Deliverable

Project Acronym:	**IMAC**
Grant Agreement number:	761974
Project Title:	*Immersive Accessibility*

D3.5-Player.

Revision: v2.1

Authors: Mario Montagud (i2CAT)

Delivery date: M19 (01-07-19, M21)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 761974

Dissemination Level

P	Public	
C	Confidential, only for members of the consortium and the Commission Services	

Abstract: This deliverable describes the architectural aspects, components and features of the ImAc player. The different pieces of software that make up the player, together with the necessary steps to make it running, are also explained. Finally, the deliverable explains the different screens, User Interfaces (UIs), menus and interaction controls available in the player for the personalized presentation of contents.

This is the second and final iteration of D3.5, after a first version delivered in M09 focused on single screen scenarios and presentation of immersive contents, subtitles and sign language videos, which was used in the first pilot phase. This final version includes improvements based on the feedback from the first pilot phase (specified as new WP2 requirements) as well as features that were planned for the second year of the project. Some examples of the latter include a more responsive and attractive UI, more mature voice control features, and support for multi-screen scenarios and embedding features into external websites.

REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	25-06-18	Mario Montagud	I2CAT	First version: template, ToC, and first draft
0.2	27-06-18	Mario Montagud	I2CAT	Second version: revised ToC, and key modules and functionalities
0.3	2-07-18	Mario Montagud	I2CAT	Third version: full and revised modules and functionalities
0.9	3-10-18	Mario Montagud	I2CAT	Complete version: updated version including all features implemented for pilot 1
1.0	6-10-18	Mario Montagud	I2CAT	Submitted version addressing the review by Chris Hughes
2.0	1-6-19	Mario Montagud	I2CAT	Second iteration of the deliverable: revised ToC (with new sections), new features and improvements to the features documented in the first iteration
2.1	3-6-19	Mario Montagud	I2CAT	Final version addressing the review by Chris Hughes

EXECUTIVE SUMMARY

The ImAc project targets the specification and implementation of an **end-to-end platform**, comprised of different parts where production, edition, management, preparation, delivery and consumption of contents take place. A **key component of the end-to-end platform is the ImAc player**, as it is the interface through which end-users will consume the available immersive and accessibility contents in an interactive and personalized manner. The platform is developed within the umbrella of WP3, in closed collaboration with the development of the edition tools for accessibility content in WP4. The technological aspects and features provided by WP3/WP4 components are driven by the **requirements gathered in WP2**, and from the **feedback obtained from WP5 pilot actions** and **WP6 dissemination actions**.

This deliverable firstly provides an overview of the identified **requirements and design criteria that have mostly determined the development of the player, its components and its features**. Then, it describes the **integration of the ImAc player within the end-to-end platform**, providing more details about the architectural aspects and software components (layer and modules) that conform the player compared to the information provided in D3.1. The deliverable additionally explains **how to access the source code as well as how to install and configure the player**. Finally, the deliverable describes the different **screens, User Interfaces (UIs), menus and interaction modalities** to control the player, enabling a personalized presentation of immersive and accessibility contents.

Note that this is the **second and final iteration of D3.5**. The first version was foreseen in M09 (but submitted in M11), and it reported on the developments performed in the first year of the project. The player version reported in that first version of the deliverable included an initial version of the UI (see Annex II), and provided support for the personalized presentation of Subtitles for the Deaf and Hard-of-Hearing (SDH) and sign language videos together with 360º videos, on single devices. That player version was used in the first pilot phase of the project. This second version of D3.5 reports on a more complete and refined version of the player, based on the scheduled developments for the second year of the project, and on the feedback gathered from the conducted pilot (WP5) and dissemination (WP6) actions, all transformed into refined and/or new requirements in WP2. Examples of features scheduled implemented in the second year of the project include: support for Audio Subtitles (AST) and Audio Description (AD) with different presentation modes and personalization features; a more attractive and complete ImAc portal (landpage); support for Easy-to-Read subtitles; a more mature and modular voice control feature, with multi-language support; and support for multi-screen scenarios. Examples of features that have been improved based on the feedback obtained so far include: refined versions of the UIs; new presentation modes and personalization features for subtitles and sign language; refined versions of the guiding methods (arrows, radar); and more efficient and robust performance, based on re-factoring of the software components.

The most noteworthy features that have been improved will be highlighted along the document, with the goal of comparing the evolved version of the player to the earlier ones. In relation to this, Annex I lists all versions of the player that have been released during the project's lifetime. Finally, note that it is also expected to take into account the insights and feedback obtained from the second pilot phase to end up with a final version of the ImAc portal and player. However, it is expected they will not result in major changes to the player.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail	
Mario	Montagud	I2CAT	mario.montagud@i2cat.net	
Isaac	Fraile	I2CAT	Isaac.fraile@i2cat.net	
Einar	Meyerson	I2CAT	einar.meyerson@i2cat.net	
Themis	Karavellas	I2CAT	t.karavellas@i2cat.net	
María	Genís	I2CAT	maria.genis@i2cat.net	
Sergi	Fernández	I2CAT	sergi.fernandez@i2cat.net	
Chris	Hughes	University of Salford	C.J.Hughes@salford.ac.uk	
Peter	tho Pesch	IRT	thopesch@irt.de	

CONTENTS

TABLES OF FIGURES AND TABLES

LIST OF ACRONYMS

Acronym	Description
AA	Accessibility Audio
AAC	Advanced Audio Coding
ACM	Accessibility Content Manager
AD	Audio Description
API	Application Program Interface (API)
AST	Audio Subtitles
AWS	Amazon Web Services
DASH	Dynamic Adaptive Streaming over HTTP
DVB-CSS	Digital Video Broadcasting - Companion Screens and Streams
FOA	First Order Ambisonics
FoV	Field-of-View
HOA	Higher Order Ambisonics
HbbTV	Hybrid Broadcast Broadband TV
HMD	Head Mounted Display
IDES	Inter-Device Synchronization
IMSC	Internet Media Subtitles and Captions
ISP	Internet Service Provider
JS	Javascript
KPIs	Key Performance Indicators
MPD	Media Presentation Description
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MVC	Model View Controller
NTP	Network Time Protocol

OBA	Object-based Audio
QoE	Quality of Experience
QoS	Quality of Service
SDH	Subtitles for the Deaf and Hard-of-Hearing
SDK	Software Development Kit
SFTP	Secure File Transfer Protocol
SL	Sign Language
ST	Subtitles
TTML	Timed Text Mark-up Language
UDP	User Datagram Protocol
UI	User Interface
W3C	World Wide Web Consortium
WP	Work Package

1. INTRODUCTION

1.1. Purpose of this document

The final goal of WP3 is to define and implement a platform integrating different components of the production chain including Editors / Editing tools, Accessibility Content Managers (ACM), packaging and distribution components, and the presentation of traditional, immersive and accessibility content.

One of the key components of the end-to-end platform is the ImAc player, which is composed of pieces of software (e.g. clients, libraries, modules, algorithms…) and User Interface (UI) components to enable the adaptive presentation of the immersive and accessibility contents considered in the project, while enabling interaction and personalization features. Likewise, the ImAc player will support the presentation of contents on both single screen and synchronized multi-screen scenarios.

The purpose of this deliverable is to describe the architectural aspects, components and features of the ImAc player. The different pieces of software that make up the player, together with the necessary implementation steps, will be also explained. The deliverable also provides a user manual.

1.2. Scope of this document

This document provides a comprehensive overview of the ImAc player, its components and dependences, and how it is integrated within the end-to-end ImAc platform. It also explains how to install/configure it and to how to use it.

1.3. Status of this document

A first version of D3.5 was foreseen in M09. It reported on the player version developed at that stage of the project, providing support for a personalized presentation of subtitles and sign language, on single devices. That player version was used in pilot 1. This document is a revised and final version of D3.5, scheduled to be submitted in M19. In particular, it describes the evolved version of the player, based on: 1) the findings from pilot 1; 2) the new gathered requirements; and 3) the features that were initially targeted for pilot 2.

1.4. Relation with other ImAc activities

The development of the ImAc player, together with the development of the components of the Immersive Platform, is included in WP3, and is part of its task T3.5. It is driven by the (home + professional) user requirements established in WP2, but refinements and extensions are also included, based on the insights from the evaluations in WP5, for each one of the pilot phases considered in ImAc. Improvements can also be integrated based on the results from the integration tasks in WP3 and from feedback gathered in dissemination actions (WP6). This is illustrated in Figure 1.

In addition, the PERT diagram in Figure 2 illustrates with more details the relation between T3.5 (and thus D3.5) and the other ImAc activities. In particular, the modules, features and UIs of the player have been derived from the user requirements gathered in WP2, which were

reflected in D2.2 and then transformed into technical specifications in D2.3. Likewise, the player is being developed based on the Architecture Design tasks reflected in D3.1.

The evolved version of the player described in this deliverable takes into account the insights and results obtained from the pilot and dissemination actions conducted so far.

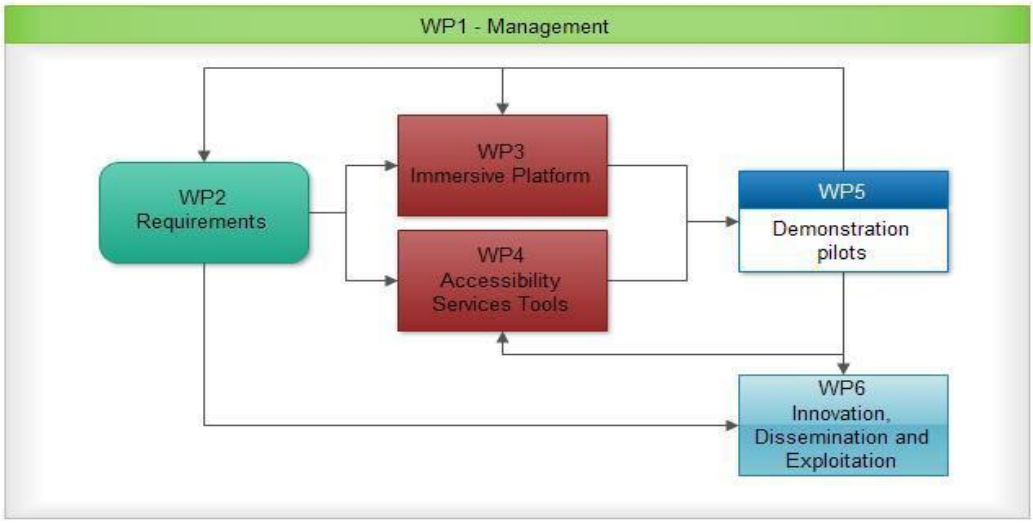


Figure 1- Relationships between Work Packages (WPs), and its cycles (iterations).

