This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n: 761999

EasyTV: Easing the access of Europeans with disabilities to converging media and content.

Content adaptation using DASH streaming services

EasyTV Project

H2020. ICT-19-2017 Media and content convergence. – IA Innovation action.

Grant Agreement n°: 761999

Start date of project: 1 Oct. 2017

Duration: 30 months

Document. ref.: D4.4

Disclaimer

The EasyTV Consortium consists of the following partners:

	Partner Name	Short name	Country
1	Universidad Politécnica de Madrid	UPM	ES
2	Engineering Ingegneria Informatica S.P.A.	ENG	IT
3	Centre for Research and Technology Hellas/Information Technologies Institute	CERTH	GR
4	Mediavoice SRL	MV	IT
5	Universitat Autònoma Barcelona	UAB	ES
6	Corporació Catalana de Mitjans Audiovisuals SA	CCMA	ES
7	ARX.NET SA	ARX	GR
8	Fundación Confederación Nacional Sordos España para la supresión de barreras de comunicación	FCNSE	ES
9	Sezione Provinciale di Roma dell'Unione Italiana dei ciechi e degli ipovedenti	UICI	IT

PROGRAMME NAME:	**H2020. ICT-19-2017 Media and Content Convergence – IA Innovation Action**
PROJECT NUMBER:	761999
PROJECT TITLE:	EASYTV
RESPONSIBLE UNIT:	UPM
INVOLVED UNITS:	CERTH
DOCUMENT NUMBER:	D4.4
DOCUMENT TITLE:	Content adaptation using DASH streaming services
WORK-PACKAGE:	WP 4
DELIVERABLE TYPE:	Prototype
CONTRACTUAL DATE OF DELIVERY:	31-08-2019
LAST UPDATE:	
DISTRIBUTION LEVEL:	PU

Distribution level:

PU = *Public,*

RE = *Restricted to a group of the specified Consortium,*

PP = *Restricted to other program participants (including Commission Services),*

CO = *Confidential, only for members of the LASIE Consortium (including the Commission Services)*

Document History

VERSION	DATE	STATUS	AUTHORS, REVIEWER	DESCRIPTION
v. 0.1	24/05/2019	Draft	UPM	Table of Contents definition and document structure
v. 0.2	9/07/2019	Draft	CERTH	Refined table of content
v. 0.3	15/07/2019	Draft	UPM	First version of Chapter 01, 03, 04, 05, 09
v. 0.4	18/07/2019	Draft	UPM	Refined table of content. New Chapter: Chapter 8
v. 0.6	26/07/2019	Draft	Salim Gannoum (CERTH/ITI) Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Chapter 02, 06, 07
v. 0.7	29/07/2019	Draft	UPM	Chapter 8
v. 0.8	31/07/2019	Draft	Salim Gannoum (CERTH/ITI) Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Addition to Chapter 02, 08

v. 0.9	23/08/2019	Draft	Jordi Fabregat (CCMA) Giuseppe Vitolo (ENG)	

Definitions, Acronyms and Abbreviations

ACRONYMS / ABBREVIATIONS	DESCRIPTION
DASH	Dynamic Adaptive Streaming over HTTP
DASH IF	DASH Industry Forum
HAS	HTTP based Adaptive Streaming
HbbTV	Hybrid Broadcast Broadband TV
HBMM	Hybrid matchmaker
HDS	HTTP Dynamic Streaming
HLS	HTTP Live Streaming
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
MPD	Media Presentation Description
MSS	Microsoft Smooth Streaming
QoE	Quality of Experience
RBMM	Ruled based matchmaker
STMM	Statistical matchmaker
XML	eXtensible Markup Language

Table of Contents

List of Figures

List of Tables

Executive Summary

This document corresponds to the deliverable D4.4 and this is a revised version of the deliverable D4.1 “Content adaptation using DASH streaming services” of the WP4. The deliverable D4.1 provided general information about the adaptive streaming. General information such as the basic operation of this type of technology, different solutions for content delivery (focusing on MPEG DASH), methods and tools for generating the necessary files and segments for playing the audiovisual content using DASH, the integration of the DASH client into the application and the functions used in the application to change the audio tracks or the language of the subtitles.

However, this deliverable is focused more on describing the developments and modules implemented in the EasyTV project for the content adaptation. This is, an editor for modifying a general MPD with other access services created in the project, a module in charge of selecting the needs and preferences of the end users, and a module in the client application to customise the audiovisual content and provide the services. All the developments mentioned above are integrated into the project.

This document is divided into nine different chapters:

Chapter 1 provides a brief introduction to the project, some of its objectives and the use of MPEG DASH as a technique of delivering content.

Chapter 2 a brief description of the content personalization process and the use of MPD files.

Chapter 3 describes the structure of a general MPD, the structure of a modified MPD within the EasyTV project and the implementation of the editor that adds the access services into the MPD. In this chapter, an example of a modified MPD with two access services is provided.

Chapter 4 provides information about the module implemented in the client application for reading the MPD, interpreting the available information and providing the services to the end users.

Chapter 5 focuses on the integration of the MPD editor, along with other modules in the EasyTV platform. Moreover, the workflow of the edition of a MPD is explained when a new service is created.

Chapter 6 a description of the way of the EasyTV client interacts with the hyper-personalization component.

Chapter 7 a collection of user cases scenarios that describe the workings of the content personalization.

Chapter 8 mentions the main innovation points in the project related to the content adaptation.

Chapter 9 summarizes the conclusions obtained in the previous deliverable and those obtained in this revised version.

1. INTRODUCTION

The EasyTV project is presented with the aim of providing innovative solutions to facilitate the universal access to audiovisual services for people with disabilities (for example, mainly visual and hearing disabilities or elderly people who typically have problems to perceive the video or audio).

In the general architecture of the project, although there are different solutions for the distribution of the content up to the final users through the Internet, adaptive streaming solution has been chosen due to its advantages. Among the different adaptive streaming solutions (HLS - HTTP Live Streaming by Apple, MSS – Microsoft Smooth Streaming by Microsoft or HDS – HTTP Dynamic Streaming by Adobe), the MPEG DASH (Dynamic Adaptive Streaming over HTTP) as international solution is used. Moreover, the EasyTV project includes the use of the HbbTV 2.0 (Hybrid Broadcast Broadband Television) specification [1], with the novelty of the multidevice scenarios where second screens devices are connected with a main screen (typically a hybrid television) and where there is synchronised content on all devices. MPEG DASH has been adopted as adaptive streaming solution, it improves the QoE on client side and optimize costs on CDN. MPEG DASH has been adopted in the HbbTV standard as a technique for delivering broadband content through the Internet network.

In deliverable D4.1, it was mentioned the different techniques to deliver the content and the advantages in the use of the HTTP (Hypertext Transfer Protocol) protocol and adaptive streaming (HAS, HTTP based Adaptive Streaming). Basically the main idea is that in adaptive streaming, the intelligence is on the client side and it is the responsible for requesting periodically the bitrate and the quality that best satisfies the device capabilities and the network conditions. There are significant advantages for the end users because it is not necessary neither to download an important amount of data to start the playback, nor to have a significant buffer size in the client. Moreover, it allows a good QoE (Quality of Experience) for both excellent and poor network connections. For poor connections a content with low bitrate and resolution will be used and for excellent connections, whenever possible, the highest quality will be offered by the content provider or the broadcaster. They have an important role in this type of scenarios because they have to encode videos at multiple bitrates and resolutions.

In the previous deliverable, also it was explained the tasks and tools related to the preparation and generation of the content to stream it. In this project, CCMA is in charge of providing the content, fragmenting it in small segments (typically between 2 and 10 seconds), creating the MPD (Media Presentation Description) manifest file, and storing the content on an HTTP web server.

In the project, the creation of different access services are contemplated and they are included in the MPD. Analysing the adaptive streaming workflow (in particular, the MPEG DASH solution), the MPD is modified with information of the new access services available for a specific content. In this way, when starting the session, the client application receives the manifest file that describes the structure of the content and the available access services. Depending on the user profile and the preferences of the end user, provided by the Hyper-Personalisation module, the client application shows a specific visual design and customisable services.

