB/HbbTV/HBB-NEXT client is required and finally a TV-set as the display unit.

The broadcast server requires a modulator board from Dektec [16]. As the client PC has a DVB tuner, Dektec DTA107S2 has been chosen, a device which supports QPSK modulation. The broadcast server and the client PC are interconnected by a coax cable with F-connectors. A local network connects the application (HTTP) server and the client PC.

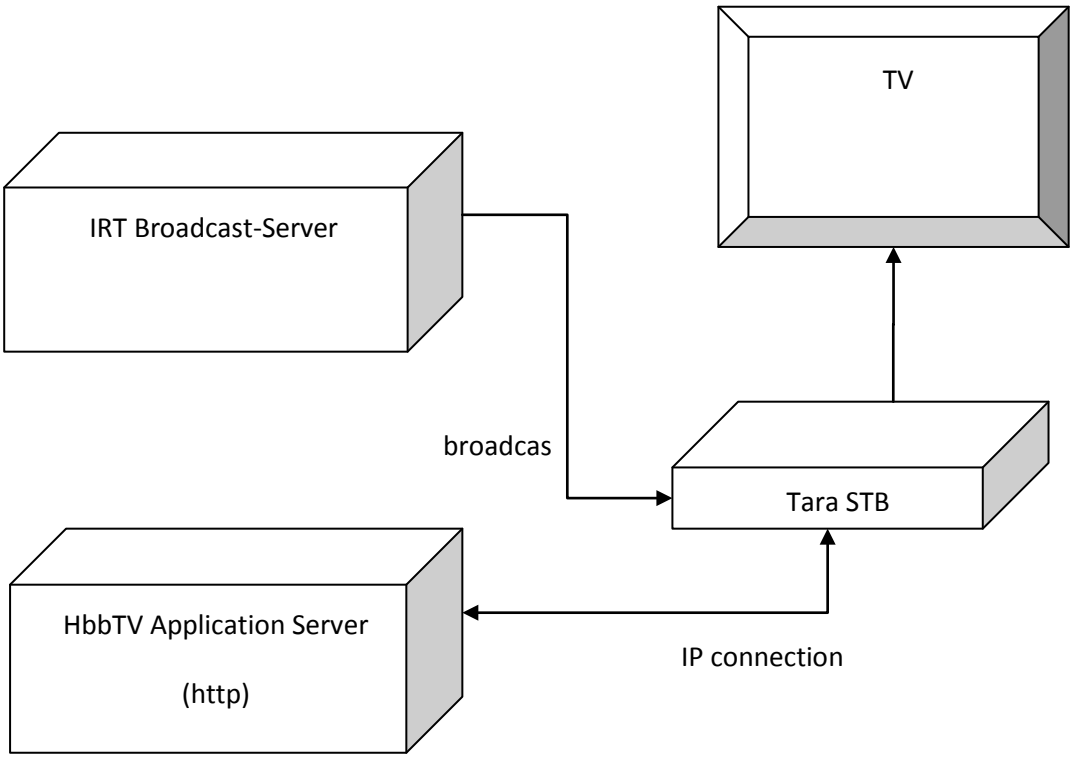

Figure 11: Hardware setup for showcase

3

On the broadcast server a pre-generated transport stream containing the timeline information for synchronization of IP content is used to generate the broadcast signal. For this offline step the "TimelineGeneratorTool" [6] from WP4 is used.

The broadcast server is configured to signal the HbbTV application, which is hosted on the application server as an autostart application, i.e. it will be started when the service is tuned on the TV.

The operating system on the broadcast server is Windows 7 (alternatively Windows XP or the Windows Server equivalents). For the Dektec device the latest drivers need to be installed.

The application server runs an Apache HTTP server to deliver the application files to the STB. It can be the same PC or server as for the broadcast server. If it is a separate PC or server it can either run any operating system that supports Apache. On the Apache installation PHP is required.

The application hosted on the HTTP server is HbbTV based, i.e. it is a set of HTML pages including ECMAscript to implement the application logic. For subtitling the application makes use of the WP4 synchronization enabler [6], which provides the timeline information to the application.

At the client PC, beside the HbbTV 1.0 feature, the additional WP4 synchronization enabler is required to support synchronization of broadcast and IP content with the timeline approach. This was achieved by a Windows Implementation from THM's WP4 enabler testbed, which is designed to run in a Mozilla Firefox Browser with additional HBBTV/HBB-NEXT plugins to extend the browser functionality for the DVB and HbbTV/HBB-NEXT requirements.

3.3. Cloud Service Offloading

3.3.1. Functional Description

The initial demonstration of cloud service offloading will demonstrate how a video processing application such as transcoding video from a broadcast stream (.ts file) into a web-friendly format can be implemented in the cloud. This means demonstrating the following features.

11. An Openstack cloud is setup to provide cloud IaaS service. This includes the creation of virtual machine images suitable for media processing.
12. The system programmatically turns on virtual machines in the cloud when requested to do so.
13. The system setups up virtual machine(s) to perform the following tasks:
14. Media processing pipeline in gStreamer to read the transport stream and provide it to audio/video transcoders.
15. The audio and video transcoders convert the stream from MPEG2 format to WebM format.
16. The transcoded audio and video streams a re-multiplexed into a WebM container.
17. This multiplexed stream is then distributed to a HTTP server capable of chunked file delivery.
18. Finally, the client browser (Chrome) connects to this HTTP server to download the stream and display it in the browser.

3.3.2. Hardware Setup

The main hardware setup is related to the cloud computing platform. This is setup at the IT data centre of one of the partners (NEC) as it is a multi-computer/server setup. Next, suitable security rules are built into the public internet facing firewall to allow controlled access to the cloud computing platform.

The minimum cloud computing platforms a two node installation with one control node and one compute node. The compute node is a high-end server capable of hosting multiple virtual machines that run compute-intensive media processing tasks. This scenario has been setup.