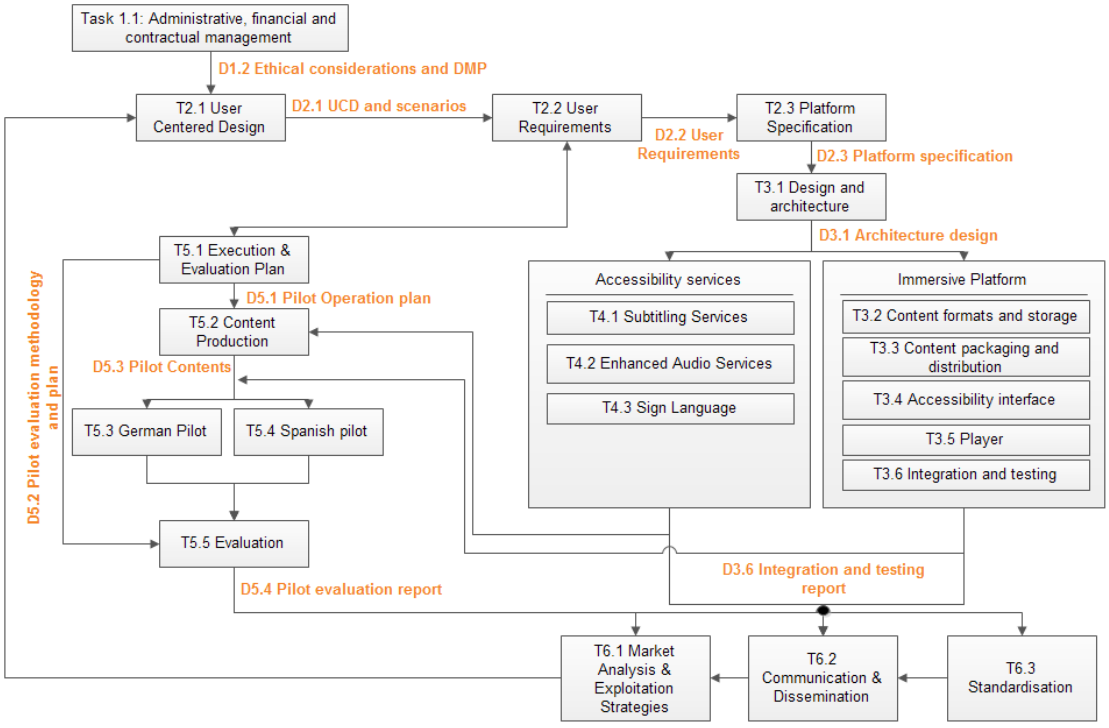


Figure 2- PERT Diagram illustrating the relationship between D3.5 and other ImAc activities

2. DESIGN CRITERIA.

2.1. Requirements from User-Centric Methodology

ImAc is following a user-centric methodology in which end-users, professionals and stakeholders are involved at every stage of the project through the organization of workshops, focus groups, tests, and the attendance to events. This allows gathering with high precision the required accessibility and personalization features, together with the scenarios of interest.

The requirements for the ImAc platform, its components - including the ImAc player, and the features and scenarios to be provided are collected and clustered under the umbrella of WP2, and detailed in D2.2. These requirements are transformed into technical specifications and features to be developed, as documented in D2.3. In particular, Table 1 lists the key (Home Users) requirements and system constraints, which have a significant bearing on the architecture and features of the player. These requirements are also assigned a priority of 'Must' or 'Should', and a code that identifies their version (see D2.2). Satisfying them is essential to developing a successful player for the ImAc project. At the moment of writing most of the requirements have been met / implemented, but the exception of: HUR.2.10.1 and HUR.2.59.0 (in progress, implementation for a subset of features, and waiting for the results of the pre-pilot tests); HUR.2.31.3 (implemented in a prototype but not adopted in the player, as initial tests have shown users do not receive well this feature); HUR.3.7.1 (activation mechanisms and contents for this have not been discussed and provided, respectively, so far); and HUR.3.11.0 and HUR.3.12.0 (tests would need to be conducted to determine the appropriate values for these thresholds).

Version	Title	Description	Prioritization
HUR.2.4.1	Access to Audio Description	The user can use a visual menu to access the service. It should be large, with a black background and white text for the highest possible contrast. Yellow will be used to highlight the choice of the user.	MUST
HUR.2.6.0	Multiplatform player for desktop, mobile phone (cardboard supported, gyroscope sensor based), TV, head mounted display	The user can start, pause, resume, forward or rewind the omnidirectional media with a graphical user interface.	MUST
HUR.2.8.0	Subtitles always on main Screen	Subtitles are always presented on the main screen, i.e. users do NOT want the subtitles to be delivered on an additional (companion) screen when	SHOULD

		accessing content	
HUR.2.9.0	Playback of audio description	The player enables the audio description to be synchronized with the main audio track	MUST
HUR.2.10.0	Player support for screen reader functionality	The player provides spoken feedback of all interface controls	MUST
HUR.2.10.1	Player support for screen reader functionality	The user can control the volume of the spoken feedback	SHOULD
HUR.2.13.0	User settings persistence	The player retains user preferences between users	SHOULD
HUR.2.16.0	Switch on/off signer	The user has the possibility to switch on/off the signer with a graphical user interface.	MUST
HUR.2.17.0	Selection of personalization options for signer	The user has the possibility to activate and deactivate different personalization options with a graphical user interface.	MUST
HUR.2.18.0	Accessibility interface signer - basic presentation mode	There is one basic presentation mode for the signer, which is always available for the user on any device. This mode presents it as follows: The signer video has a fixed position on the bottom right area of the field of view and the user decides what direction he/she wants to look.	MUST
HUR.2.19.0	Accessibility interface signer - position in viewing field	The user can select between a predefined set of different horizontal positions for the signer in the “basic presentation mode”.	MUST
HUR.2.21.0	Accessibility interface signer - speaker location indicator	The signer is always positioned in the user’s field of view according to personalization settings and an arrow under the signer indicates the position of the current speaker	SHOULD
HUR.2.21.2	Accessibility interface signer - speaker location indicator	The signer is always positioned in the user’s field of view according to personalization settings and radar interactively indicates the position of the speaker by positioning a dot inside	SHOULD

		the radar field in relation to the viewer's orientation	
HUR.2.24.0	Switch on/off subtitles	The user has the possibility to switch on/off subtitles with a graphical user interface.	MUST
HUR.2.25.0	Select subtitle tracks	The user has the possibility to select different subtitle tracks with a graphical user interface.	MUST
HUR.2.26.0	Selection of Personalization options for subtitles	The user has the possibility to activate and deactivate different personalization options with a graphical user interface.	MUST
HUR.2.27.0	Accessibility interface for subtitles - presentation mode	The subtitles are always visible in the user's field of view in the middle slightly below eye line, two-lined and each speaker has its own colour.	MUST
HUR.2.28.0	Accessibility interface for subtitles - position in viewing field	The user can select between a predefined set of positions in the viewing field (top, bottom) with a graphical user interface.	MUST
HUR.2.29.0	Accessibility interface for subtitles - size	The user can select between a predefined set of sizes (small, medium, and large) with a graphical user interface.	MUST
HUR.2.30.0	Accessibility interface for subtitles - background	The user can select between a predefined set of backgrounds (semi-transparent box, outline,-with a graphical user interface.	MUST
HUR.2.31.1	Accessibility interface subtitles --speaker location indicator	The subtitles are always positioned in the user's field of view according to personalization settings and an arrow left or right indicates the position of the current speaker.	SHOULD
HUR.2.31.2	Accessibility interface subtitles- speaker location indicator	The subtitles are always positioned in the user's field of view according to personalization settings and a-radar interactively indicates the position of the speaker by positioning a dot inside the radar field in relation to the viewer's orientation.	SHOULD

HUR.2.31.3	Accessibility interface subtitles- speaker location indicator	The subtitles are always positioned in the user's field of view according to personalization settings. When a speaker talks for the first time in a scene, the field of view is automatically changed towards that speaker by the video player "automatic speaker location indicator"). Afterwards the user can change the direction he/she wants to look.	SHOULD
HUR.2.36.0	Switch on/off audio description	The user has the possibility to switch on/off audio description	MUST
HUR.2.37.1	Selection of Personalization options for audio description	The user has the possibility to select different languages of the service and different audio description modes (see also requirements HUR.3.40.1, HUR 3.42.1 and HUR.3.8.0)	MUST
HUR.2.38.0	Interface adapted to user device	The user interface is adapted to the device used by the user	MUST
HUR.2.39.0	Change colour of subtitles	The user is able to adapt the colour of the subtitles to his/her needs.	SHOULD
HUR.3.40.1	Presentation mode for Audio Description	AD placed on the action (privilege of sound) - AD moves were the action is	SHOULD
HUR.3.42.1	Presentation mode for Audio Description	AD anchored to soundscape (1st person past tense) - the AD sitting next to you (left or right)	SHOULD
HUR.2.45.1	Accessibility interface for Audio Description - identify position	When there is an interesting secondary AD utterance the player places an audio beacon (beep) in that direction. Pressing the pause or "listen to beacon" button pauses the main audio and plays the AD for that object.	SHOULD
HUR.2.49.0	Accessibility interface for signer - comfort field of view	Users have the possibility to personalize the comfort field of view according to their preferences. Recommended are three levels (40%, 50%, 60%) in a 16:9 area according to the pre-pilot tests.	SHOULD
HUR.2.49.1	Accessibility interface for signer - comfort field of view	The user gets a visual feedback (dotted line) when selecting a new comfort field of view	SHOULD

HUR.2.50.0	Accessibility interface for subtitles - comfort field of view	Users have the possibility to personalize the comfort field of view according to their preferences. Recommended are three levels in a 16:9 area (50%, 60%, 70%) according to the pre-pilot tests.	SHOULD
HUR.2.50.1	Accessibility interface for subtitles - comfort field of view	The user gets a visual feedback (dotted line) when selecting a new comfort field of view	SHOULD
HUR.2.51.0	Switch on/off audio subtitling	The user has the possibility to switch on/off audio subtitling in parallel or separately to audio description.	MUST
HUR.2.52.0	Access to Audio Subtitling	The user can use a visual menu to access the service. It should be large, with a black background and white text for the highest possible contrast. Yellow will be used to highlight the choice of the user.	MUST
HUR.2.53.0	Selection of Personalization options for audio subtitling	The user has the possibility to select different languages of the service	MUST
HUR.2.54.0	Volume control of interface	The user can control the volume of the main content with a graphical user interface.	MUST
HUR.2.55.0	Volume control of audio description	The user can control the volume of the AD (independently of the main volume) with a graphical user interface.	MUST
HUR.2.56.0	Volume control of audio subtitling	The user can control the volume of the AST (independently of the main volume) with a graphical user interface.	MUST
HUR.2.57.0	Personalization options for interface	The user can activate/deactivate different personalization settings with a graphical user interface.	MUST
HUR.2.58.0	Language selection for interface	The user can select the language of the graphical user interface (categories, options, icons/abbreviations for accessibility services).	MUST
HUR.2.59.0	Voice commands	The user can control all interface settings with voice commands.	MUST
HUR.2.60.0	Accessibility interface signer - language	The user has the possibility to select different languages of the signer	MUST

HUR.2.61.0	Accessibility interface - speaker location indicator	When using the radar speaker location indicator, the user has the possibility to return to the main action of the video by clicking on a specific point on the radar icon.	SHOULD
HUR.2.62.0	Progress bar	The user can monitor the progress of the video via a progress bar and can jump to a specific point in time by clicking on that point in the progress bar	MUST
HUR.2.63.0	Access to interface	The user is informed how to open the interface via a banner display once a video starts playing	MUST
HUR.3.1.0	Sign Language Service	Sign Language Service must also be considered, appearing simultaneously to the person speaking.	MUST
HUR.3.2.0	Accessibility interface for subtitles - notices for dramaturgically-significant sounds	The user gets written translations of dramaturgically significant sounds, which are important for the plot.	MUST
HUR.3.4.0	Simplified Subtitles	Simplified subtitles may be useful for users with the need of easy-to-read texts.	SHOULD
HUR.3.5.0	Immersive Subtitles information	Subtitles are always visible somewhere on the screen, whether the object they represent is visible on the screen or not.	MUST
HUR.3.6.0	Playback of 3D audio	Audio is presented as “3D audio”. This may be e.g: via a suitable surround sound speaker system (preferably including height speakers) or via a binaural signal played back via headphones	MUST
HUR.3.7.1	Different voices for main and secondary actions	The main AD track keeps playing and as the user moves their head secondary AD tracks can be played depending on the direction the user is looking. These tracks would not overlap and should use different voices for the main and secondary audio tracks.	SHOULD
HUR.3.9.0	Sign Language Service	Body shift that is traditionally used by signers to indicate a change of speaker is avoided because it makes no sense for 360° content	MUST

HUR.3.11.0	Suppression of speaker location indicator - Subtitles	If a the duration of a subtitle frame is below a given amount of time (threshold to be specified) the speaker location indicator for that frame is suppressed automatically by the player	SHOULD
HUR.3.12.0	Suppression of speaker location indicator - Signer	If a the duration of a signer segment is below a given amount of time (threshold to be specified) the speaker location indicator for that segment is suppressed automatically by the player	SHOULD

Table 1 – Key Requirements gathered in WP2 (D2.2) for implementing a player meeting the demands and preferences of the Home Users

2.2. Use of web components

As discussed in D3.1, ImAc is relying on the use of web components for the development of the tools making up the end-to-end platform, and this also includes the player. The use of web components guarantees cross-device, cross-platform, cross-browser and cross-network support and eliminates the need for any installation and update at the client side. Thanks to the use of web technologies and components, anyone with a web browser and Internet connection can access and make use of the player.