Using the MPD as a file that indicates the access services, it is not necessary to create and send additional files. However, a new module in the client application has to be created for understanding this additional information because the DASH client can interpret only the information defined in the standard (this is, the periods, the adaptation sets, the representations and the segments). Knowing the available access services, the application will show those that best satisfied the needs and preferences of the final users. Different tests have been carried out to verify that a modified MPD does not cause problems and the DASH client plays correctly the audiovisual content.

In the following chapters, both the MPD editor and the client application module for reading the manifest file are described. Also it is explained the integration of these developments into the EasyTV project and the communication between the Hyper-Personalisation module with the client application.

2. USING MPEG-DASH FOR CONTENT ADAPTATION IN EASYTV

Personalizing the audio-visual content for the user is one of the EasyTV platform goals. The responsible component for this task is the hyper-personalization framework. Figure 1 shows an architectural overview of hyper-personalization place in EasyTV platform.

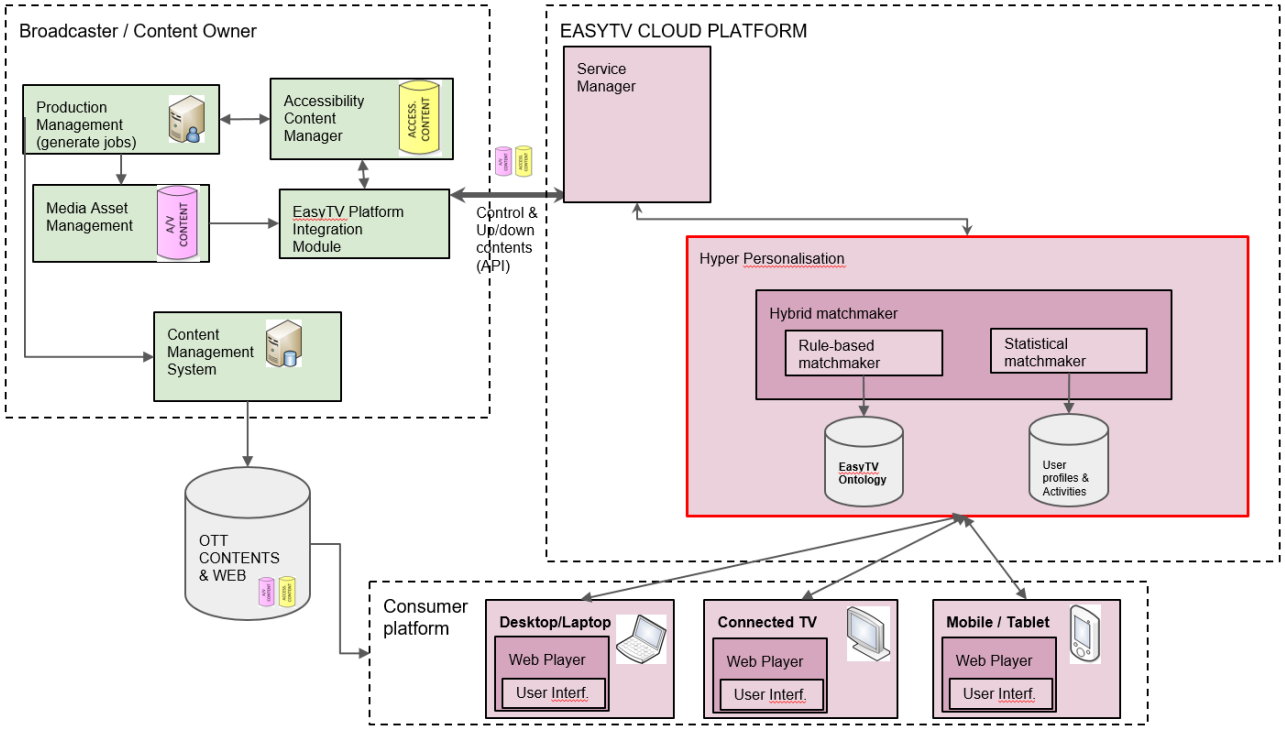

Figure 1: Integration of the hyper-personalization into the platform

Content personalization refers to the task of satisfying the user content accessibility preferences and/or suggesting new ones. User preferences are declared in the user profile and are of two types UI preferences and content accessibility preferences. Accessibility services available to a specific content are indicated by a special tags that are added to the content's MPD file.

The hyper-personalization framework uses a collaborative, content-based and hybrid matchmaking approach [2], to best fulfil the user content needs. More specifically, it tries not only to fulfil the user content accessibility preferences, but also to suggest new ones in case it can be done. In cases where the user needs cannot be fulfilled by the available accessibility services an alternative set is suggested. These are extracted from available user's data and are considered a best match for the user disability on the one hand, and a very good substitution for the lack of requested accessibility services on the other. More specifically, there are three main cases:

- **Flat match**: the user content accessibility preferences can be fulfilled by the available content accessibility services. This is a straightforward match between the user needs and available content accessibility services; this case is the simplest one. Users that belong to this case are considered experienced users, in addition to users that do not require any help to access the content.
- **Flat match with suggestions**: the user content accessibility preferences can be fulfilled by

the available content accessibility services. However, there are strong indications in the user profile that further suggestion for accessibility services can be made.

- **Best substitution suggestions:** due to lack of accessibility services, the user needs must be fulfilled with other accessibility combination. This approach requires the full capabilities of the matchmaking process in order to find the best substitution for the user preferences. By full capabilities we actually mean what similar users have to suggest and what actually the user profile indicates.

Table 1: Content adaptation cases

Matching case	Description
Flat match	I need A, and MPD files has A => Apply A
Flat match with suggestions	I need A, and MPD files has A => Apply A and the personalization process suggests to set B also.
Best substitution suggestions	I need A, and MPD files lacks A => the personalization process suggests to set B_1, B_2, …

Indicatively Table 1 sums up these cases. The content personalization process is able to distinguish between these cases in order to avoid making unnecessary suggestions to users that do not need any. This is achieved by the combinatory output of the collaborative and rule-based. The collaborative approach indicates what similar users prefer and the content-based approach what the user information has to suggest.

2.1. Content adaptation process

Content accessibility service is a service that enhances the accessibility of audio-visual content. The audio-visual content is encoded using sound and picture, which are perceived using sight and hearing senses. The sharpness of these two senses varies between users.

Content accessibility services are of two types: transformational or enhancement services. Transformational services converts the content encoded in one form into another, thus are further categorized into audio-to-visual or visual-to-audio. For example, sign language and sound detection are services that transform audio content into visual, and audio subtitle from visual to audio. On the other hand, enhancement services improve the information accessibility of a specific channel, for instance, face detection enhances the visual content and audio language enhance the audio content, Figure 2 illustrates these facts.