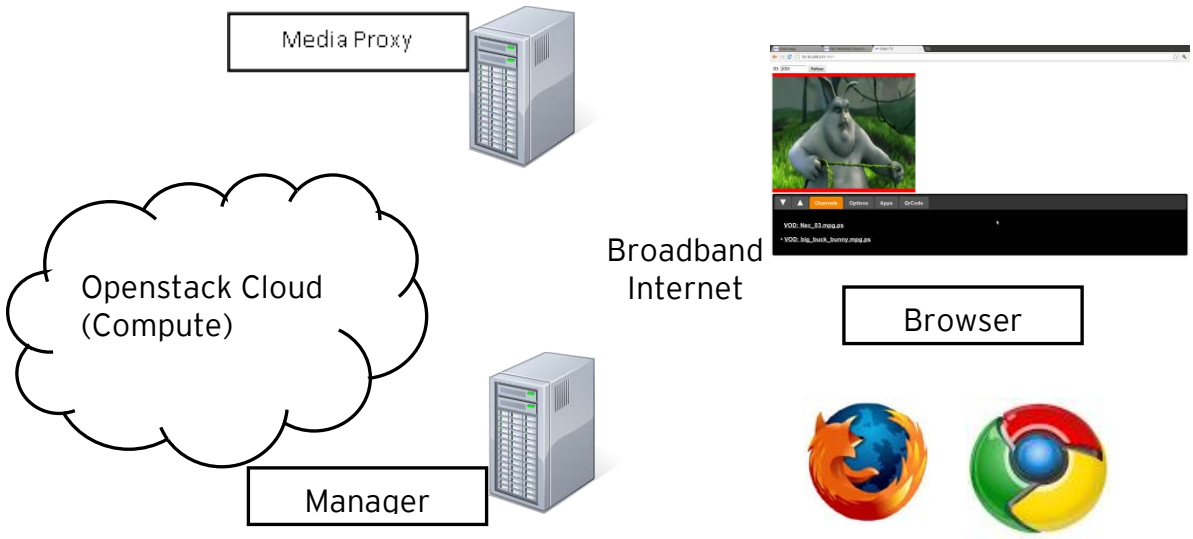

Figure 12: Deployment Diagram of HBB-Next Cloud Offloading Service

Figure 12 shows the deployment diagram of the HBB-NEXT Cloud offloading service. The Openstack compute cloud hosts the media processing virtual machines. The manager orchestrates the entire back end. The media proxy converts gStreamer output from the media processing machines into HTTP chunked format and also handles all client sessions and interactive applications between the service and the user. The user connects to the service via a web browser.

For this prototype at least two machines are required for the Openstack cloud. In addition, two physical machines are used as a manager and a media proxy server. Finally, the client can be a laptop or an Android 4.0 tablet. A high speed internet connection is needed between all entities. An additional gigabit Ethernet network is setup between the cloud machines for fast media stream transfer between the media processing machines.

3.3.3. Software Setup

The following software components are setup:

19. Openstack Essex open-source IaaS: This enables the managed deployment of virtual machines, their connection networks, machine images, security and authentication and object storage.

20. gStreamer (on Linux machine image): This software provides hundreds of media processing filters that can be composed into different media processing pipelines to transform media data (audio, video, subtitles etc.) as required by the application.

21. Node.js with additional libraries for several web frontend features: Node.js is a server-side Javascript-based programming framework for rapid development of content rich web applications.

22. Python with libraries for scientific computing and messaging queue interfaces: Several additional Python libraries such as scikit and PIka are used to construct the various algorithms of the HBB-NEXT Cloud offloading service.

23. RabbitMQ AMQP compatible servers: RabbitMQ is an open source implementation of the AMQP (Advanced Message Queuing Protocol). This messaging service is used by the various managing components of the system, as well as the media processing pipelines.

24. MongoDB noSQL database server: The MongoDB noSQL database server is used for storage of monitoring information.

In addition, a Linux-based machine image was created with the requisite software libraries for gStreamer (including Python bindings) and AMQP libraries. This machine image is launched whenever more processing resources are requested by the system. Then, media processing tasks are run on this.

4. WP5 Test Applications

In this chapter two test applications are presented which partly demonstrate the functionality of the enablers Multi-modal Interface for Multi-user Service Personalisation Engine and Content Recommendation.

4.1. Speaker Identification among a Group of Users

4.1.1. Functional Description

The scenario describes a situation in which multiple users are present in front of the system and speak. The speaker identification function identifies a speaking person and allows the system to provide new features to the users, e.g. invitation screen showing or personalization of offered TV programs based on profiles of different people.

The purpose of this demonstration scenario is to welcome the currently speaking person and show a personalized list of favourite TV programs.

4.1.2. Hardware Setup

For the realization of the scenario “speaker identification among a group of users” two computers are needed, one for the speaker identification application and one for the main application. These computers are connected via Ethernet network (with IP addresses assigned).

4.1.3. Software Setup

The final HBB-NEXT system shall run as a single application integrating multiple modules according to the final system architecture. One of these modules is the speaker identification module which provides information to other modules about speaking users.

In the current stage the speaker identification module is a standalone application running on a Windows computer with a microphone attached. The application utilizes different complex algorithms used in several phases of speaker identification. After identifying a speaking user, the application sends an XML formatted message in TCP packets to the main application which contains information about the identified user and the probability of the correctness of the identification. The main application (running either on the same or on a different computer) listens to a chosen TCP port and reacts on received XML formatted information, e.g. with showing a welcome screen addressing the identified user and showing a personalized list of favourite TV programs.

An example of XML formatted information sent by the speaker identification module is shown in the following figure. The speaker identification module recognized a speaking person as John with

probability of 70%, as Michael with probability of 40% and as Joseph with probability of 20%. The main application has to decide which user has been detected, e.g. by combining it with information from the face recognition module which could lead to a better decision.