More details about the advantages of web-based development can be found in D3.1 and in [1].

2.3. Integration in the End-to-End Platform and Workflow

The ImAc platform architecture is specified in D3.1. It determines the required components and modules to make up the platform, and the required interconnections and interactions between them. Likewise, D3.1 specifies the technologies and formats to be used in the ImAc platform. All these aspects and design criteria have also strongly influenced the development of the player, as it is a key component of the ImAc platform for the consumption of the contents created, edited, stored and delivered by other components of the platform.

Based on these design criteria, the architectural and technological aspects determined in D3.1, the next sections provide more details about the software architecture and components used for the development of the ImAc player.

3. COMPONENT DESCRIPTION.

3.1. General Description

The end-to-end ImAc platform is comprised of different parts / components where production, editing, management, preparation, delivery and consumption of contents take place. This is explained with further details in D2.3. Figure 3 provides a high-level overview of the logical layers or main parts of the ImAc platform, based on the current-day end-to-end broadcast workflows. In this figure, green boxes are used to identify components being developed within the ImAc project, and dashed transparent boxes are used to represent components that exist in typical broadcast platforms and workflows, commonly having interactions with the components developed in ImAc, but that are not developed in ImAc. This deliverable focuses on the Content Consumption components, by explaining the different pieces of software of the ImAc player, how to configure and use it, and how the different features are provided/enabled.

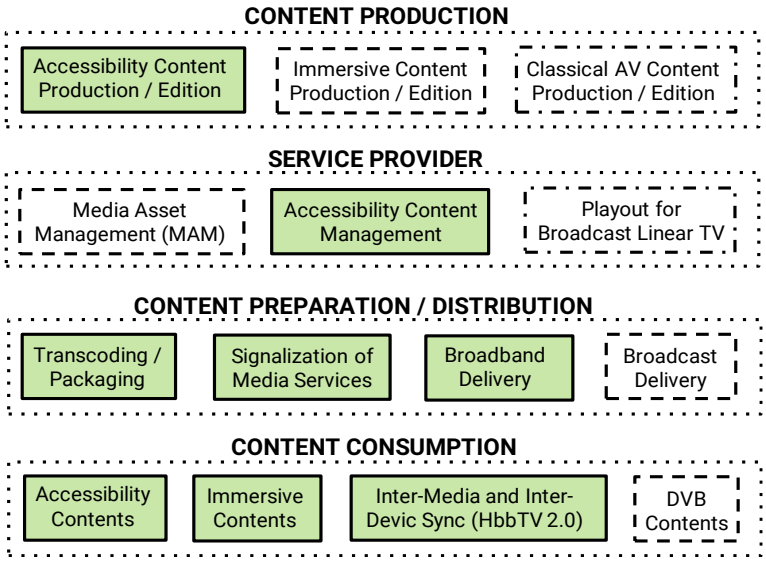


Figure 3- Main layers or parts of the end-to-end ImAc system

In relation to this, Figure 4 illustrates the logical architecture of the end-to-end ImAc platform, and how its different modules/components interact between themselves and with other components. In particular, the player can be typically embedded within the web services of the broadcaster or service provider. It is also able to automatically adapt to the targeted features and/or available capabilities of different types of consumption devices being used, such as Connected TVs, PCs, smartphones, tablets or Head Mounted Displays (HMDs), regardless of their platforms. This implies that the player can be integrated within these different devices, for a variety of platforms and Operating Systems. The integration of the player within the Hybrid Broadcast Broadband TV (HbbTV) services and apps of the broadcaster is also supported, so the traditional TV programmes can be enriched with accessible immersive content presented on companion devices. In particular, and as shown in the figure, the connected TVs will play out traditional TV contents (plus optionally accessibility contents) and

the companion devices will play out the immersive contents (i.e. 360º video and spatial audio) in combination with accessibility contents. Synchronization solutions to time align the presentation of the contents on all the involved devices are also provided, in order to enable more immersive, personalized and engaging multi-screen experiences.

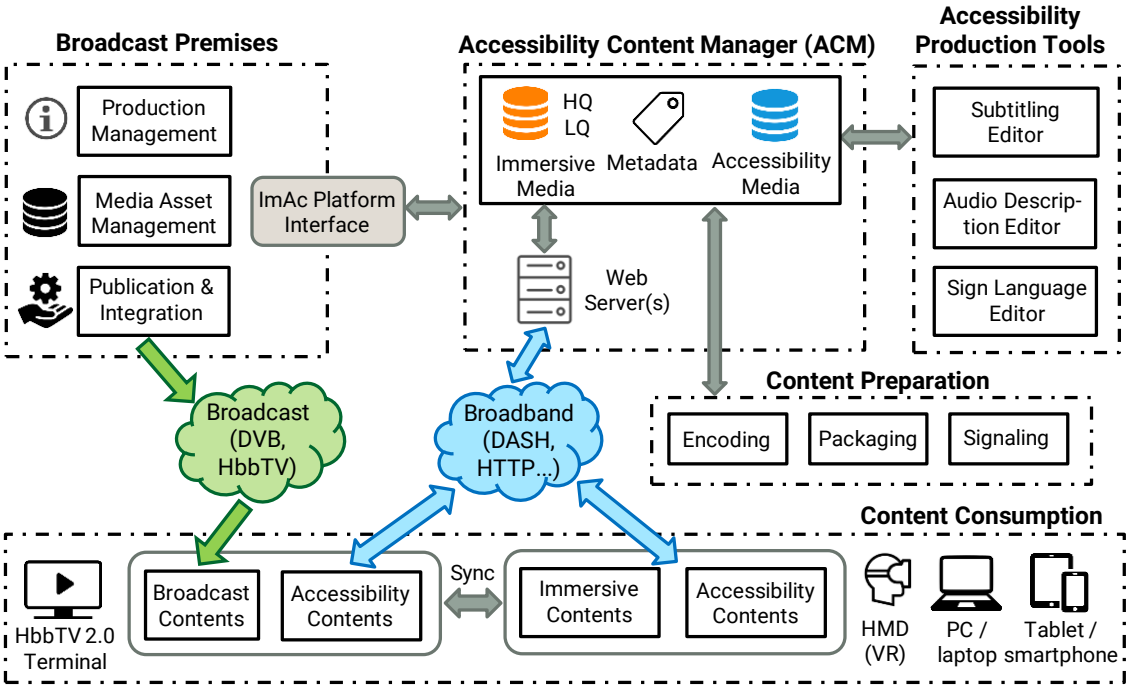


Figure 4- Architecture and components of the end-to-end ImAc system

Figure 5 shows the building blocks of the ImAc platform that are responsible of creating, preparing and storing the immersive and accessibility contents, together with their metadata for their delivery to the player. Concretely, the set of steps from media ingest & authoring to consumption can be summarized as follows:

1. The content provider uploads a high quality 360º video (and optionally an edited cover / thumbnail) to the ACM.
    - A low quality version of the video is automatically generated, to be used for the creation of accessibility content with the editors.
    - Accessibility contents can be also imported through the ACM.
2. The creation / editing of accessibility content is requested by the broadcaster (Accessibility Department) through the ACM.
3. The professional editors receive a notification (e.g. via e-mail). Once the job is completed, the created / edited accessibility contents and metadata generated by the Editors are uploaded to the ACM and linked to the uploaded 360º video (assets).
4. The broadcaster (Quality Assurance Department) receives a notification and verifies the files.
    - If changes are required, a message is sent to the Editors.
    - If the files are correct, publication of the content is triggered.
5. Content is prepared for distribution. This includes the following steps:

- Multi-quality encoding and packaging (segmentation).
- Signalling (availability of contents and their relationships).
- Storage on web servers.

6. Contents available for consumption.
    - Selection via the ImAc portal or the broadcaster website.
    - Notification while watching related TV programme (HbbTV).

More details about the architecture, the individual components and the interactions between them, as well as about the workflow, are provided in D3.1. Likewise, the steps from 1 to 4 are explained with further details in D3.2, the step 5 in D3.3, and this deliverable focuses more on step 6.

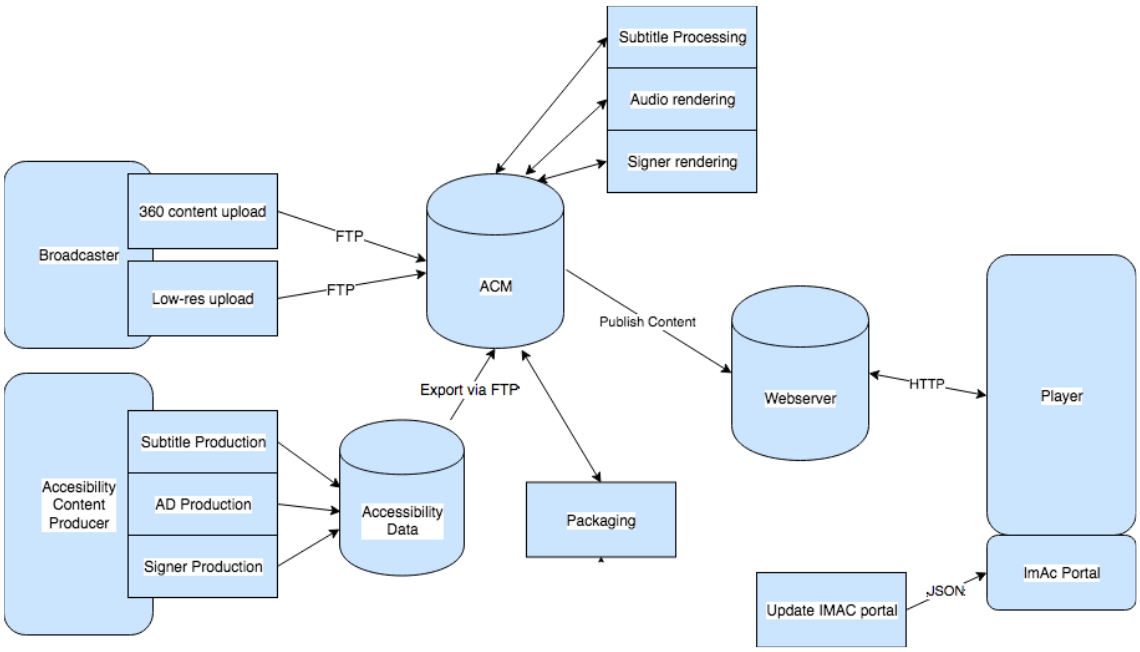


Figure 5- Workflow for Creation and Preparation of Contents and Metadata for the ImAc player

3.2. Media Technologies and Formats

This sub-section provides an overview of the media technologies and formats considered in the project, as they need to be supported and processed by the player.

Regarding media delivery and signalling solutions, Dynamic Adaptive Streaming over HTTP (DASH) and HbbTV are considered. On the one hand, DASH is used to deliver the immersive and accessibility content and to signalize their availability and features via standard-compliant extensions to the Media Presentation Description (MPD), detailed in Annex II. On the other hand, the features of the latest HbbTV release are used to support hybrid broadcast broadband multi-screen scenarios. In such cases, DASH is used for the delivery of the immersive and accessibility contents.