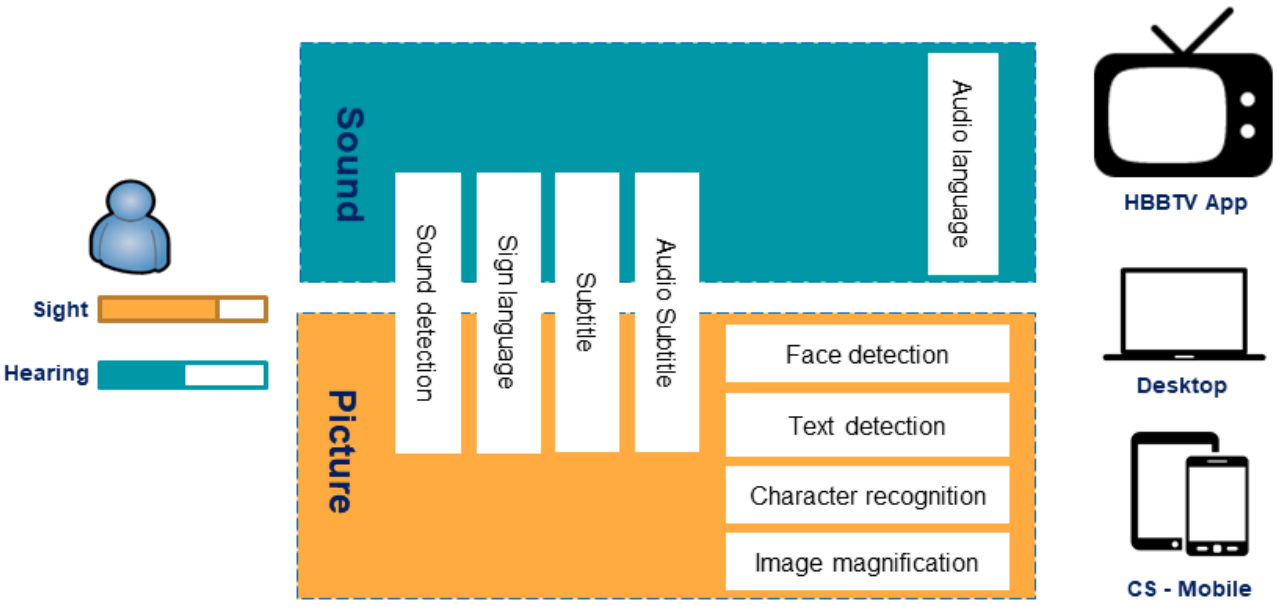

Figure 2 Audio-visual content, accessibility services and user senses

Following this skeptic, the content adaptation process consist mainly of identifying the user need for assessment and then fulfilling them with proper content accessibility services. It must be noted here that multiple accessibility services can be combined to enhance accessibility of visual content on the contrary with the audio content, where only one accessibility service can used.

Rule-based matching

Accessibility service belong to one of visual or hearing aid services. Visual services make the video/image accessible and hearing services make the audio accessible. The process of finding substitution is based on this categorization. Table 1 lists possible substitutions.

Table 2 Accessibility service substitutions

Accessibility service	Content	Substitution
Screen magnification	Video/image	None, always available
Face detection	Video/image	Audio subtitles + subtitles OR Subtitles OR Screen magnification
Audio Subtitles	Video/image	Subtitles + font size (assuming low vision)
Text detection	Video/image	Screen magnification
Character recognition	Video/image	Screen magnification
Subtitles	Audio	Sign language with the same language OR

		Sign language with a different language that the user understand OR Audio with different or similar language
Sign language	Audio	Face detection OR Subtitles
Sound detection	Audio	None
Audio language	Audio	Audio subtitles + subtitles OR Subtitles

The information shown in Table 2 are encoded in rules in the RBMM, which enable suggesting proper substitution for unavailable accessibility services.

On the other hand, the task of suggesting new accessibility features uses the user available information to identify whether the user is in need of visual or auditory accessibility features. The only available user information are the user preferences, which can be a good source for identifying these cases. For instance, a user with high audio volume level is a good indication that suggestions for hearing accessibility services can be made. The following table indicates some of these preferences.

Table 3 Indications of user preferences

User preferences	Comment
Magnification level	High magnification level is a strong indication that the user needs some visual accessibility services.
Font size	Large font size is a good indication that the user needs some visual accessibility services.
Audio level	Although users may increase the volume level for many reasons (high surrounding noise, far distance from audio source, hearing difficulties, etc) High volume level is a good indication that the user may need hearing accessibility services.
High contrast level	The contrast level (between the foreground and background colors) can be used to indicate the user visual requirements.
Face, text and sound detection	Enable one or more of these accessibility services can be a strong indication that the user need some visual accessibility help.
Sign language, Sound detection	Enable one or more of these accessibility services can be a strong indication that the user need some auditory accessibility help.

The rule-based matchmaking approach suggests content accessibility services based on the user preferences only. From a technical point of view, the rule-based matchmaker uses semantic web, which follows a graph data model that represents data in graph like structure. Following this model, data is represented as triple statements of the form *(subject predicate object)*, in relation to the graph

structure, statements' subjects and objects are the graph vertices and predicates are the edges or links. With the help of rules new triple statements are inferred from the existing ones, which corresponds to adding new edges or/and nodes to the graph.

The suggestion process starts by converting the user profile and MPD file content into triple statements. The inference rules used are rules that identify user needs and substitution services as depicted in Table 3 and Table 2. After the inference phase, the generated data model is queried for newly generated statements, which corresponds to suggested accessibility services.

Statistical matching

The statistical matchmaking approach corresponds to the collaborative filtering approach. It depends heavily on the available users' data, namely user's profiles and their activities.

For the statistical approach, first similar uses are found, which corresponds to find all users that have similar needs & preferences including their content accessibility services preferences. At this stage MPD file information is used to further filter out these similar users. For this purpose, the users distance from the content of MPD file is calculated and users are sorted based on their distances to MPD file available services. This corresponds to find the user with nearest preferences to those contained in MPD files. For calculating this distance only a subsection of the user's preferences are used, namely their content accessibility preferences. This filtering step is necessary to handle the case where the user preferences cannot be fulfilled and thus a good accessibility service substitution must be found. The result of the process is to find similar users that have the same content accessibility preferences as those of MPD file content. Figure 3: Collaborative content adaptation Figure 3 indicatively shows the step of this process.

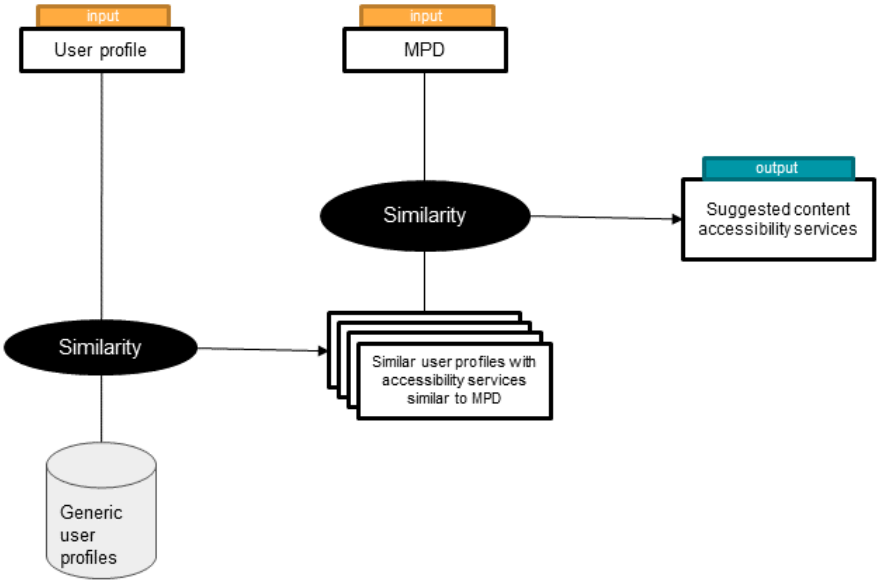

Figure 3: Collaborative content adaptation

Hybrid matching

The hybrid matchmaker combines the outcome of both matchmaking. The outcome of the rule-based matchmaker is suggested based only the user preferences and the outcome of the statistical matchmaker can be seen as what other similar users have to say in similar situations. Based on these outcome and the weight assigned to each approach the hybrid matchmaker decides what to

finally suggest to the user.

Content accessibility services are of Binary value {Enable, disable}, thus their hybrid outcomes are to decided which approach to follow. Of course that is in the case where both approaches have contradicted preferences. For example, the STMM and the RBMM have suggested for face magnification service {Enable, Disable}, with assigned weights of {60%, 40%} the hybrid matchmaker would follow the STMM approach, which is to enable magnification service. Table 2 shows the hybrid outcome of all combinations.