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
     <name>SPEAKER IDENTIFICATION</name>
     <users>
          <user>
               <name>John</name>
               <probability>70</probability>
          </user>
          <user>
               <name>Michael</name>
               <probability>40</probability>
          </user>
          <user>
               <name>Joseph</name>
               <probability>20</probability>
          </user>
     </users>
</application>
```

Figure 13: Example of XML sent from speaker identification module

4.2. Group Recommendations

The aim of this test application is to recommend TV events for users or user groups.

The application consists of three parts. The first part is the recommendation engine software, which runs on a PC and creates a list of recommended TV events for a specific user or a group of users on the group context server. An Android application is used to register and unregister users. And at last, an HBB-NEXT application, as an HTML webpage forms the user interface to connect to the server and display the results for the group. A detailed functional description of the three components can be found in D5.2.

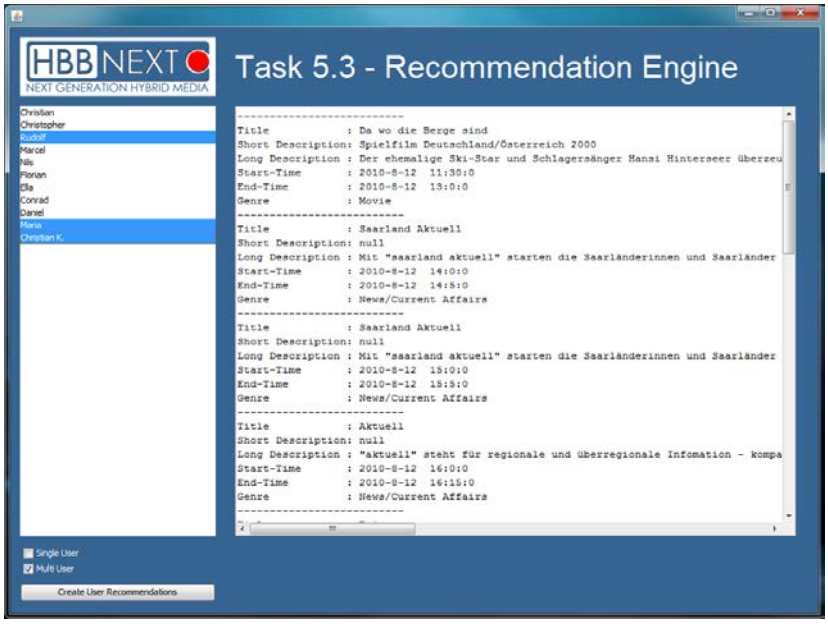

Figure 14: Technical UI to select and retrieve group recommendations

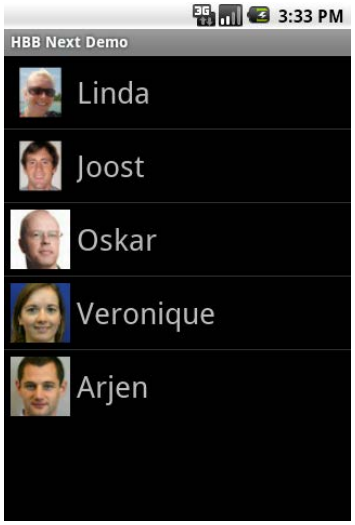

Figure 15: UI on the mobile phone

4.2.1. Hardware Setup

The minimal setup of the demo application is one computer or machine, on which the prototype can be run. To use all abilities a monitor and an Android based smartphone is needed. The monitor is used to display a QR-Code and the smartphone is used to read the QR-Code from the monitor. To connect the smartphone to the computer, a network connection between both devices must be set up.

4.2.2. Software Setup

The prototype runs as a single Java application and was developed with the JRE system library version JRE6 [17]. To get the application work on the test machine, the test machine needs an installed Java runtime environment. This Java runtime environment must be compatible with the version of the JRE system library. Because the application is completely developed in Java it does not matter on which operating system the application is installed.

At the current state of the prototype, the application uses XML files as data source. Both the user profiles and the TV events are loaded from XML files.

The second part of the recommendation engine is the Android application on a smartphone. It is installed on a smartphone and uses the services provided by the computer (on which the first part of the software is running on).

For testing and for demonstrating the multi-user recommendations, the smartphone application offers the ability to login more than one user.

5. Roadmap to Milestone 3

With the first version of enabler and application developments after one year HBB-NEXT reached an important milestone (Milestone 6). However, Milestone 6 also entails a roadmap for the second year of HBB-NEXT until Milestone 3 (PPR2). This roadmap is described in the following sections.

5.1. Work Package 2 Roadmap

For WP2, task 2.1 (Usage Scenarios), task 2.2 (Requirements) and task 2.4 (Business Models) were successfully concluded in the course of year one of the project. All corresponding deliverables were submitted and the partners cooperated fruitfully in all steps. The only remaining task that lasts till the end of the project is now task 2.3 (User Oriented Validation and Evaluation).

The project framework for this is as follows: User Validation Phase 1 was successfully completed and it lasted until end of August 2012. The corresponding deliverable was submitted in time and early results of the tests were fed into Development Cycle 2 as envisaged. Now, User Validation Phase 2 starts on 1 October 2012 and will last until end of March 2013 (M13 – M18). The aim of this phase according to the DoW[2] is the validation of initial components and basic functionalities of the HBB-NEXT enablers. Then, User Validation Phase 3 will follow from 1 January to 31 March 2014. This final phase is not covered in this roadmap, as this is only an outlook up to Milestone 3, the Second Periodic Progress Report, which is due end of September 2013.

Thus, we will focus on User Validation Phase 2 for the coming year. The HBB-NEXT DoW outlines the following structure: While User Validation Phase 1 focused on mock-ups (paper prototypes), Phase 2 is dedicated to testing the output of the project's Development Cycle 2 which culminates in MS6 and produces the initial versions of the HBB-NEXT core components of Work Packages 3, 4 and 5. So, in User Validation Phase 2 we will validate the early applications of these initial technological enablers (software components) which were developed in Task 6.3, covering both functional elements and GUIs) in lab tests involving target users. The results (D2.3.2 – end of March 2013) will be fed into Development Cycle 3 (starts 1 October 2012, ends end of February 2014). The ultimate aim of User Validation Phase 2 will thus be to shape the final software components and applications that will form the HBB-NEXT integrated prototype.

In addition, results of the early validations of User Validation Phase 1 still need to be considered constantly in development phase 3, not only in development phase 2 as was originally envisaged.

[2] Description of work

User Validation 2 will feature lab tests performed by RBB and KU Leuven with target users. The respective methodology will be devised and fine-tuned between these two partners as has been done very efficiently in User Validation Phase 1.

This general approach was fine-tuned at the consortium meeting in Bratislava from 17 to 19.9.2012: Time-wise, the user tests of validation phase 2 will take place in January 2013 (mid-February 2013 at the latest). This will allow for a) sufficient time to compile deliverable D2.3.2 until end of March according to plan and b) to make early results available for the technical work packages by end of February. This is important because Milestone 7, which will result in intermediate versions of the software components, is situated at 1 June 2013 and there will be three months' time to consider these early results for the intermediate versions. The more detailed results then documented in D2.3.2 can then be fed into the final version until the end of Development Cycle 3.