Regarding content codecs and formats, the following solutions are considered:

- 360º Video: In terms of codecs, H.264 with MP4 encapsulation is mainly used. H.265 and AV1 can be also considered, depending on technical and standardization advances during the project's lifetime. In terms of projection formats, both Equirectangular and CubeMap projections are supported. Indeed, ImAc is proposing novel configurations of CubeMap projection formats to optimize resources consumption (processing and bandwidth) while providing a satisfactory Quality of Experience (QoE) [2].
- Sign Language (SL) Video: Similarly as for 360º video, but using (2D) traditional video.
- Audio, including Audio Description (AD) and Audio Subtitles (AST). Generally, Advanced Audio Coding (AAC) is used for the audio encoding. Other codecs or extensions (e.g. HE-AAC/MPEG Surround, MPEG 2D Audio, Dolby AC4, DTS-X) could be supported, depending on the availability of Software Development Kits (SDKs) or licenses, as these codecs may be proprietary or patented. The audio content can be provided in different formats in order to support a wide range of devices and spatial audio modes:
    - Stereo.
    - Ambisonics. This is a client-side rendering system, which generates binaural audio for headphones. First Order Ambisonics (FOA) is easy to achieve but does not provide a fine-grained immersion. Higher Order Ambisonics (HOA) results in a good spatial resolution, but has higher costs in hardware requirements on client side. The player includes libraries to process Ambisonics audio.

  Initially, it was also considered to deliver Object-based Audio (OBA) up to the end user device. This would allow for high flexibility regarding the audio scene as well as the playback system, because the audio is rendered on the user device. However, relatively high processing power is needed to perform the client-side rendering. Additionally, the format for OBA delivery has not been established yet. Thus, it was decided to rule out this option for this project
- Subtitles (ST). Internet Media Subtitles and Captions (IMSC) file format has been adopted. It is a profile of Timed Text Mark-up Language (TTML), standardised by World Wide Web Consortium (W3C). The rationale for having chosen IMSC is provided in D3.1. In ImAc, standard-compliant extensions to IMSC are being specified, and proposed for standardization within the umbrella of W3C.

Figure 6 provides an overview of the content packaging and distribution module, with the mentioned media technologies and formats that are supported and processed by the player. Similarly, Figure 7 illustrates the delivery technologies and content formats used to provide the player with all considered media assets.

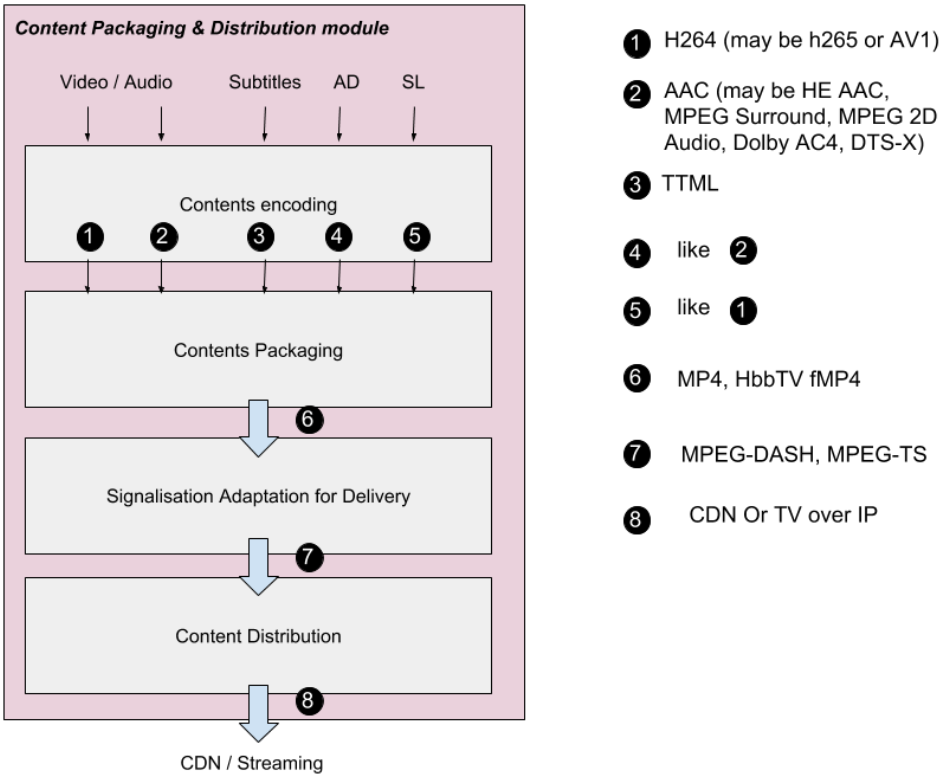


Figure 6- Overview of the Content packager and distribution module

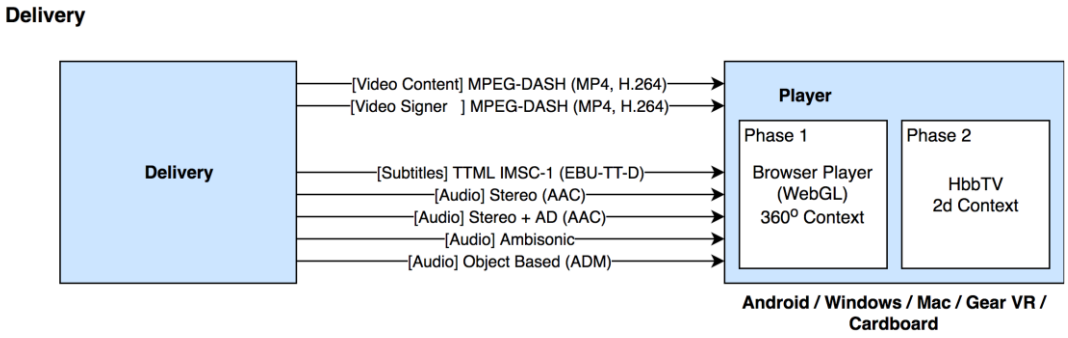


Figure 7- Delivery technologies and content formats for each considered media asset

More concrete details about the content formats, resolutions and versions for each media asset, as well as about the **metadata models** that have been proposed to signalize and describe the available contents, are provided in Annex II.

3.3. Player Architecture

The ImAc player is composed of a set of components to provide all targeted features envisioned in the project, including the list of requirements gathered in WP2 (detailed in D2.2 and summarized in Section 2.1).

Figure 8 illustrates an abstraction of the main layers, modules and libraries that make up the ImAc player, together with the relationships and interactions between them. All these components are mainly targeted at enabling the presentation of contents, the different interaction features and modalities, dynamically setting the available personalization options, and assisting the users in these tasks.

The names of the involved JS libraries implementing the functionalities of these components are also indicated. By using these libraries, the instances of the classes implementing each of the ImAc player's requirements are created. In the figure, AV stands for Audio+Video, V for Video, A for Audio, AA for Accessibility Audio, including Audio Subtitles (AST) and Audio Description (AD), SL for Sign Language, ST for Subtitles, and DVB-CSS for Digital Video Broadcasting - Companion Screens and Streams, which is the protocol in charge of providing the Inter-Device Synchronization (IDES) capabilities.

Note that the names of the libraries indicated in the libraries are meant to be intuitive and some of them may not perfectly match with the actual ones. Also not that not all the libraries and modules that make up the player have been described in the figure, but the key ones. The source code of the player, together with the necessary documentation, are provided in the project repository and have been also publicly shared on Github as open-source (see Section 6).

3.3.1. Presentation of Contents

Three main layers are in charge of the presentation of the contents in the ImAc player. These include:

- The *Immersive Layer*: it is responsible for the presentation of both traditional and immersive audiovisual formats. For immersive media, it includes 360º videos (in both Equirectangular and Cubemap Projection formats) and 3D spatial audio (Ambisonics).
- The *Accessibility Layer*: it is responsible for the presentation of accessibility contents considered in the project, namely: ST/SDH, AST, AD and SL.
- The *Assistive Layer*: it includes relevant features to assist the users for a more effective usage of the player. Some examples are: voice control (recognition and feedback, augmentation / zooming capabilities, and media processing techniques to improve the interpretation of contents).

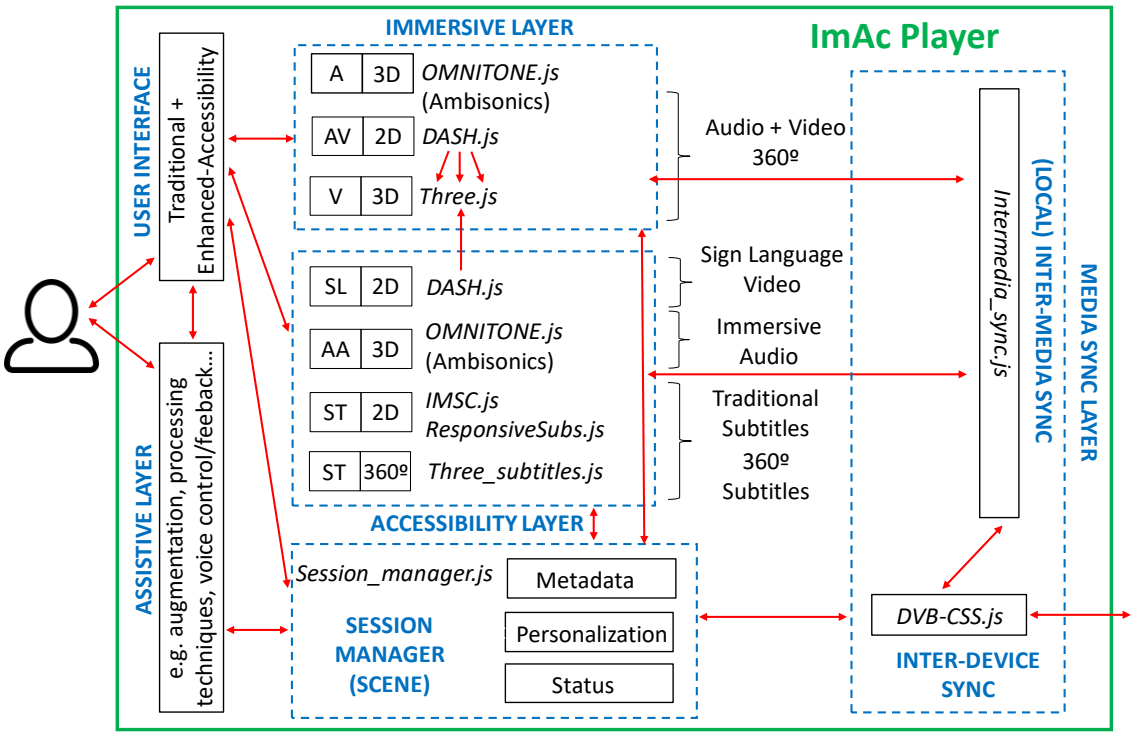


Figure 8- Abstraction of the Main layers, modules and libraries that make up the ImAc player

Likewise, the *Media Sync Layer* is in charge of ensuring a synchronized consumption of contents, both within each device (i.e. local inter-media synchronization) and across devices in a multi-screen scenario (i.e. inter-device synchronization).

In addition, two main modules of the ImAc player can be highlighted:

- The *Session Manager*: it is the module responsible for interpreting and selecting the list of available assets from the content provider, keeping an updated status about the contents being presented and the active devices in multi-screen scenarios. It also keeps track of the available personalization options and of the settings currently selected.
- The *User Interface (UI)*: it is the module through which users enable the presentation of contents, interact with the player and set the available personalization features. The design and development cycle of the UI for the ImAc portal and player are detailed in D3.4, and will also reviewed in Section 7 when describing how to use the player. Interestingly, each selected content from the portal has an associated URL, which also allows embedding the player with specific contents in a third-party website and/or in an HbbTV app.