Table 4: Hybrid outcome

RBMM	STMM	HBMM
Enable	Enable	Both approaches agree on enabling the service. HBMM = Enable
Enable	Disable	The user preferences highly indicate the user needs for accessibility services, however, this is only followed when the RBMM weight is relatively high > 30% HBMM = $\begin{cases} Enable \ \ when\ RBMM\ weight > 30\% \\ Disable\ otherwise \end{cases}$
Disable	Enable	Other similar users suggest that the user is in need for accessibility services; however, his preference does not. HBMM = $\begin{cases} Enable \ \ when\ HBMM\ weight > 60\% \\ Disable\ otherwise \end{cases}$
Disable	Disable	No indication that the user is in need for the accessibility service. HBMM = Disable

The outcome of the HBMM achieves distinguishing the cases referred in the start of the section. A user that does not require any accessibility services and his/her profile lacks such preferences will end up belonging to the last case and the hybrid outcome would be to disable the service.

3. MPD FILE EDITOR

The Media Presentation Description file is an XML (eXtensible Markup Language) document that contains media information about the content and describes the structure of the streaming. This

structure is defined in ISO/IEC 23009-1 [3] and, although it was explained in detail in deliverable D4.1, the structure of this XML file is:

- Periods: the periods describe a part of the content with the start time and the content duration. Normally, there is one period in the MPD, but it possible to use different periods to divide the content into different episodes or to separate the main content and the advertising.

- Adaptation sets: the adaptation set contains a set of streams that belongs to the same service. For example, if there are several video resolutions for the same video, the adaptation set would contain all of them. Typically, there is one adaptation set for the video, one adaptation set for each audio and one adaptation set for each subtitle.

- Representations: as it is mentioned above, an adaptation set can have different representations, each representation, in the case of the video, with content encoded with different parameters: codec, resolution or bitrate.

- Segments: the representation is formed by an initialization segment and by one or more content data segments. Typically, the duration of these segments is between 2 or 10 seconds to allow a quick change of quality if the network conditions vary.

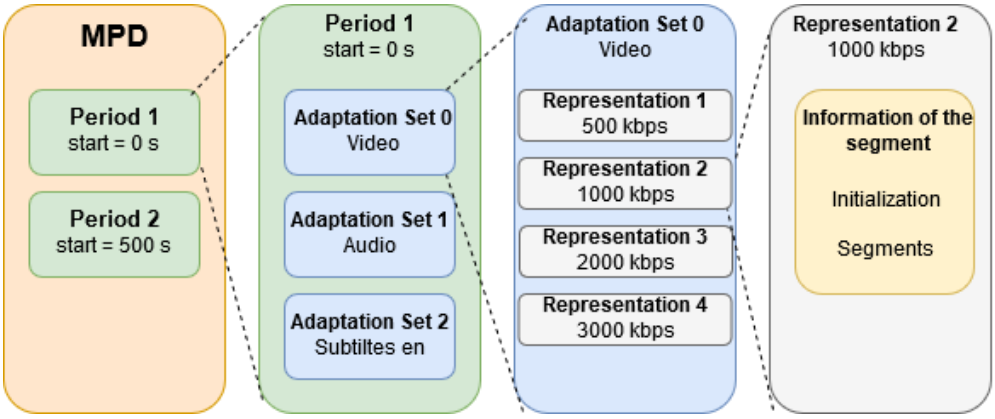

Figure 4: Structure of a general MPD

In deliverable D4.1 it was described the use of two tools for content preparation, content fragmentation in segments and the generation of the MPD. The tools were FFmpeg [4] and MP4Box [5]. The first is a free software tool that allows to encode and transcode videos and audios using different parameters. The second tool was used for the generation of the initialization dash files, the segments and the manifest file. It is a program available in the open source multimedia project called GPAC.

In EasyTV project, the broadcaster is in charge of providing the audiovisual content in MPEG DASH format. Videos, audios and subtitles need to be fragmented, so they will be included by the broadcaster in the manifest file.

However, other access services are contemplated in the project and it is not necessary to fragment it. Some of the access services are: face detection, text detection, sound detection and character recognition. How these services are created is outside of this deliverable, but all of these access services will generate a JSON (JavaScript Object Notation) file that includes all the necessary information for the client application that can provide the service to the end user.

To guarantee the correct operation of the DASH client, the additional information will be add outside the period tag and inside the MPD tag. Tests have been done in this way and without any problem.

The player used in the client application is the dash.js JavaScript library, a library that allows the use of the standard for the playback of the streaming [6]. Dash.js is an open source DASH player and is provided by the DASH Industry Forum (DASH IF), a forum created with the objective of favouring the use of this standard and to make devices compatible with the technology.

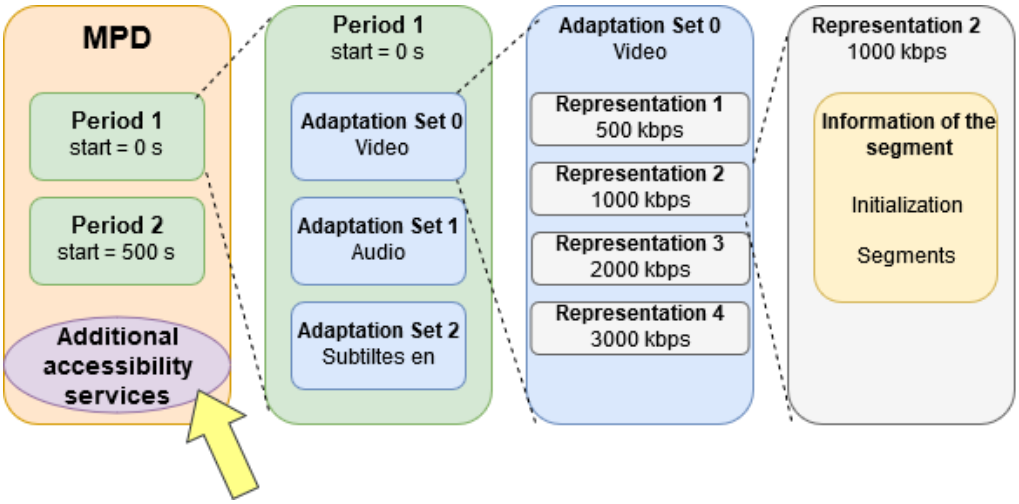

Figure 5: Structure of a modified MPD with access services

The additional information will be a tag that contains information about the availability of the access services. If an access service is available, the MPD will include a tag that identifies the service and the URL (Uniform Resource Locator) where the JSON file about this specific service is stored. This is possible thanks to the fact that MPD is a XML file, and new tags and information can be easily added.

Using specific tags for each access service, the module implemented in the client application can know the available services and download the JSON files to provide the service. The table shows the access services and the associated tag. In case there are new services, the new service will be associated with a new identification tag.

Table 5: Identification tag of each access service

Access service	Identification tag
Face detection	<face_detection>
Text detection	<text_detection>
Sound detection	<sound_detection>
Character recognition	<character_recognition>

For adding the tags and the URL of the JSON, a python script has been implemented. The structure of the script is shown in the following figure.