What will be in the focus of the tests? At Bratislava five HBB-NEXT applications were defined which cover all HBB-NEXT enablers and which will feature in the HBB-NEXT prototype platform. As for Milestone 6 the focus was very much on finalising the enablers/technological components as such; the early applications available now at Milestone 6 are functional test applications that can be validated very soon. The only exception is the IFA showcase of Work Package 4. The actual client-side applications, which were specified in Bratislava and their GUIs, will be created now in the early phase of Development Cycle 3 based on, and integrated with the software components of Milestone 6. Early interaction designs for testing the user relevant aspects of these applications then need to be available by the end of this year in order to prepare the various user tests that take place in January and early February. It will be crucial for the success of this next phase that the technical Work Packages 3/4/5 and the monitoring and application Work Package 6 cooperates very closely with Work Package 2. This will be guaranteed by the application team approach described in detail in the roadmap for Work Package 6, see section 5.5. As described there, each application team features a Work Package 2 representative, who is responsible for bringing in the user relevant aspects (based, of course, to a great extent on user validation phase 1) and who will prepare the user validation tests.

5.2. Work Package 3 Roadmap

The main R&D areas of WP3 are multimodal interface, security and identity management and application reputation and trust. The planned work in the coming year is detailed below.

Multimodal interface

Regarding the multimodal interface, STUBA plans to continue on the implementation and enhancement of various methods for user identification with the focus on multimodality (audio/image) and to start implementing multi-level identification for secure authorization, starting from the current design described in D3.2. This will be fed into deliverables D3.3.1 and D3.3.2, both in Month 20 (May 2013), Milestone 7.

STUBA plans to introduce user identification based on iris recognition and 3D face recognition. The main I/O problem of the first approach is to find a suitable device (camera) for precise iris detection and analysis. The second approach will be based most probably on Kinect. Another task for STUBA will be the integration of the selected approaches to one enabler to demonstrate simple scenarios for user identification and system personalization.

Security Management and Identity Management

ST will continue with detailed internal architecture design of Security Manager and IdM. One of the main roles of the Security Manager will be introducing of multi-factor authentication logic for several levels of identification and authentication requirements while the Identity Management will act as a back-end for the authorization processes. We assume to have a strong cooperation with STUBA, as they will provide input data for decision making procedures and policy enforcement. All of this will be then demonstrated in a simple scenario.

It is also planned to introduce the concept of a Key & Certificate management for the whole HBB-NEXT domain as a part of the Security Management. This shall include the design of an authentication and authorization strategy for users coming from different HBB Next domains/providers - visited HBB Next domain.

Trust and Reputation

Regarding the trust and reputation management enabler, it is planned to fully implement and test the enabler with all its functionalities starting from the current design described in D3.2 and the prototype shown in D6.3.1. To this end, we will contribute to both D3.3.1 and D3.3.2, both in M20 (May 2013). The former will provide the documentation regarding the implementation of the enabler, while the latter will be the enabler itself (software). Moreover, we will participate in the internal synchronization point IS3 in M18 (March 2013).

More specifically, we plan to enhance the current prototype of the trust and reputation management enabler to allow it support more advanced functionalities such as the dynamic

selection of the most appropriate reputation computation engine based on the current system conditions and the capabilities of the device in use. We will also study how to preserve the privacy of the users when providing feedback regarding a specific HBB-NEXT application. To this end, we will analyse the potential integration of the trust and reputation enabler with other enablers within WP3 (specifically the Identity management one) and even with enablers from other work packages.

Finally, we also plan to improve the test demo by studying the potential integration of the enabler within the set-top-box provided by TARA.

5.3. Work Package 4 Roadmap

The planned work for WP4 will continue the current work and build upon the developed technologies and solutions from the first project year. Since the work in WP4 is split up in two major fields, named as Synchronization and Cloud Services, the following descriptions will keep this structure according to the tasks. The WP4 work is split up between the partners in order to achieve a maximum scope of key technologies to cover the whole chain of platforms and technologies needed to provide the enablers processing cycle. In this way the partners cooperate to implement their solutions on different platforms and make inter-working their solutions.

Synchronization