The interactions between these modules and layers can be also observed in Figure 8. Next, the main libraries, tools and dependences used to enable the synchronized and personalized presentation of contents in the ImAc player are briefly introduced, and listed in Table 2.

Subtitles	2D	IMSC_360.js	Modified existing library
	Responsive subtitles	ResponsiveSubs.js	Newly developed library
	360º / 3D	three_subtitles.js	Newly developed library
Video	2D	DASH.js	Modified existing library
	360º / 3D	Three.js (WebGL)	Existing library
Audio	360º / 3D	Omnitone.js (Ambisonic)	Existing library
Scene / Session Management		SESSION_MANAGER.js	Newly developed library
(Local) Inter Media Sync		intermedia_sync.js	Newly developed library
Inter Device Sync		DVB-CSS.js	Newly developed library

Table 2 -Main Libraries and Components for Presentation of Contents in the ImAc player

3.3.1.1. IMSC_360.js

imscJS is a JavaScript library for rendering IMSC Text and Image Profile documents in HTML5. It is available here: https://github.com/sandflow/imscJS. This library has been extended to accommodate the relevant metadata in ImAc, such as the angle information for each subtitle frame and the presentation mode for subtitles.

3.3.1.2. ResponsiveSubs.js

ResponsiveSubs.js is the developed library to provide the responsive subtitles features considered in the project, which enable the re-blocking of subtitles by adhering to the number of characters that can fit into the display container at the chosen font size. More information about this feature and library is provided in D3.1.

3.3.1.3. three_subtitles.js

three_subtitles.js is a library for dynamically presenting subtitles for 360º videos in a 3D scene. It allows setting the proper position and depth, and/or the proper presentation mode for the subtitles (e.g. including positioning icons or auto-adjusting the field of view based on the position of the speaker/s). This library inter-operates with IMSC_360.js, interpreting the metadata for the proper presentation of subtitles in 360º / 3D environments.

3.3.1.4. DASH.js

DASH.js is a reference client implementation for the playback of DASH contents via JavaScript and compliant browsers. It is used in the player for the presentation of both traditional and immersive media formats (i.e., sign language and 360º videos, respectively). It is available here: https://github.com/Dash-Industry-Forum/dash.js/wiki.

3.3.1.5. Omnitone.js

Omnitone.js is a robust implementation of Ambisonic decoding and binaural rendering written in Web Audio Application Program Interface (API). Its rendering process is powered by the fast

native features from Web Audio API (GainNode and Convolver), ensuring an optimum performance. It is used in the player for the presentation of Ambisonic audio formats, both for higher immersion and better accessibility. It is available here: https://github.com/GoogleChrome/omnitone

3.3.1.6. Three.js

Three.js is a cross-browser JavaScript library and API used to create 3D scenes, including animated computer graphics, in web browsers. It is used in the player to compose the 3D environment, combining the 360º video, all visual accessibility contents and presentation modes and the UIs. It is available here: https://threejs.org/

3.3.1.7. Intermedia_sync.js

This is a newly developed library to guarantee a synchronized presentation of all considered immersive and accessible contents within each single device.

3.3.1.8. DVBCSS.js

This is a newly developed library with an implementation of the DVB-CSS protocol, adopted by the HbbTV standard, to provide IDES between a main TV and companion devices. In ImAc, this library is used to synchronize the playback between the involved devices.

3.3.1.9. SESSION_MANAGER.js

This is a newly developed library that implements all the required functionalities for Session Management. It stores relevant information about the session, such as the metadata obtained from the content provider (e.g., describing the available contents, their relationships, etc.), the contents being currently presented, the available personalization options together with the current settings, as well as the status of the session (e.g., elapsed time, duration, other active devices...).

3.3.1.10. Other libraries

Other third-party libraries have been used for the development of the player. Some examples are:

- jquery-3.2.1: HTML document traversal and manipulation, event handling, etc.
- socket.io: interaction channel between clients and servers.
- ntpsync-0.2.3: synchronization with an external Network Time Protocol (NTP) server to provide synchronized playback of contents.
- express-4.16.4: Node.js module useful for many purposes, such as routing, performance optimization, redirection, caching, etc.

3.3.2. Logical and Process Views

Figure 9 shows a high-level overview of the **logical view** for the player. It illustrates the interactions between the user and the player, as well as how the presentation of the immersive and accessibility contents can be dynamically activated/deactivated and adapted

according to tracking functions (e.g. users' movements, interactions…) and to the settings of the available personalization features.

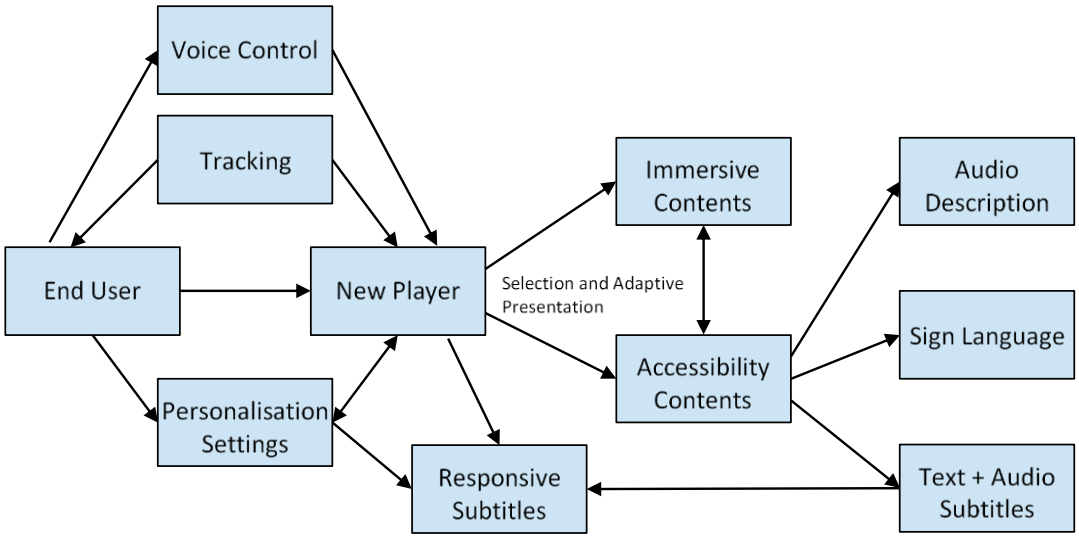


Figure 9- Logical View for the ImAc player

The ImAc player supports different interaction modes. When using a PC for the consumption of videos, interaction with the player can be done via the mouse or keyboard. When using a tablet or smartphone, interaction with the player can be done via the touch screen, although it is also possible to navigate around the 360º area by using the gyroscope sensor. When using a VR-enabled device, the head movement sensors can be used to navigate around the 360º area and to move the cursor. The controllers of the VR devices can also be used to move the cursor, while their touchpad and buttons can be used to select and/or navigate between menus of the UI. Finally, voice control is also available even for those devices not having a built-in or connected microphone via a jack, but that e.g. can communicate with a remote Amazon Echo (or Google Home) dot. This is explained in Section 5.

Next, the involved processes and modules to interact with the player and their menus, and to enable to presentation of contents, are described with more details.

When a 360º video is selected for its consumption, an ApplicationManager module starts to construct a set of elements (sphere for 360º video, camera, audio, menus...) via their associated modules, inside the 3D scene, by making using of the three.js library.

During the initialization of the different elements in the scene, the menu is created and hidden for resource consumption optimization, and the player shows and hides the different elements based on the user's interactions, trying to optimize the resources consumption.

The menu element creation starts in the MenuManager module, which will be influenced depending on whether the user has chosen the consumption of contents in VR mode (menu is attached to the scene) or in tablet mode (menu is attached to the camera). The next step consists of creating the structures for all the different menu views and initializing them. A

Model View Controller (MVC) architecture pattern is used for the menu creation. The menu is divided into 3 groups:

- **MainMenu**: The main menu is composed of the reproduction (play, pause and seek), volume (plus, minus and level), settings (settings, preview and menu type) and video progress bar elements. Although each group has its own view, each of these menu elements are controlled by a unique controller and model.
- **SettingsOptionMenu**: This group is in charge of the settings sub-menu. The internal structure of the different menu levels is described in *SettingsMenuStructure*. This file includes all the necessary data to be injected in the MVC model.
- **Preview**: The preview MVC group creates the previsualization of the access services and its personalized configuration. By enabling this option, the user has the opportunity to pre-visualize the selected settings by hiding the menu and auto-pausing the video in order to check if the selected settings are the ones desired.

The MVC flow starts with the controller been initialized, when the data to be shown in the view is loaded into the model and the view is updated with the new data. This new data contains all the functionalities for the different elements shown in each menu view. This flow is repeated every time the user navigates through the different views of the menu. Both menus (Traditional and Enhanced Accessibility – which is basically an enlarged version of the Traditional menu in the evolved version of the player) use the same views. The transition between them is accomplished by just re-scaling the visual elements. This can be easily achieved because the elements' size depends on the menu width. The whole menu view structure is described below when introducing the *ViewStructureMenuManager* module.

The developed MVC architecture in the ImAc player is represented in Figure 10, and the main involved modules are briefly described below:

- **AplicationManager**: This module initializes all the elements in the scene (camera view, video mesh, audio, VR mode...). It continuously updates the renderer and checks any interaction triggered via the mouse or controllers (event-driven behaviour). It also initializes the player in VR and tablet modes, depending on the device in which the player is been used, and the option chosen by the user when initializing the player.
- **MenuManager**: This module takes care of initializing all the menu pages view structures and all its controllers. It also attaches the menu to either the scene or the camera, depending on whether the VR or tablet mode has been enabled.
- **Controllers**: The controller is in charge of loading the correct data model (*Init()* method) and updating the corresponding view with the new data (*UpdateData()* & *UpdateView()* methods). It also includes a method to remove all the elements in case the user changes the menu page (*Exit()* method).
- **Models**: The data model lists all the elements that can change in the view due to user interaction.
- **Views**: The view module is in charge of updating the view elements with the given data model.

- **ViewStructureMenuManager**: This module describes the placement, size and appearance of all the different elements of the menu. It also joins different elements into groups. These different groups will use their own view, but can share the controller and model.
- **AudioManager**: This module takes care of anything that has to do with presentation of audio. It includes processes like initializing and updating the level or position if Ambisonic audio is used.
- **SubManager**: This module takes care for anything that has to do with presentation of subtitles. All the different methods for positioning, resizing and updating are registered here.
- **SignManager**: This module takes care of anything that has to do with presentation of sign language video. All the different methods for positioning, resizing and updating are registered here.
- **MenuFunctionsManager**: This module lists all the functions of the different controls and elements (buttons and dropdown elements) in the different menus of the UIs. This is the model part in the MVC architectural pattern.
- **MenuDictionary**: This module includes all the different text elements of the menus translated into all the supported languages.

In relation to the logical view, the **process view** for the player is mostly determined by mixing the steps and components included in Figures 8, 9 and 10. In particular, end-users create an instance of the media player by opening a web browser, typing the target URL and selecting the appropriate contents, on the initial page (ImAc portal), or by directly associating a companion device with a main TV (in the 2nd pilot phase). Right after, the ApplicationManager creates the 3D scene, with all involved elements and components included in the *Immersive Layer*. The *Accessibility Layer* will be enabled through controls of the UI, although it may be also automatically enabled based on the user's profile. That layer includes modules for the presentation of the considered accessibility assets. When more than one media component is being simultaneously presented, their spatial and temporal relationships must be preserved during playback. The *Media Sync* module is in charge of this. Moreover, key functionalities of the player are provided by the *Session Manager* module, which stores relevant information about the session status and the user's activity, as described before.