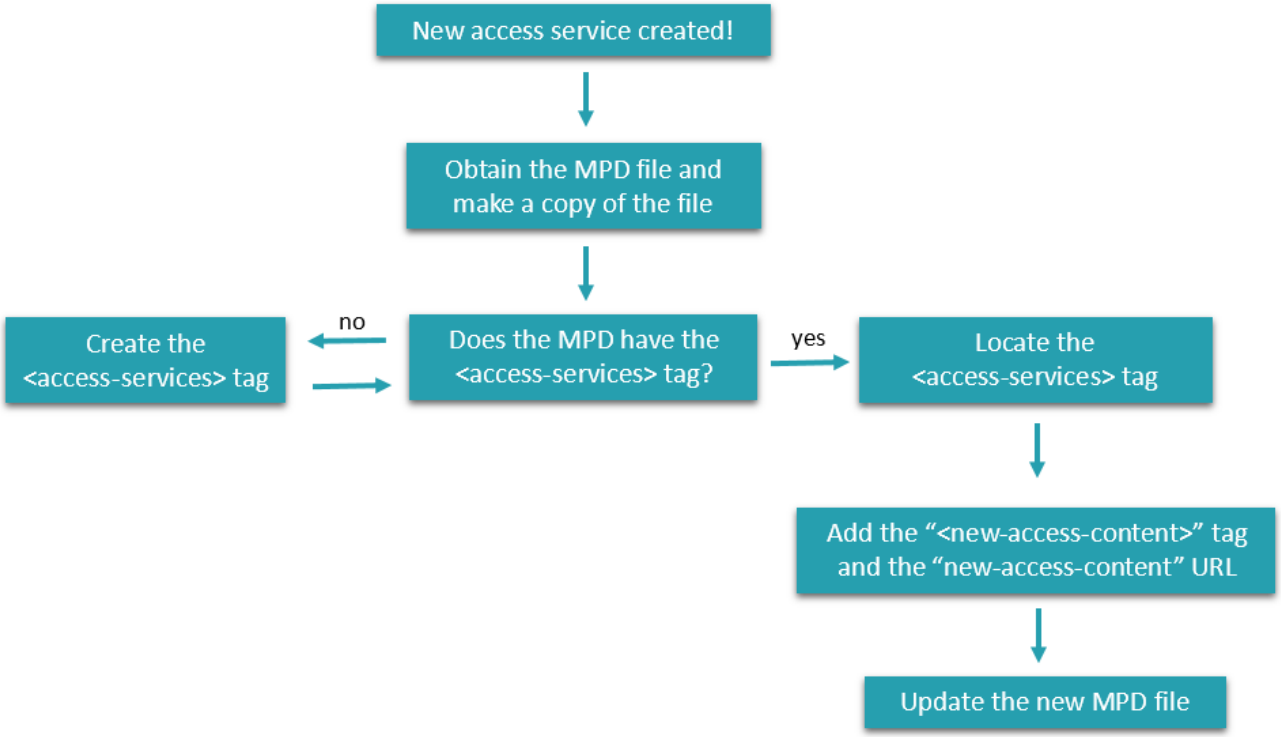

Figure 6: Structure of the MPD editor script

When a new access service is created in the EasyTV Cloud Platform, the MPD editor tool is executed. The script receives the location of the MPD, the type of service that has been generated and the URL where the JSON file is stored.

At the beginning of the edition, a copy of the MPD is done and stored with the date and time when the operation takes place. To have a backup of the MPD is important if the edition of the file fails.

When the backup is created, the implementation reads the MPD and tries to locate the access services tag. If the MPD does not have this tag (there is not available any access service), the tag is created, outside the period tag and inside the MPD tag. When the access service tag is located, inside this tag, the identification tag of the service and the URL of the JSON file is written.

Finally, all changes are saved and the application would access to this last modified MPD. For that, the name of the MPD does not change. If a new access service is created, the editor adds the information into the last MPD.

The following figure shows a modified manifest file. Two access services (face detection and text detection) were generated, and after the creation, the editor added the additional information. In the file, it is possible to see the new structure of the MPD and the information about these services.

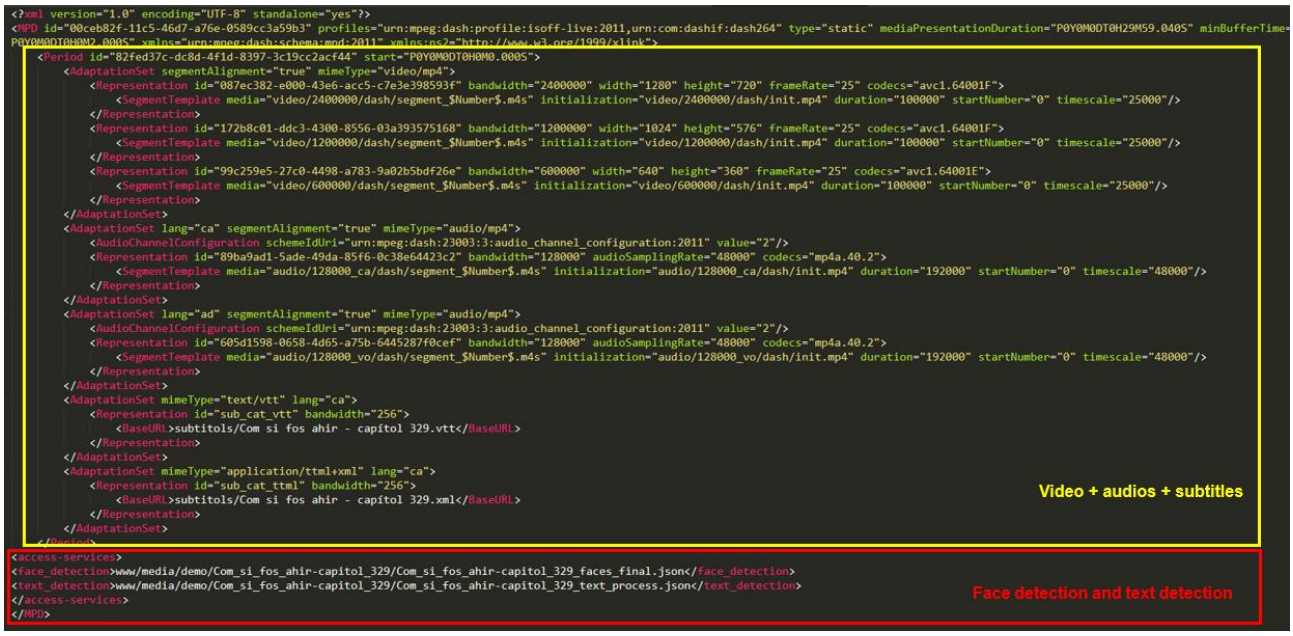

Figure 7: Example of a modified MPD with access services

In the example, this audiovisual content has a video with three different qualities (1280x720 at 2.4 Mbps, 1024x576 at 1.2 Mbps and 640x360 at 600 Kbps), two audios (catalan language and original version), and two subtitles (both in catalan, but with different format). For this content two access services are available: face detection and text detection.

4. MODULE IN CS APP TO CUSTOMIZE THE AUDIOVISUAL CONTENT

Once the access services are generated and the MPD is modified, a module in the application in charge of reading and interpreting the file is needed. This module is connected with the Hyper-Personalisation module, a module integrated in the EasyTV Cloud Platform and that allows the management of the user needs and preferences of the final users and is in charge of providing to the application, depending on the user profile, the necessary information to provide the access services by default.

The module for reading and interpreting the MPD, implemented in the client application, has to locate the access service tags, download the JSON files, interpret the information and provide the services.

The module makes use of:

- Creation of an XMLHttpRequest.
- XMLHttpRequest.responseXML to obtain the MPD.
- getElementByTagName (“access-service-tag”) to know if a service is available for a specific content.
- firstChild.nodeValue () to save the content of the tag, in this case, the URL of the JSON file.

When the application downloads the JSON file, it has to interpret the information to provide the service. The structure of the JSON files will be described for the four access services discussed in this deliverable. Although the structure is very similar, there is some small differences to take into account when the application shows the services.

Face detection service:

The JSON file collects, for each frame, if there is a scene change and the faces detected for that frame. For each of the faces appears the position of the face in the video, the size of the bounding box that contains the face, the name of the character, an estimation of the age and genre, and if the character is speaking or not. How to detect the faces, identify people, estimate the age and gender and know if someone is speaking, is outside the objectives of this deliverable.

Figure 8: Example of a JSON file for face detection

In this example, only it the information of one frame is shown. In this frame, there is a scene change (value 1) and only one face is detected: position in the video (818,120), bounding box size 191, the character is Marta, age 40, gender female (value 0) and is speaking (value 1).

All this information allows to show graphically the data in the application and perform, for example, an automatic magnification of the face detected since the position and the size of the face are known.

Text detection service:

The JSON file for this service collects the text detected and the first frame and the last frame where the text appears.