One major goal for WP4 is to achieve inter-device synchronization for the same content distributed over different channels. In detail, video from a DVB-S broadcast channel shall be synchronized with the same video streamed over IP via the Internet. The partner THM will build upon its development and research on inter-media timeline synchronization and enable this solution for inter-device synchronization. There, the client application will be enhanced to synchronize the timeline on different HBB-NEXT devices for frame-accurate, synchronized playback. In cooperation with TARA Systems, THM will aim on integrating this client solution for the Broadcom set-top-box to build an integrated demonstrator. The partner TNO will continue its successful work on its gstreamer implementation for synchronization services to support synchronization on PCR as well as on timeline level between different devices like tablets and smartphones. In cooperation with NEC, TNO plans to integrate its gstreamer implementation, to serve the cloud services as part of NEC cloud tool chain for synchronization in the cloud. The partner IRT plans to implement the timeline approach in its DVB Server product to enable it for the generation and insertion of a timeline during live playout. This solution shall provide support for multiple events, i.e. multiple timelines and additional signalling for remote components, to allow a validation on the integrity and accuracy of a shared timeline between different server instances for example. Also IRT will work on

resynchronization of "distributed" service components, delivered in multiple TS into a single transport stream. The partner RBB serves the other partners to provide content samples and application use cases to test and demonstrate their solutions appropriately. In this process RBB will further develop their showcase application to provide inter-device functionality for the demonstrator.

Cloud Services - Scalability

In order to improve the scalability of the Media Platform it is important that the media cloud management layer has full visibility of all running pipelines and client requests. Such a level of visibility will enable the management layer to identify if a running pipeline can be used to satisfy a pending request or whether a new pipeline needs to be set up. In addition, the media cloud management layer has to recognise which pipelines are no longer needed and therefore they can be shutdown.

A key component to be investigated and developed is a data structure and a related matching algorithm. For each incoming media request, the pipeline configuration to serve such a request is generated. This configuration will be matched against the running configurations which are known to the cloud media manager. If no match is found, a new pipeline will be launched. Otherwise, the output of the matching pipeline will be used to serve the request. Likewise, appropriate algorithms have to be developed when pipelines are shutdown in order to decide when they should be removed from the system. First results are expected for February 2013

Cloud Services - Offloading

In order to allow dynamic offloading of processing and rendering to cloud systems on demand, it is essential that the cloud media manager is aware of the client capabilities. Therefore, part of the research will focus on investigating mechanisms for identifying and collecting browser and device capabilities (e.g. supported codecs, display resolutions, etc.). Such information will be made available to the cloud media manager to make decision on the distribution of work between the cloud and the client. As an example of cloud offloading, the task will investigate how media synchronisation of media streams can be provided to the client. In particular, depending media streams (for example, news and related sign language stream) are combined in a synchronised manner in the cloud and delivered as a single stream to the end-user device. First results for this are expected for September 2013.

Cloud Services - Media Adaptation

After year two, as part of the offloading of processing and rendering in the cloud, the task will spend the last three month investigating the media adaptation aspect. Any media provided by the cloud has to be adapted (i.e. transcoded) to the resolution and one of the supported codec of the client device. Moreover, techniques to transmit video in the appropriate protocol/container are also needed. First results are expected for December 2013.

5.4. Work Package 5 Roadmap

Work Package 5 includes several fundamental technology developments, including multi-model interface technologies, personalisation engine technologies and content recommendation technologies.

Work Package 5 will also carry out an integration effort, together with Work Package 3, and coordinated from Work Package 6, achieving a demonstration of the use case that is described above. The integration effort includes user tests.

The paragraphs below will first describe the fundamental technology developments, and then the integration plans.

Multimodal interface

By mid-2012, HBB-Next partner STUBA has developed several implementations of multi-modal interfaces for both Work Package 3 and 5. These include user identification by face and voice recognition. Next steps are the recognition of multiple users simultaneously (multiple simultaneously viewed faces, multiple simultaneously heard speakers) and combining face and voice information to increase the likelihood of correct person identification. Additionally, there will be continued activities with speech recognition for voice commands, and gesture recognition for gesture control. Integration activities are planned to use the outputs from the multimodal interfaces in HBB-Next applications.

Personalisation Engine

HBB-Next partner ST will be building and integrating a personalisation engine based on non-preference-related parameters like age, gender and time-of-day. The framework will enable HBB-Next application to provide a level of personalisation.

Hybrid content recommendations with metadata integration and multi-user support

HBB-Next partners IRT, THM and TNO will collaborate on the realisation of a hybrid content recommender, using metadata integration technology being developed by IRT, content-based filtering algorithms by IRT, collaborative filtering algorithms by THM, the PREF recommender framework of TNO, and multi-user recommender solution being developed by TNO. The system will combine live and on-demand content. It will implement Java and RESTful APIs to provide standardised access to the components.

Implicit ratings

When the HBB-Next recommendation concept was presented to content providers and recommendation service providers in an HBB-Next feedback workshop in Netherland, early 2012, it was recognized that explicit ratings (user explicitly rates content by giving points) is ineffective in a passive TV watching context. Therefore HBB-Next has initiated an additional, not previously planned activity on implicit ratings, i.e. deducing user preferences from watching behaviour. Research question include getting access to the clickstream, identifying the users present, and automatically interpreting the clickstream. This activity will be carried out in collaboration between IRT, TNO and KuL, with the IRT focus on the clickstream collection, and the TNO and KuL focus on the automatic interpretation.