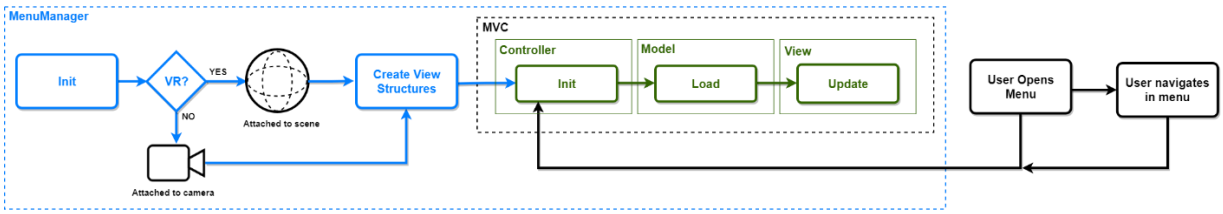


Figure 10- Model View Controller (MVC) architecture in the ImAc player

3.3.3. Physical View: Multi-Screen Scenarios

The considered scenarios in ImAc can involve either only (maybe independent) companion devices (presenting both immersive and accessibility contents) or a single main TV (presenting traditional or immersive contents) and one or multiple (n) companion devices (presenting both immersive and accessibility contents). In the latter case, a time-aligned presentation of all contents being presented on each one of the involved devices needs to be guaranteed, while enabling interaction and personalization features. An example scenario is the consumption of a director-controlled view of an event on a main TV, augmented with the consumption of alternative 360º views and accessibility contents on companion screens.

This is illustrated in Figure 4, but Figure 11 shows the involved physical nodes or entities, and their configurations in more detail, at the Content Consumption part of the ImAc platform.

The project contemplates two development phases. In the first one, initiated after the 1st pilot phase, only web scenarios are considered. This means that both the main TV and companion devices will include a web-based player for the presentation of the ImAc media contents. Simple discovery and association mechanisms have been implemented (e.g. via QR code or by typing a pin code) in JavaScript, with the use of Node.js server and web-sockets. Similarly, the DVB-CSS synchronization mechanisms adopted by HbbTV have been also implemented in JavaScript. A proof-of-concept of the synchronized multi-screen scenarios, including the dynamic switching between multi-camera 360º views on the companions devices, was presented in [3] and in [4], although they did not include all envisioned features yet.

The second development phase relies on the use of commercial TV equipment supporting the latest version of HbbTV standard v2. In such a case, the discovery, association, app launching and synchronization mechanisms are natively supported by HbbTV v2, and thus are available in the compliant equipment / devices (e.g. some recent models of Panasonic TVs). In these scenarios, the multi-screen features can be enabled by embedding the player into a native app, for app launching and communication with the TV and HbbTV features via User Datagram Protocol (UDP) sockets. This communication channel will be use to interact with the player and to control it. These scenarios allow considering broadcast linear (although not live) TV content to be played out on main (connected) TVs in the pilots. In such a case, the main TV will be able to process metadata inserted into the broadcast stream(s), signalizing the existence of accessibility services.

More details and technical specifications about the HbbTV related protocols, features and servers can be found in [5] and in [6].

The designed solutions to achieve a proper association between the involved devices and a synchronized playback are scalable and lightweight. Thus, the number of simultaneous devices presenting ImAc media contents in local scenarios is mostly limited by the available bandwidth (e.g. determined by the contract with the Internet Service Provider or ISP).

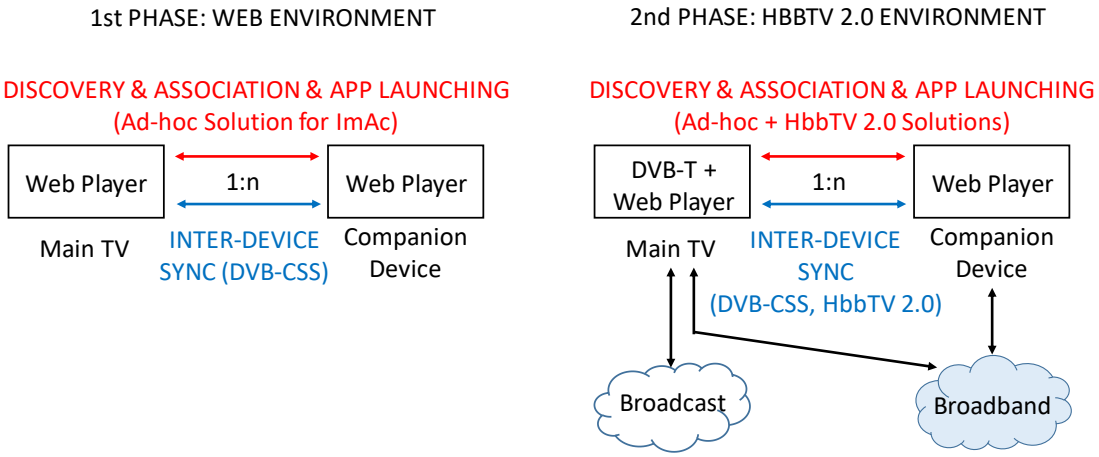


Figure 11- Physical Nodes and configurations in the consumer side

4. ASSISTIVE TECHNOLOGIES

The use of assistive technologies is considered in ImAc to achieve a more efficient usage of the player and a better comprehension of the contents (see the Assistive Layer in Figure 8). This includes the use of: zoom and enlargement functionalities for visual menus and/or controls; use of screen-readers; voice recognition; spoken feedback to the execution of these commands; etc. The first one will be described when presenting the newly designed UIs. The last two ones are briefly explained in the next sub-section. In addition, the use of media processing techniques to enhance accessibility will be explored at a later stage of the project.

4.1. Voice Control

Apart from graphical menus and controls, voice control has also been identified as a desirable requirement and interaction modality for the ImAc player. Voice control is becoming increasingly adopted (e.g. Siri, Google Now, Amazon Echo…), and this is also happening for accessibility services (e.g., Voice Over, Talkback…).

During the ImAc project, an exhaustive survey on voice control tools has been conducted. The first version of this deliverable reported on a work-in-progress implementation of the voice control feature, relying on the use of the World Wide Web Consortium (W3C) Web Speech API. However, the performance achieved in that implementation was not so successful, in terms of delays, robustness and recognition accuracy. In addition, not all the envisioned consumption devices can have access to a microphone. Therefore, it was decided to implement a more universal, modular, robust and accurate voice control solution. The decision consisted of using making use off-the-shelf interfaces such as the Amazon Echo, Google Home or Siri, and developing the required communication and interaction capabilities with the ImAc player.

Each of the off-the-shelf solutions relies upon an external cloud based service to perform the voice recognition, rather than being processed locally. This means that in order to integrate with the ImAc player an intermediate ‘gateway’ would be required in order to receive the commands from the external web service and forward it to the player, e.g. via web sockets.

At the time of development, Amazon Echo provided the most advanced API, so it was decided that it would be used for the main voice control interface for testing during the project.

4.2. Remote Control Gateway

In order to pass commands from voice control devices a ‘gateway’ has been implemented. This allows for a generic mechanism for connecting new devices, which simply need to connect to a web socket and pass a standard command. The gateway has been built using Node.js and Socket.io in order to provide a persistent service. Every device utilizing the gateway is registered on its first connection and a web socket is maintained for each device as shown in Figure 12.

There are three types of ‘device’, which can connect to the gateway:

1. Controllers – Any device that issues commands.

2. Players – Any device that consumes commands.
3. Monitors – Any device that consumes all communications for testing and monitoring.

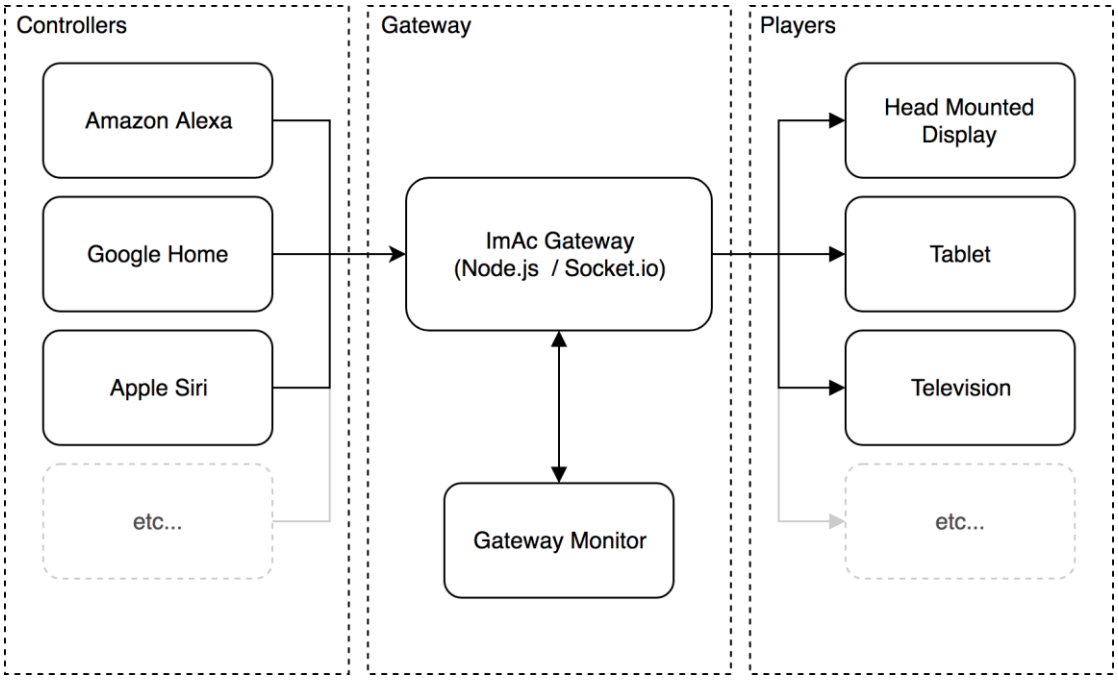


Figure 12- Gateway Architecture

Each device uses a specific Identifier, which matches the controllers to the player. For example, a player with a specific ID will only be forwarded commands from controllers with the same ID. There could be multiple players and controllers using the gateway, but only those with the same ID will talk to each other.

A web browser can be connected to the gateway to create a monitor, which shows the overall status of the server. The Monitor displays three sections as illustrated in Figure 13:

- '*Currently connected devices'*, shows all of the connections to the gateway. The gateway-monitor page will appear in the list of devices as a monitor, and you can then see each of the connected players, controllers and other monitor pages that are currently open. A graphical visualization showing the flow of commands through the network is also displayed.
- '*Latest Activity*' shows every activity that the server has performed since you opened the page. This includes devices connecting and disconnecting as well as commands being sent.
- '*Links*' gives two HTML links to other pages on the server, which replicate a controller and a player. Opening either of these will ask for the user to type in an ID, and then replicate either a controller or a player.

4.3. Voice Control Devices

In order to identify each of the controllers, the devices internal serial number is used as a unique identifier. This unique identifier is then used when connecting to the gateway, and used to direct any commands from the device to the registered player.

The basic workflow for the Amazon Echo integration is shown in Figure 14.