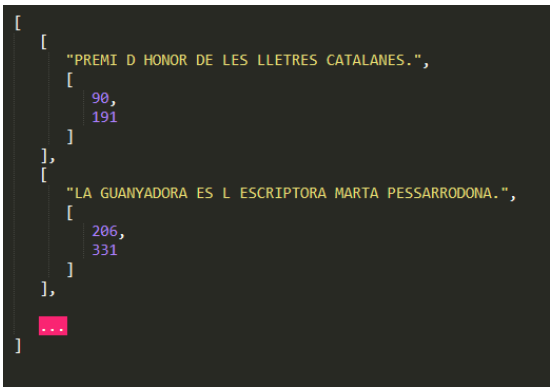

Figure 9: Example of a JSON file for text detection

In this example, two texts have been detected by the implemented algorithm. The first text appears between frame number 90 and frame number 191. The second text appears between frame number 206 and 331. Knowing the text, and the initial and end frame, the application can show the text with different font size, different font colour and different background colour, in order to customise the text.

Sound detection service:

The JSON of this service collects the type of sound detected in the video, and the initial time and end time, in seconds. The application can show graphically this information for helping people with hearing disabilities.

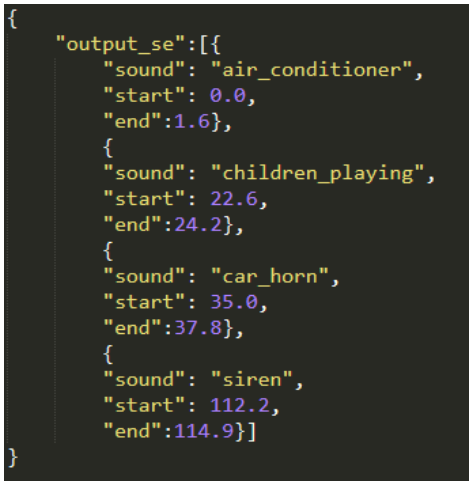

Figure 10: Example of a JSON file for sound detection

In this example, four different types of sounds have been detected: air conditioner, children playing, car horn and a siren. For each type of sound, the time when the sound starts and finishes is shown.

Character recognition service:

For this service, the JSON file retrieves the name of the character who is speaking, when the character starts and when finishes to speak.

```
{
    "output_sp":[{
        "speaker": "John",
        "start": 0.0,
        "end":2.4},
        {
        "speaker": "Marie",
        "start": 2.6,
        "end":5.2},
        {
        "speaker": "John",
        "start": 6.0,
        "end":10.4},
        {
        "speaker": "Cristopher",
        "start": 11.2,
        "end":14.4}]

}
```


Figure 11: Example of JSON file for character recognition

The idea is to complement this service with the face detection service mentioned above, with the aim of increasing the accuracy of the algorithms using both audio and video information. In this way, the application can show graphically the characters that appear in the scene, the age and genre, and who is speaking. Also it is possible to create an audio with this information, for helping people with visual disabilities.

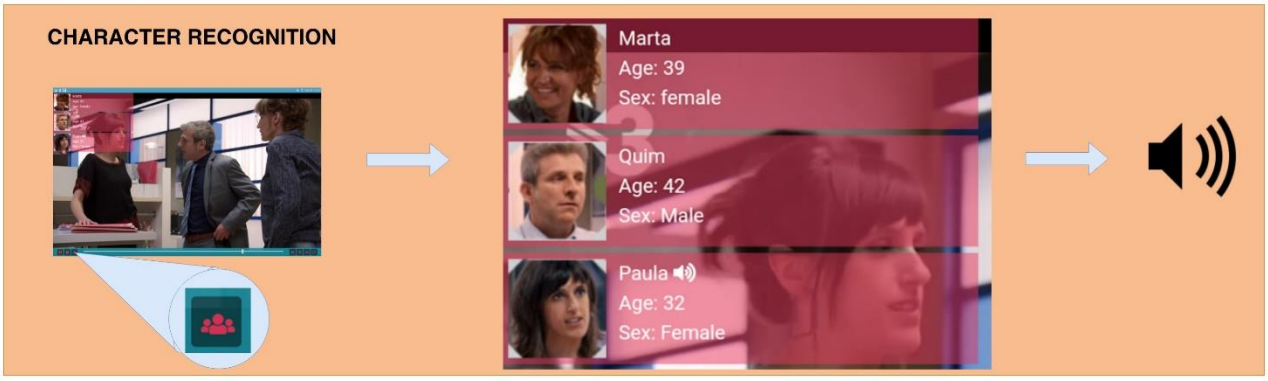

Figure 12: Automated character recognition service in the client application

In this example, in the scene, three characters appear: Marta 39 female, Quim 42 male, and Paula 32 female. In that moment, Paula is speaking.

5. INTEGRATION OF THE MPD FILE EDITOR IN THE EASYTV PLATFORM

This chapter describes the integration of the MPD editor in the EasyTV platform. As it is explained in chapter 3, this editor is a script responsible for adding specific information in the MPD file when a new access service is created. In that chapter, also are mentioned the four access services contemplated nowadays in the project and those that are added into the MPD when are generated.

The different modules for the four access services are integrated in the same server, within the EasyTV platform. On this server is also included the module responsible for modifying the MPD and a copy of the contents provided by the broadcaster to carry out the processing of the services. The following figure shows the structure of the server and the integration of the server into the platform.

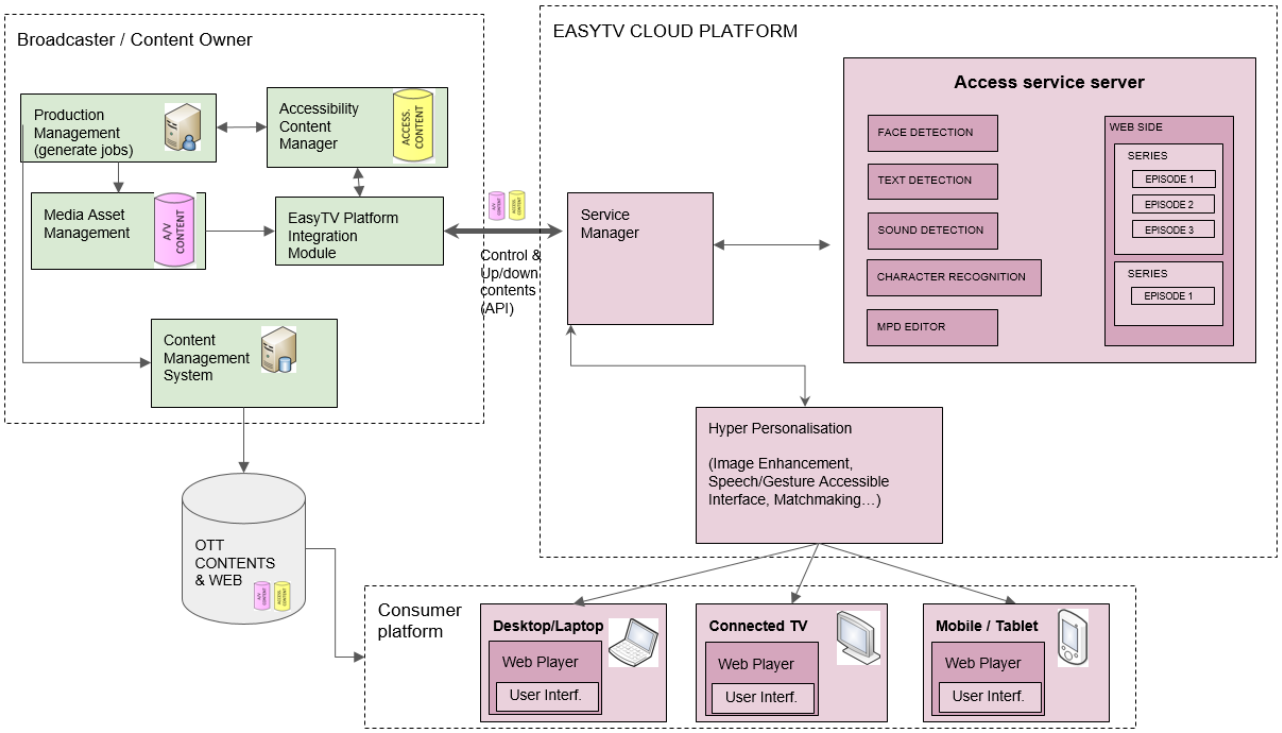

Figure 13: Integration of the MPD editor into the platform

The service manager is a gateway and orchestrator between the broadcaster premises and the EasyTV platform. The server manager controls tasks and actions related to the correct performance of the platform. In this particular case, when there is a new audiovisual content, the service manager can start the creation of the different access services. When each access service finishes and the JSON file is generated with the information of the service, the MPD editor is executed to include the access service in the manifest file.