Notification service

HBB-Next partner IRT is developing a notification service as a central architectural element for the integration of HBB-Next applications. The notification service will be used to cleanly separate output from the multimodal interface, user identification and authentication on the one hand, and applications like the content recommender on the other hand. It will also be studied how the notification service can be used in the process of collecting clickstreams.

User tests

HBB-Next partner KuL will carry out user tests related to the activities in WP5, most likely including the interaction with the multimodal interface, the impact of personalisation on the user experience, the interaction of users with (multi-user) content recommendations, and the user aspects of implicit ratings.

5.5. Work Package 6 Roadmap

Work package 6 has two running tasks. Task 6.1 to define and document the system architecture and task 6.3 to develop the end user applications and to integrate the enabler into an integrated prototype for showing the capabilities of the HBB-NEXT platform.

Task 6.2, that designed application designs by using paper prototypes and mock-ups, already finished and fed back the results into WP2 for the first round of end user validation.

System architecture

The system architecture is documented in the deliverable series D6.1.x. The first version has been delivered in July 2012. The next version is planned for September 2013.

The system architecture document shall be a specification of the HBB-NEXT platform and shall provide sufficient information to build and provide HBB-NEXT applications and it shall be the basis for HBB-NEXT contributions to standardization of technical specifications including technical requirements, APIs and protocols.

The following is the roadmap of T6.1 until the next version of D6.1. In order to align the documentation of the external interfaces of the modules coming from the technical work packages (3, 4, 5), WP6 provides guidelines on the documentation in D6.1 chapter 3. These guidelines will be checked until November 2012 and then communicated to the technical work packages. The next step will be to look at the current state of the HBB-NEXT architecture and make a check on which interfaces need to be documented in D6.1, i.e. which ones are important for HBB-NEXT applications and the HBB-NEXT platform. This shall be done until end of 2012.

In order to start application development the API definitions for all important modules will be done by January 2013. The first draft of D6.1.2 is planned for May 2013 which is aligned with the next release of the enablers from the technical work packages. The final draft of D6.1.2 will be submitted in October 2013. Updates on the architecture, APIs etc. will be provided at the end of the project in March 2014 (D6.1.3).

Application Development and Integrated Prototype

At the third HBB-NEXT Consortium Meeting in Bratislava, a detailed approach was devised for Task 6.3. This task is intertwined with the technical Work Packages 3, 4 and 5, with the user validation task (T2.3) and with Task 6.1. It is planned that the integrated prototype shall cover all relevant HBB-NEXT enablers. The inspiration for this approach was the very fruitful and productive cooperation in

creating the IFA showcase of Work Package 4; the cross work package cooperation between WP2, WP4 and Task 6.3 ran very smoothly, and resulted in an interesting IFA showcase. This was taken now as best practice example for the future work coordinated by Task 6.3.

The basic approach is as follows: While the basic, user-oriented design of each application will be elaborated in Work Package 2, application team members of the respective technical Work Packages 3, 4 and 5 will bring in the knowledge of the platform and their modules. The actual applications will mainly be developed in the context of Task 6.3.

Each application team of task 6.3 has one leader who coordinates the application work from start to finish and reports about status, plans, problems and progress. Each team covers representatives from all work packages relevant for a certain application:

- At least one person from WP2, who is responsible for the interaction design and the user evaluation of each application (context of user validation phase 2).
- At least one person from WP6, who is/are responsible for the end user application and the smooth integration into the system architecture and open APIs. They also coordinate the team.
- People from the technical work packages (WP3, 4 and 5) who are responsible for the needed background development and who will later integrate the results of user validation phase 2 into their intermediate and final software versions.

In Bratislava not only this strategic approach was coordinated, but also the end-user applications that shall be developed to showcase the HBB-NEXT platform milestone 3, were devised. It was decided to implement a set of applications that cover the different technological enablers developed by HBB-NEXT. The idea is a portal where all applications are bundled and where they can be accessed by the user. This could take the form of a TV launcher bar (red button principle currently used in HbbTV). The end-user applications including the portal application shall run on the set-top-box outlined in appendix A as one major component of the integrated prototype.

Each application covers several enablers and demonstrates how they might be combined. The applications are:

a) Enhancement of the settings/**accessibility application (IFA Showcase)** to show inter-media/device synchronization and cloud offloading, multi-user identification.
IRT is the leader of this application team and also responsible for the application development (WP6). RBB will provide a concept for the technical goals in inter-media/device

synchronization, cloud offloading and multi-user identification. The user interface design will be adapted and the application will be evaluated by user tests (WP2 and WP6).