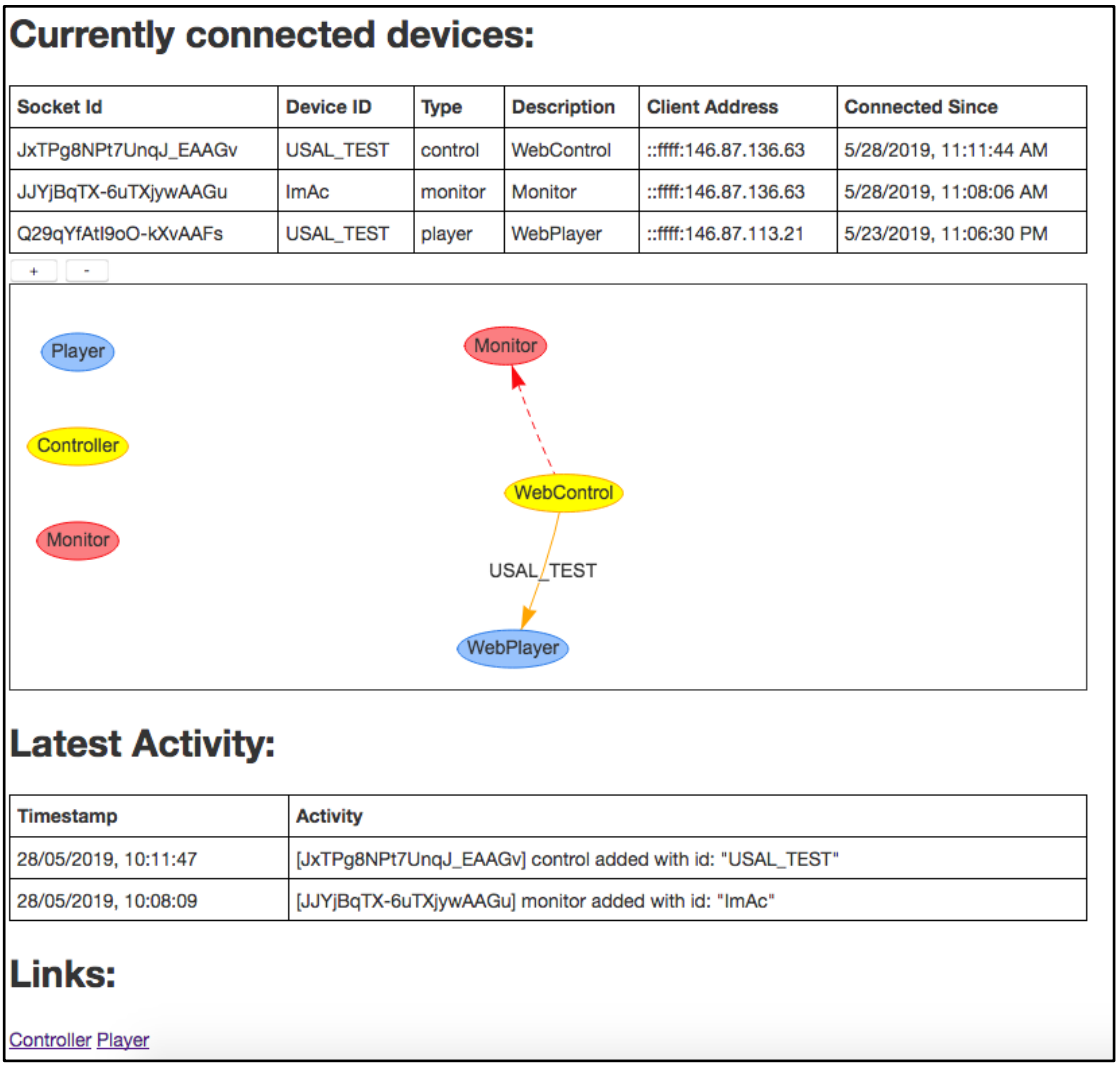


Figure 13- The Gateway Monitor View

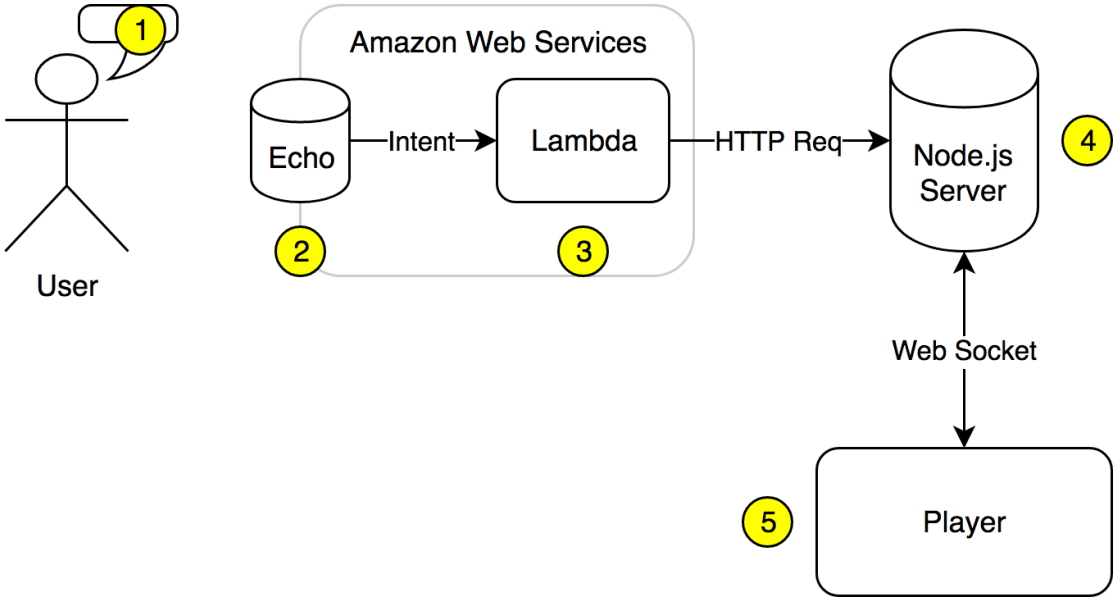


Figure 14- Workflow for the Amazon Echo Integration

In order to identify each of the controllers, the devices internal serial number is used as a unique identifier. This unique identifier is then used when connecting to the gateway, and used to direct any commands from the device to the registered player.

1. The User

The user issues commands to Alexa. An application built for Alexa is referred to as a 'skill' and you begin interacting with a skill either by issuing the command '[Wake Word] open [skill invocation name] and issuing commands or by saying '[Wake Word] ask [skill invocation name] to [command].

2. The Echo

The Alexa skills are built within the Amazon Web Services (AWS) framework (developer.amazon.com). Each skill contains a number of 'intents', where each 'intent' is designed to trigger a specific event. However, there may be multiple phrases that could be used to derive the same action. Variables such as numbers can also be defined within intent.

The intents are each defined with a unique name, and a set of sample phrases which could be used to trigger them:

```
{
 "name": "playVideoIntent",
 "slots": [],
 "samples": [
    "play"
 ]
 },
```

If it is required to capture a variable, such as a number ,the slots can be defined as a wildcard for where the values would go:

```
"name": "subsSizeIntent",
"slots": [
  {
    "name": "subSize",
    "type": "AMAZON.NUMBER"
  }
],
"samples": [
  "change subtitle size to {subSize}",
  "set subtitle size {subSize}",
  "subtitle size {subSize}"
]
```

3. AWS Lambda

The identified intent name is posted to Lambda. This is an AWS application, which allows you to run JavaScript code within AWS in a server-less manner. In essence, what this does is to bridge each intent from the Echo and forward it directly to the gateway. It also formulated a response, which is the spoken response returned to the User. It does this by sending an HTTP POST request to the gateway with the current intent.

4. The Gateway

The gateway manages all of the clients and forwards the incoming intent to any client with the same identifier through the previously connected websocket.

5. Player

The player receives each intent and interprets it as an action. For the pre-pilot tests, the player supports the following intents identified from the voice control device:

- play
- pause
- listCatalog
- volume_up
- volume_down
- volume_x
- AD_on
- AD_off
- return

- help
- forward
- backward
- open
- menu_open
- menu_close
- didnotunderstand

NOTE: ‘didnotunderstand’ is used as a generic catch all for if a command is issued but not identified.

Sample Controller implementation:

```
var socket = io();

//Register the player with a device_ID
socket.on('connect', function(){
socket.emit('setClientID', {
customId:'device_ID',
type:'control',
description:'WebControl'}
);
});

//Send a command (such as play)
function sendCommand(cmd){
socket.emit('command', 'play');
}

//Process a response
socket.on('response', function(msg){
        console.log(msg);
});
```

Sample Player implementation:

```
var socket = io();

//Register the player with a device_ID
socket.on('connect', function(){
  socket.emit('setClientID', {
customId:'device_ID',
type:'player',
```

```
description:'WebPlayer' }
  );
});

socket.on('command', function(msg){
        //Process the command msg
});
```

4.4. Connection to the Player

Once the Echo device_ID is known, in the ImAc portal / player you can enable Voice control from the menu. This is listed under 'General Settings'-> 'Voice Control'. When you turn it on, it will ask for an ID. This is the device_ID that it is listening to – i.e. the id identified in the previous step, as shown in Figure 15.

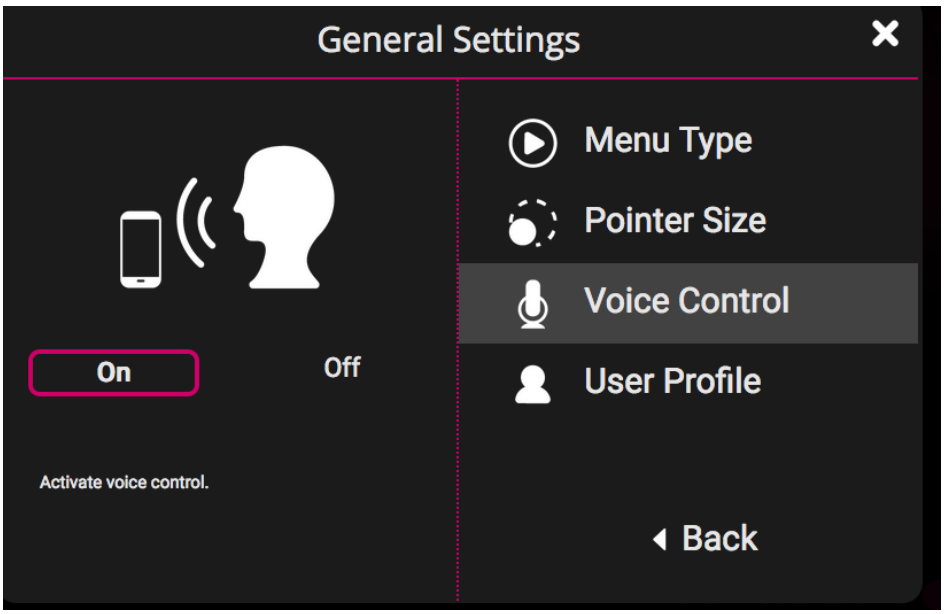


Figure 15- The portal / player voice control Menu

The player will also remember this ID and setting, so it is suggested to put the ID into the player whilst it is in the browser mode, before going full screen VR. The gateway-monitor can be accessed In order to confirm that player, with the given ID, has connected correctly.

You can connect more than one player to the gateway if you chose, such as one on a HMD and another in a browser and they will receive the same commands.

4.5. Commands implemented for the pre-pilot tests

The map of the flow of commands considered for the pre-pilot tests phase 2 is shown below. This diagram is only presented in English, however it has been directly translated into Spanish

and Catalan. The list of intents and their responses, in each of the languages are listed in Annex III. It is expected to re-define and extend this list, once analysing the results from the pre-pilot tests, and preparing the implementation for pilot 2.