For the generation of the services, the job is the same in all the four cases: start the service, process the algorithms of the service, generate the JSON file with the results and start the editing of the MPD file. The editing of the file is done at the end of the workflow to guarantee that the service is added when the process is finished and the JSON is generated.

To execute the job that generates a new access service for an audiovisual content, the service manager generates a POST request to the server. The structure of the request is:

http://IP-server:port/service_name/start/series_name/episode_name

The request consists of the IP address and server port, and the type of service to be generated, together with the name of the series and the episode of the series.

6. INTEGRATION OF THE HYPER-PERSONALIZATION MODULE WITH THE CS APP

On each user requests to access a new content, the client side has to interact with the hyper-personalization in order to get a personalized response. The following Figure 14 shows the interaction with the hyper-matchmaker.

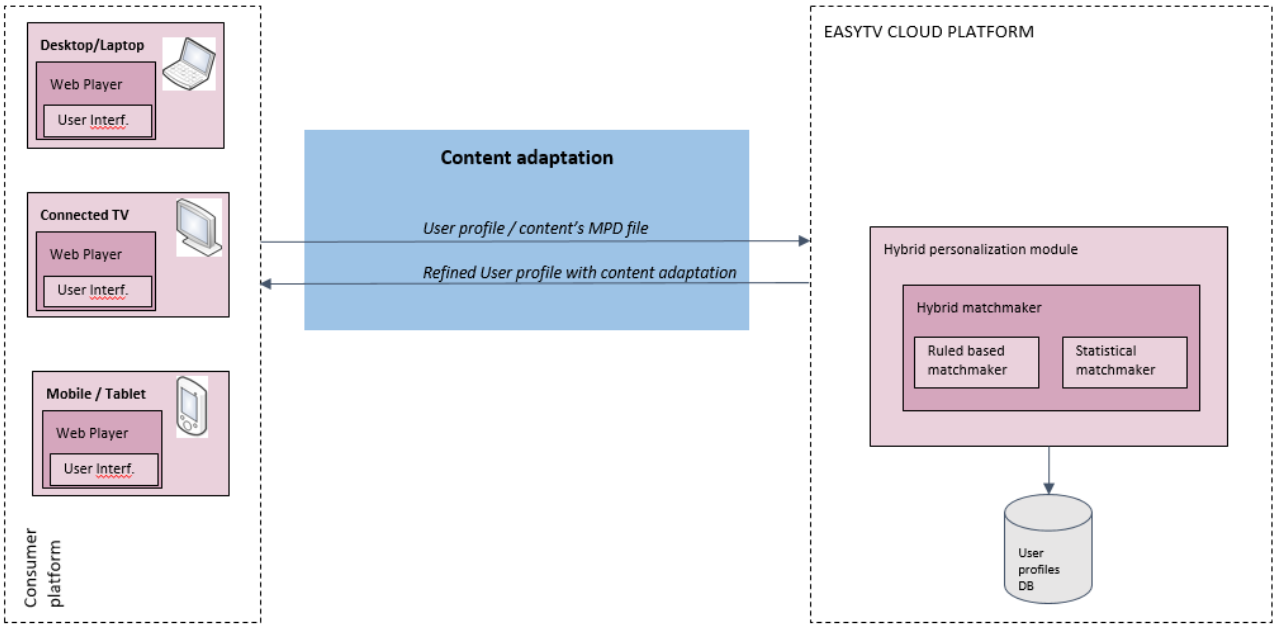

Figure 14: Personalization Calls

The client side, mainly the CSApp, sends the user profile and the content of MPD file. The outcome is actually a refined user profile, with a personalized set of preferences.

7. INDICATIVE USE CASE SCENARIO

7.1. Flat match case

A user with some visual difficulties has the profile shown in Table 4. The user has for content preferences, the face detection service. Assuming the content MPD file is the one shown in Figure 7, the user content can be fulfilled. More specifically, due to the lack of user preferences that indicates a perception difficulties, no RBMM rule would be triggered and no additional suggestions would be made. For the same reason, STMM results of similar users do not indicate any requirement for further accessibility suggestions. The hybrid outcome does not make any further accessibility suggestions. The CSApp sends the user profile and the MPD file content and gets in return the output shown in Table 7.

Table 6: Flat match user profile


```
{"user_preferences": {
        "default": {
                "preferences": {
                "http://registry.easytv.eu/common/content/audio/language": "CA",
                "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 12,
                "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/faceDetection": true,
                "http://registry.easytv.eu/application/cs/accessibility/textDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 0,
                "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false,
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
                "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
                "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
                "http://registry.easytv.eu/application/cs/cc/audioSubtitles": false,
                "http://registry.easytv.eu/application/cs/audio/audioDescription": false,
                "http://registry.easytv.eu/application/cs/audio/volume": 10
          }
        }
    }
}
```


Table 7: Flat match personalized user profile


```
{"user_preferences": {
        "default": {
                "preferences": {
                "http://registry.easytv.eu/common/content/audio/language": "CA",
                "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 23,
                "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/faceDetection": true,
                "http://registry.easytv.eu/application/cs/accessibility/textDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 0,
                "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false,
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
                "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
                "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
                "http://registry.easytv.eu/application/cs/cc/audioSubtitles": false,
                "http://registry.easytv.eu/application/cs/audio/audioDescription": true,
                "http://registry.easytv.eu/application/cs/audio/volume": 10
          }
        }
    }
}
```


7.2. Flat match with suggestions case

A user has preferences for audio and subtitle language of Catalan, and requested face magnification to be set on. Based on the MPD file of the content shown in Figure 6 the user preferences can be matched directly. The CSApp sends the user profile and MPD file content and gets in return the output of Table 9.

The outcome shows that the user content preferences has been fulfilled (enable text detection) and that additional accessibility services have been suggested. The user profile has strong indication that the user may need additional accessibility services. The user level of magnification for instance is high in combination with the lacking of face detection service triggers a rule in the RBMM to suggest face detection service. In addition, the STMM clustering method has found that similar users have preferences for colors (foreground and background colors) with higher level of contrast.

The output of the personalization is then is to suggest enabling the face detection in addition to higher contrast colors for foreground and background colors.

Table 8: Flat match second user profile

```
{"user_preferences": {
        "default": {
                "preferences": {
                "http://registry.easytv.eu/common/content/audio/language": "CA",
                "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
                "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 70,
                "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false,
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58,
                "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
                "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
                "http://registry.easytv.eu/application/cs/cc/audioSubtitles": false,
                "http://registry.easytv.eu/application/cs/audio/audioDescription": false,
                "http://registry.easytv.eu/application/cs/audio/track": "CA",
                "http://registry.easytv.eu/application/cs/audio/volume": 39
          }
        }
    }
}
```

Table 9: Flat match with personalized suggestions user profile

```
{
“recommended_suggestions”: {
                "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
                "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243"
},
"user_preferences": {
        "default": {
                "preferences": {
                "http://registry.easytv.eu/common/content/audio/language": "CA",
                "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
                "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 50,
                "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false,
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58,
                "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
                "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
                "http://registry.easytv.eu/application/cs/cc/audioSubtitles": false,
                "http://registry.easytv.eu/application/cs/audio/audioDescription": false,
                "http://registry.easytv.eu/application/cs/audio/track": "CA",
                "http://registry.easytv.eu/application/cs/audio/volume": 39
          }
        }
    }
}
```

7.3. Best substitution match case

Although most of the accessibility services offer different kind of information, some accessibility services can be considered a good substitution to others. A good substitution of the face detection service is the custom magnification service. Similarly, there are cases where combination of accessibility services could be a good substitution to others. For example, text detection in combination with audio subtitles can assist people with visual difficulties.