TNO and THM are responsible for the background development related to all synchronisation aspects (WP4).
ST collaboration is related to the cloud offloading (WP4) and the identity management (WP3).

b) The **Open Recommendation application** covers broadcast and internet content, including user identification (multi-modal interface), single user and group recommendation, multi-level authentication.
This recommendation application computes the likely interesting video content for a single appropriate content for all present viewers in front of the TV. It covers many TV channels and perhaps Internet content providers such as YouTube. The basis for recommendations is a user profile and login functionalities, the login should work out trough different ways, e.g. voice or face recognition. Using this application "multi-level authentication" can be demonstrated. For example, for playing age-restricted programmes the security level is higher than for playing non-age restricted programmes.
TNO is coordinating this application team and together with IRT and THM responsible for the background development plus implementation (WP5 and WP6). All issues about user management, especially multi-level authentication are assigned to ST (WP3).
KU Leuven will provide early interaction designs, and will be responsible for the evaluation of the demonstrator. The actual application will be built by IRT (WP6).
STUBA is responsible for the multi-modal interface, e.g. login via voice or face and ST will develop the identity management (WP3).

c) The broadcaster **EPG application** covers channels which are associated with one broadcaster and includes user identification, user management, single user and group recommendation, and multi-level authentication.
This application will be based on an existing HbbTV EPG application and will be extended by including recommendations for multiple users. The technical basis therefore is user identification and user management. As for the enhancement of the existing EPG, 2nd screen usage will also be studied.
IRT is leader of the application team and responsible for the coordination and communication in WP6, related to integration, architecture and APIs. The HbbTV application development is also the task of IRT in close cooperation with RBB (WP6).

RBB will create the enhanced user interaction concept and finally the new design to show all target functionalities. This includes usability tests and (WP2 and WP6) evaluation about the acceptance of such a service in the ARD/RBB target group.
IRT, THM and TNO are the technical developers for the personalized recommendation engine. For this personalisation aspect login functionality and user identity management (developed by ST – WP3, WP5) have to be integrated.

d) **Application store** including multi-level authentication, application trust and reputation management and user identification
The idea of this application is that a user is entering the room and a "basic identification" module (face recognition for example) recognizes him/her. The personalized app store will appear and provide the appropriate applications. When the user for example tries to buy an application the user needs a "better" authentication (higher security level), otherwise he/she is not allowed to do that.
NEC leads this application team and is also responsible for the reputation management in this application store. A click dummy and parts of integration will be provided by NEC to KU Leuven in December.
KU Leuven will organize usability tests in January and the results will be fed into development phase two components and integration happens by September 2013
STUBA and ST are the technical developers and will provide login functionality and user identity management (WP3).

e) **Interactive TV application** including multi-screen usage and synchronization
Here the idea is to develop an HbbTV-based application that enables live user interaction with a broadcasted programme. RBB will define a scenario; afterwards IRT will create a matching application team. This application is still optional, a final decision has not yet been made, or else it might be done but slightly later than the other apps.

By the end of 2012 a "click dummy" for each application has to be available for user validation sessions in January and February (see WP2 roadmap). The actual application development will start in early 2013, with a first integration phase from March to May and a second one in July 2013.

6. Conclusion

At Milestone 6 of HBB-NEXT this document presents two major project strands: 1) The application development efforts and 2) A general roadmap for year two of the project. As to application development, the work in the context of Task 6.3 of HBB-NEXT, the early applications are now in place in time for Milestone 6 as was envisaged. Apart from the work in work package 4 (IFA Showcase) which is already more advanced, these are early test applications rather then finally conceived end user oriented applications. The aim at the end of Development Cycle 2was to have such a first generation of test applications which demonstrate the capacities of the version 1 of software modules from work packages 3, 4, and 5. This way, attractive early prototypes for the project enablers are now in place and were already presented at a number of exhibitions, conferences and workshops.

Now Development Cycle 3 has started. At the Consortium Meeting in Bratislava in September 2012 the project partners jointly elaborated an approach how to fine tune this Cycle up to Milestone 3 (End of Project Year 2). This concerns both the further development of the software modules up to Milestone 7 (May 2013, Intermediate version of enabler software modules) and also the enhancement of the HBB-NEXT applications demonstrating the functionalities of these modules (September 2013, Result of intermediate integration process and Prototype Releases). The Roadmap presented in Chapter 5 of this document describes the plans by work packages and enablers. This includes also the definition of the interdependencies of development and validation, i.e. the fine tuning of work package 2/User Validation Phase 2.