A user with the profile Table 10 has content accessibility preference for face detection and subtitles;

however, the user requests a content that lacks face detection service. Based on rules that define substitutions of accessibility services, the RBMM suggests enabling the magnification service. In addition, the statistical approach finds out that similar users with subtitles and face detection have enabled the audio subtitle service in similar cases. The user personalized output is shown in Table 11

Table 10: Best match user profile


```
{"user_preferences": {
        "default": {
                "preferences": {
                "http://registry.easytv.eu/common/content/audio/language": "CA",
                "http://registry.easytv.eu/common/display/screen/enhancement/font/type": "serif",
                "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 23,
                "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#b23b41",
                "http://registry.easytv.eu/common/display/screen/enhancement/background": "#18d4dc",
                "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/faceDetection": true,
                "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
                "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 0,
                "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false,
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58,
                "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
                "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
                "http://registry.easytv.eu/application/cs/cc/audioSubtitles": false,
                "http://registry.easytv.eu/application/cs/audio/audioDescription": false,
                "http://registry.easytv.eu/application/cs/audio/track": "CA",
                "http://registry.easytv.eu/application/cs/audio/volume": 70
          }
        }
    }
}
```


Table 11: Best match personalized user profile


```
{
“recommended_suggestions”: {
                "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 40
},
"user_preferences": {
        "default": {
                "preferences": {
                "http://registry.easytv.eu/common/content/audio/language": "CA",
                "http://registry.easytv.eu/common/display/screen/enhancement/font/type": "serif",
                "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 23,
                "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#b23b41",
                "http://registry.easytv.eu/common/display/screen/enhancement/background": "#18d4dc",
                "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/textDetection": false,
                "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 0,
                "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false,
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
                "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58,
                "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
                "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
                "http://registry.easytv.eu/application/cs/cc/audioSubtitles": true,
                "http://registry.easytv.eu/application/cs/audio/audioDescription": false,
                "http://registry.easytv.eu/application/cs/audio/track": "CA",
                "http://registry.easytv.eu/application/cs/audio/volume": 70
          }
        }
    }
}
```

8. INNOVATION POINTS IN EASYTV

In the EasyTV project there are two main innovation points related to the content adaptation: the modification of the MPD file structure for offering additional access services and the possibility of satisfying the needs and preferences of the final users thanks to a module in charge of fulfilling the preferences and suggesting new ones if it is possible.

Changing the structure of the MPD, it is possible to provide new access services created in the project and taking advantage of the use of the MPEG DASH technology. A general MPD offers the structure of videos encoded at different bitrates and resolutions, audios in different languages or encoded at several bitrates, and subtitles in different languages. This structure is defined by the standard. However, modifying the file, the project includes additional useful information to describe the availability of new access services.

This modification is possible thanks to the basic operation of the MPEG DASH technology. That is, in a simple streaming scenario between an HTTP server and a DASH client, the MPD file is sent at the beginning of the session so the application can easily read and interpret the file, and can play the audiovisual content and the access services that best satisfied the user's preferences. Moreover, this is possible because the MPD has an XML format, and tags can be added and the file keeps a valid format.

Providing all content information in the MPD, only one file is required for describing the availability of all the media components associated to a specific streaming. It can be concluded that a useful and efficient solution has been achieved to provide new services not contemplated in the MPEG DASH standard, but using this adaptive streaming solution.

The innovation part of personalizing the user needs and preferences is located in the new ways already known technologies are used. The hyper-personalization component handles three content adaptation cases: flat match, flat match with suggestion and best substitution, depending on the whether user needs can be match directly with available services or suggestions and substitution for these are be found.

The hyper-personalization uses three well known filtering approaches: collaborative approach, rule-based approach and hybrid approach. The rule-based approach is mainly based on the information of the user needs and MPD file content. The collaborative approach uses the user available data to find similar uses and extract from them suggestions. Lastly, the hybrid combine the output of both approaches.

User needs and accessibility services are classified into visual or auditory and for each accessibility service one or more accessibility services or settings are declared to be its substitution. In addition, user needs and preferences are used to find out what kind of help the user actually need. This information are declared as rules in the rule-based matchmaker and thus helps the process of suggesting new accessibility features and suggestions to the user.

The collaborative approach also uses the user information and MPD file content. Initially similar users are found, regarding all their preferences, and are then filtered using the available accessibility services (MPD file content). The end result, is to find out the most similar users that have chosen MPD file content for accessibility services. These remaining users are then used to extract suggestions.

Lastly, the hybrid approach combines the output of both approaches using weights. These weights are altered in correspondence with the user selected suggestions. In that way, the user feedback is integrated in the personalization process.

9. CONCLUSIONS

In this deliverable, along with deliverable D4.1, a detailed study of the MPEG DASH technology as adaptive streaming solution for the audiovisual content delivery to the final users has been explained.

Deliverable 4.1 was a general document to explain the concept of the adaptive streaming, the different approaches and, in particular, MPEG DASH for broadband content delivery. With this technology, tests had been done to deliver content to a multidevice scenario contemplated in the new HbbTV specification and in the workflow of the EasyTV project. The tests started with the preparation of the audiovisual content, the generation of the files needed for doing streaming using DASH, the integration of a DASH client into the client application developed in the project, and the possibility of changing and personalising the contents included in the MPD file.

However, deliverable D4.4, a revised version of the previous deliverable is focused on the developments done for the tasks related to the content adaptation using DASH streaming services, and the integration of these implementation into the EasyTV platform. There is a module in charge of modifying a general MPD manifest file, adding extra information corresponding to other additional access services. This is possible because the MPD is a XML file, and tags and URLs can be easily added without causing problems in the MPEG DASH operation. Once the MPD has been modified, the client application can read the modified MPD, know what access services are available, obtain the information of the services and provide the particular service to the final user. Therefore, it is necessary a module created in the client application and responsible for performing these actions mentioned above. Both modules are integrated into the EasyTV project, the first in the platform and the second in the application.

Moreover, in the EasyTV platform, the Hyper-Personalisation module is in charge of establishing the visual design of the application and the content that is shown by default to the users, depending the user profile, the needs and preferences. In this deliverable, how this module selects the content and the customisation of this content is explained. Also, the integration of the module into the platform and the communication between the modules.

With the developments done, it is possible to see how the use of MPEG DASH technology is very useful to provide access services and personalised content in the second screen devices in a multidevice scenario of HbbTV 2.0. The chosen technology , the modifications and implementations developed within the EasyTV project, allow both the advantages offered by the MPEG DASH as an adaptive streaming technique using HTTP protocol, and provide easily the mechanisms for people with visual or hearing disabilities (or elderly people) can access to audiovisual content with customisable services and other access services created in this project.

REFERENCES

[1] “Hybrid Broadcast Broadband TV”, ETSI Standard TS 102 796, 2018.

[2] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Systems 46 (2013) 109–132, 2013.

[3] “Information technology; Dynamic adaptive streaming over HTTP (DASH); Part 1: Media presentation description and segment formats”, ISO/IEC 23009-1:2014, May 2014.

[4] FFmpeg. A complete, cross-platform solution to record, convert and stream audio and video. [Online, https://ffmpeg.org, accessed August 2019]

[5] MP4Box. GPAC, Multimedia Open Source Project. [Online, https://gpac.wp.imt.fr, accessed August 2019]

[6] GitHub: Dash.js, A reference client implementation for the playback of MPEG DASH via JavaScript and compliant browsers. [Online, https://github.com › Dash-Industry-Forum › dash, accessed August 2019]