Basically, five HBB-NEXT applications were defined which together cover all HBB-NEXT enablers and which, after the final integration at the end of 2013 will feature in the HBB-NEXT prototype platform. Here, Task 6.3 is intertwined with the technical Work Packages 3, 4 and 5, with the user validation task (T2.3) and with Task 6.1. Each of the five applications is in the responsibility of one specific application team which works in an inter-workpackage approach: While the basic, user-oriented design of each application will be elaborated in Work Package 2, application team members of the respective technical Work Packages 3, 4 and 5 will bring in the knowledge of the platform and their modules. The actual applications will mainly be developed in the context of Task 6.3. Each application team of task 6.3 has one leader who coordinates the application work from start to finish and reports about status, plans, problems and progress. Each team covers representatives from all work packages relevant for a certain application. The app team approach

ensures an early inter-work-package and inter-task integration, since everything will be agreed upon within the app teams.

On the actual application side the next step will be to develop more end user oriented applications. Till end of December a prototypical version of click dummy applications for the five selected applications is to be ready. These will undergo a user testing till February. The early user test results will allow a user oriented prioritization of functionalities and will help to enhance the graphical user interfaces. Likewise the results will be fed into the intermediate versions of the enabler software modules due end of May 2013 for Milestone 7. The actual application development phase with the intermediate integration process will then start in March 2013 and last until May with a first integration meeting during the next Consortium Meeting in early March 2013. The integration will take place among the different enablers while some enablers, like synchronization will also be integrated to run on the TARA set top box.

The intermediate integration process will culminate in the first version of integrated software modules with tested applications which will be available at Milestone 3, at the end of project year two. The project plans to present to show these intermediate demos at IBC 2013. These versions will then be enhanced and fully integrated for Milestone 8, the HBB-NEXT prototype.

7. References

[1] Apache Foundation. Apache tomcat (2007). URL http://tomcat.apache.org/

[2] Recordon, D., Fitzpatrick, B.: OpenID authentication 2.0-final (2007)

[3] Oracle. Java SE. URL http://www.oracle.com/technetwork/java/javase/downloads/index.html

[4] F. Gómez Mármol, M. Kuhnen, G. Martínez Pérez, Enhancing OpenID through a Reputation Framework, in: Proceedings of the 8th international conference on Autonomic and Trusted Computing, ATC'11, number 6906 in LNCS, Springer-Verlag, 2011, pp. 1–18.

[5] HBB-NEXT deliverable 3.2: "DESIGN AND PROTOCOL: User ID, Profile, Application Reputation Framework"

[6] HBB-NEXT deliverable 4.2: "DESIGN AND PROTOCOL: Middleware Components Content Synchronization and Cloud Offloading"

[7] HBB-NEXT deliverable 5.2: "DESIGN AND PROTOCOL: Multimodal Interface and Context Aware Recommendation Engine"

[8] Hybrid Broadcast Broadband TV (HbbTV): ETSI TS 102 796 V1.1.1 (2010-06)

[9] Ruby On Rails: http://rubyonrails.org/

[10] MediaSync 2012 workshop: http://www.icin.biz/tutorials/2012#workshop1

[11] GStreamer, open source multimedia framework: http://gstreamer.freedesktop.org/

[12] HbbTV 1.5: https://www.hbbtv.org/pages/about_hbbtv/specification.php

[13] MPEG DASH: ISO/IEC 23009-1, 2012.

[14] Apple HLS: https://developer.apple.com/resources/http-streaming/

[15] HBB-NEXT deliverable 7.3.1: "1st Report on Dissemination"

[16] http://www.dektec.com/

[17] http://www.oracle.com/technetwork/java/javase/downloads/index.html

8. Abbreviations

MPEG DASH	MPEG Dynamic Adaptive Streaming over HTTP [13]
HLS	Apple HTTP Live Streaming
HTTP	Hyper Text Transfer Protocol
QPSK	Quadrature Phase-Shift Keying (modulation type on digital TV)
JRE	JAVA runtime environment

9. Annex: Prototype Set-Top-Box

The HBB-NEXT terminal device is a major component of the HBB-NEXT system architecture and acts as the interface between the HBB-NEXT network services/applications and the end-user. Therefore, the HBB-NEXT terminal must be capable of presenting various types of media data, of running interactive and downloadable applications and of receiving input from the end-user. Typical primary HBB-NEXT terminal devices will be TV sets with integrated digital TV decoders and set-top-boxes connected to a TV set. They have to be compliant to the HbbTV standard [8] offering a basic framework for hybrid broadcast and broadband services. Today, such devices are publicly available and implement state-of-the-art technology though HBB-NEXT is extending this framework. Some modules like the AV content synchronization will be implemented on the HBB-NEXT terminal prototype.

For the development of the HBB-NEXT integrated prototype application, TARA Systems provides a powerful reference set-top-box which has been supplied by the chip vendor Broadcom. This set-top-box BCM97346SFF acts as the reference design for Broadcom's decoder chip BCM7346. Target applications for this system-on-chip (SoC) are set-top-boxes and TV sets for the connected home.

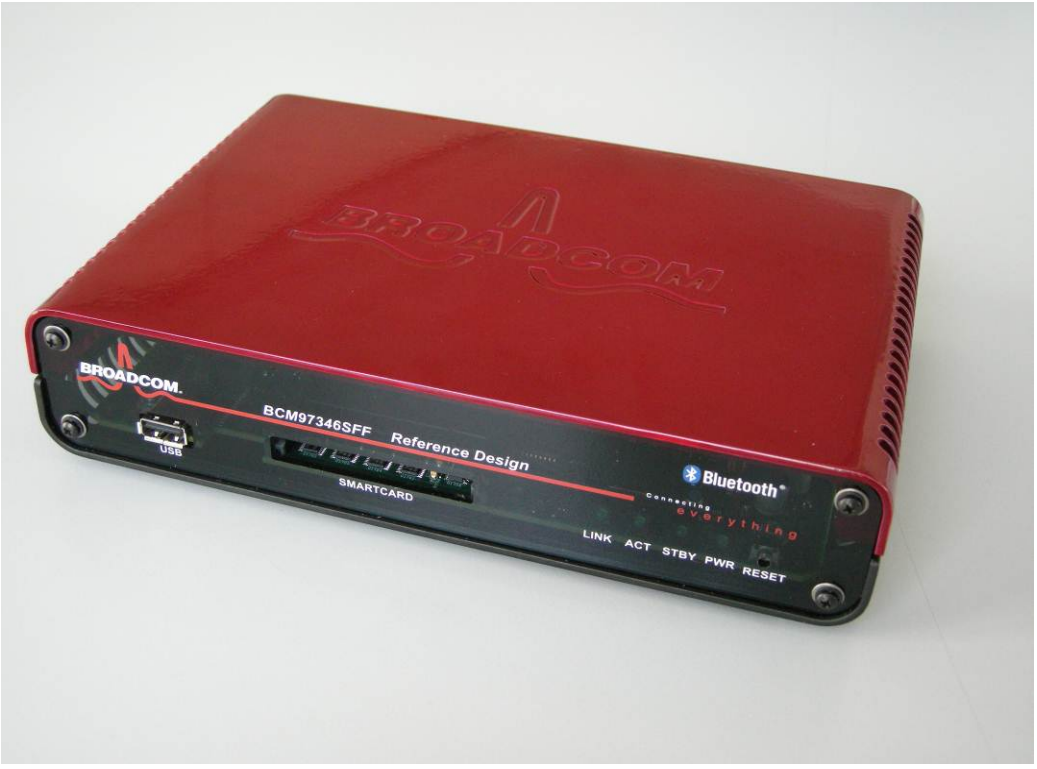

Figure 16: Broadcom reference set-top-box for chip-set BCM7346

The following list outlines the major hardware features of the device BCM7346:

- 40nm technology
- MIPS 5000 Core @ 1.3GHz with >3KDMIPs
- OpenGL ES 2.0/ VG 1.1
- HDMI 1.4a (3DTV)
- HD + HD PIP Display
- Dual SATA3
- Integrated dual DVB-S/S2 demodulator and tuner

The software of the set-top-box can be divided into three major parts:

25. Linux OS, drivers and Codec firmware supplied by Broadcom
26. Embedded CE HTML5 browser supplied by Opera
27. DVB middleware software Inaris including the HbbTV plug-in for extending the HTMl browser and the graphical user interface (GUI) application TVolution supplied by Tara Systems

Figure 17 shows the overall software architecture of the prototype set-top-box. In fact, the mentioned CE HTML Browser is the Opera browser and the driver and OS is Broadcom's.

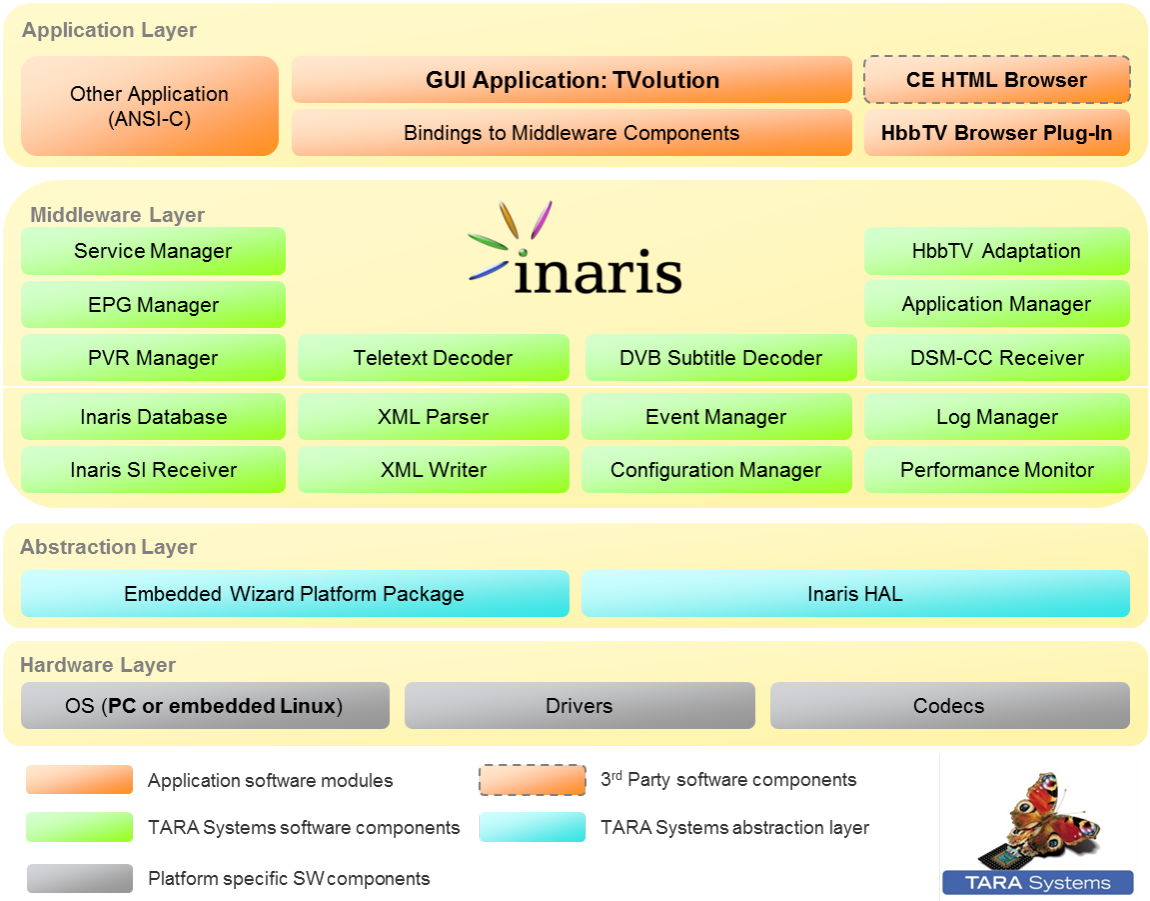

Figure 17: Set-top-box software architecture