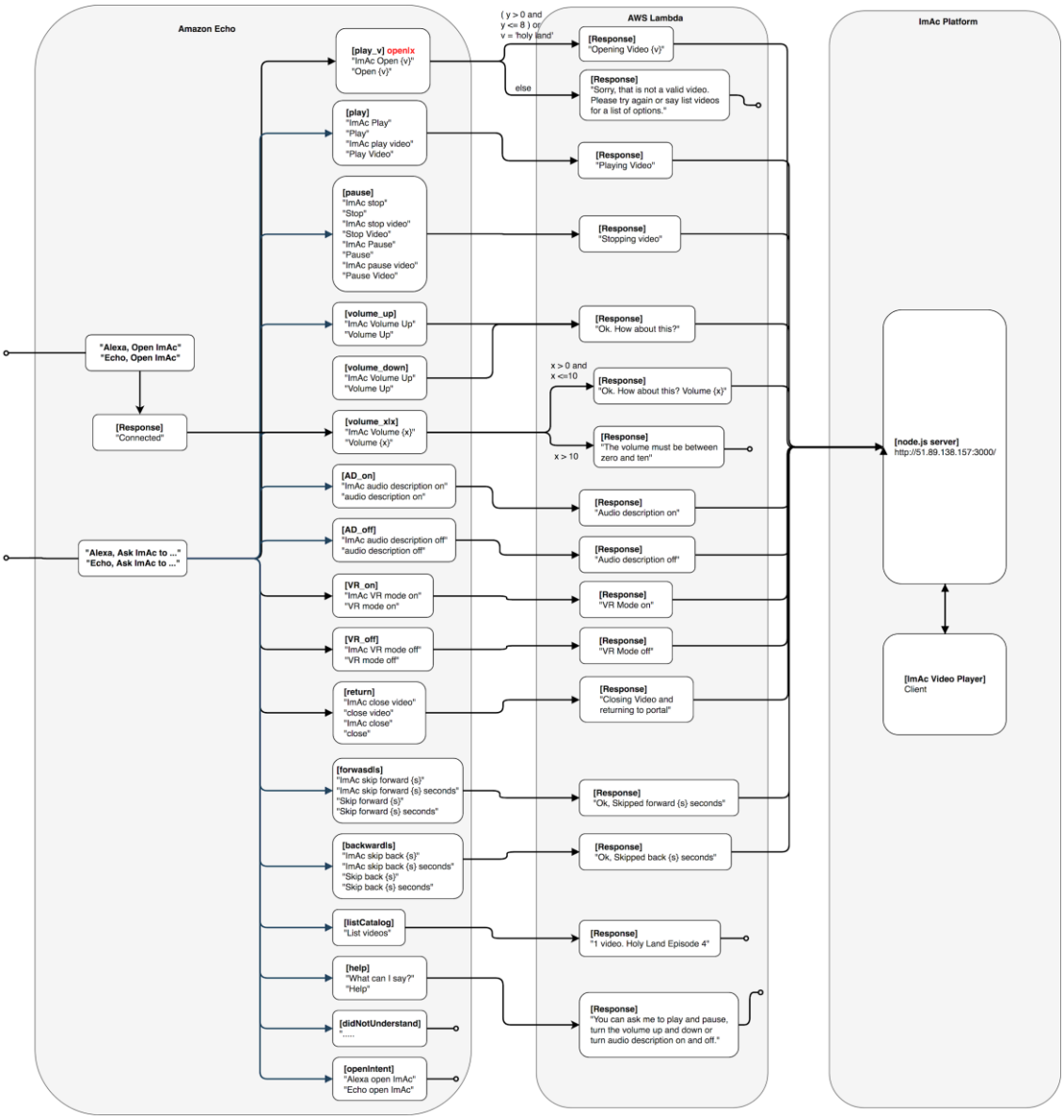


Figure 16- Flow of commands available in pre-pilot tests

5. EVALUATION AND MONITORING MODULES

5.1. Key Performance Indicators (KPIs) measurement

In video streaming services, it is essential to monitor relevant Quality of Service (QoS) and Quality of Experience (QoE) related metrics to better understand what are the limitations, propose appropriate solutions and corroborate the obtained performance. This is especially relevant when streaming high quality and omnidirectional media, which is more challenging and requires more resources than the streaming of traditional media.

With this premise in mind, a modular and extensible testbed to monitor and register Key Performance Indicators (KPIs) when using the ImAc player has been developed. In this context, the dash.js player provides an API that allows obtaining statistics about certain KPIs regarding the incoming media streams. This API is used during the streaming session to get many of these KPIs, in addition to other related ones that are calculated by using the information from these KPIs, together with newly developed methods.

Although it is possible to register the gathered statistics within the memory of the web browser and/or in local files for each session, a Node.js server with a MongoDB database has been developed in order to register these KPIs, by making use of an HTTP-based communication protocol. This provides a better performance and stability of the player, getting persistence of the gathered statistics, and eliminates the need for having control on the consumption device on which the player has been run to retrieve the statistics. In addition, this allows gathering statistics from distributed and large-scale sessions, and enables higher flexibility for their analysis (e.g., by applying clustering or correlation strategies). The frequency of measurement and registration of the KPIs (and thus of the communication with the database) can be configured in the player. Figure 16 shows a high-level overview of the developed testbed.

In particular, the following KPIs are measured and registered in the presented testbed:

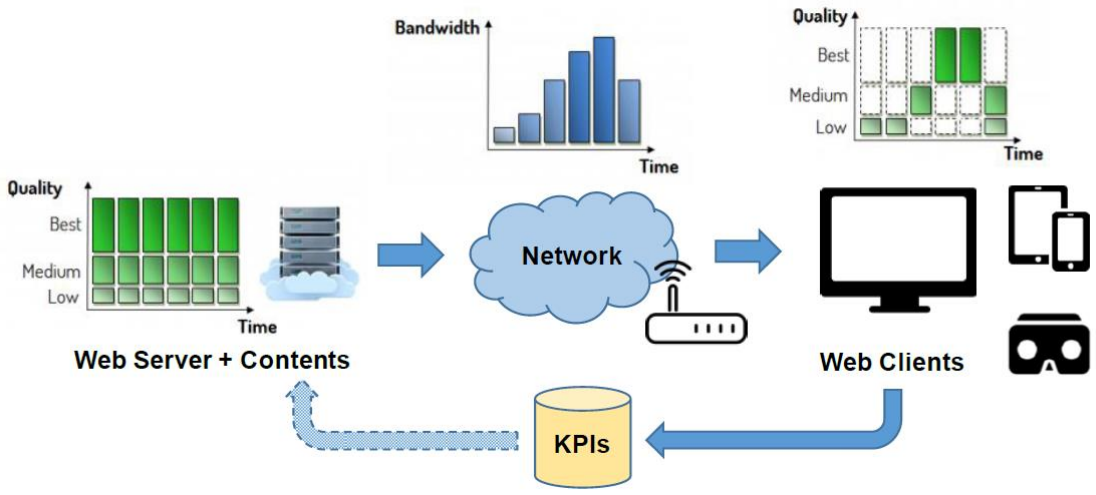


Figure 17- Overview of the testbed for KPI measurement and registration

1) Objective / QoS-related KPIs:

- Video Start-up Latency: Delay since the play button is clicked until the video is actually watched.
- Evolution of Latency: Periodic measurement of the end-to-end delay during the streaming session. Jitter can be also calculated as the variation of the delay.
- Throughput: Effective bandwidth during the session.
- Video Quality: The representation or quality index selected for each DASH segment during the session. Based on this KPI, the Average Video Quality, as well as the Number, Frequency and Magnitude of Quality Switches can be also calculated.
- Video Stalls: The evolution of both the playout time and the absolute time (e.g., NTP based timestamps) can be periodically monitored. If they do not advance in accordance, and no playout control commands have been executed, it can be a sign of the occurrence of video stalls or a non-natural evolution of the playout process. This measurement determines how smooth / uninterrupted the playout is during the media session.
- Buffer Fullness Level: Occupancy of the playout buffer during the session (e.g., in percentages or in time units). If playout stalls happen, it can be due to buffer underflow / overflow situations. This information can be also used to select the most appropriate quality for the segments to be downloaded in order to avoid, or recover from, the undesirable underflow / overflow situations.
- Asynchrony: By comparing the playout and absolute times, the asynchrony (i.e., playout time difference) between media elements played out within the same device (i.e., local inter-media synchronization) and across devices (i.e., inter-device or inter-destination synchronization) can be calculated.
- Viewing Direction: When watching 360º videos, the current latitude and longitude angles referred to the centre of the FoV can be measured. This information can be more accurate if eye-tracking functionality is available.
- Duration of the session: Amount of time since a video is selected and the session for this video is terminated. It can be shorter than the duration of the video if the session is terminated before the end of the video or seek forward commands have been executed, but also can be longer if pause or seek backwards commands have been executed.

2) Subjective / QoE-related KPIs: In the presented testbed, the received audio and video files can actually be played out. Therefore, having knowledge about the obtained objective KPIs is important to better understand the allowable thresholds and ranges that can be tolerated by users, and result in satisfactory QoE levels, as well as to correlate QoS and QoE metrics. Next, key QoE related aspects are highlighted:

- Video Start-up & End-to-End Latency: To what extent delays are tolerable by users.
- Video stalls / pixilation / visual anomalies: To what extent these effects are tolerable by users.
- Overall Media Quality: To what extent the overall media (audio / video) quality is tolerable by users, and the adopted quality switching algorithm can impact the perceived QoE (e.g., abrupt transitions can be noticeable and even annoying to users).

- Synchronization levels: To what extent inter-media and inter-device synchronization skews impact the perceived QoE. The impact of playout adjustments to achieve synchronization also apply in this context.
- Smooth 360º Exploration: To what extent the transitions between FoVs within the whole 360º area are perceived as smooth / instantaneous to users, and the perceived video quality in such transitions are perceived as acceptable. This KPI metric is relevant when using tiling / FoV-based streaming solutions, which will allow minimizing latency and bandwidth consumption compared to traditional solutions based on streaming the whole 360º area.

All these aspects can be evaluated by conducting subjective tests, making use of questionnaires, including Mean Opinion Score (MOS) metrics or likert scales as evaluation metrics. Then, the obtained results from the subjective evaluations can be correlated with the measured objective results (e.g., when having used specific configurations and/or having forced specific conditions).

All these metrics can be extended in future releases by making use of the dash.js API or even extending it.

More details about this testbed and examples of statistics that can be register by using it can be found in [7].

5.2. User Activities / Behaviour

The intention in pilot 2, especially in the open pilots, is to monitor and register the users' activity when accessing and using the ImAc player. Different tools for achieving this have been analysed, and Google Analytics (https://analytics.google.com/analytics/web/) has been chosen. The following information and statistics can be gathered:

- Regarding the users: an id, the region from which they access, the acess time, and the type of device and browser being used. It is also possible to register if it is the first time the user access the player or has accessed before, and their viewing patterns, by recording the centre of their FoV periodically.
- Regarding the ImAc portal: options and settings that have been used, and the selected video, including timestamps for all actions.
- Regarding the selected video: what access services are selected (e.g. ST, SL, AD, AST) and their languages, for how long, what settings of the player menu are used / enabled, and what playout controls are used, including timestamps for all these selections.

At the time of submitting the deliverable, the analysis of the possibilities and the convenient features and aspects to be monitored is ongoing, and a discussion between the WP5 partners is planned in September to decide on the final implementation.

6. HOW TO

6.1. How to Access the Code

The code of the ImAc player, together with some examples of immersive and accessibility contents, are available at the Secure File Transfer Protocol (SFTP) server used in the ImAc project. Credentials to have access to it via FTP clients, like FileZilla, have been created. These are detailed below:

- Server: imac.gpac-licensing.com
- User: ****
- Password: ****

Likewise, the source code of the player is available on Github, so the research community can make use of it. Link: https://github.com/ua-i2cat/ImAc

6.2. How to Install / Configure the ImAc player

The ImAc player has been developed exclusively using web-based components. Therefore, just two key software components are required to install and make use of it: a web server (like Apache); and a web browser at the client side (Chromium preferably). The resources available at the SFTP server must be downloaded and placed into the proper folder within the web server. After this process and activating the web server, the ImAc player will be available via the concrete URL formed by the IP address or the web domain of the server, and the given extension or directory where the ImAc player resources have been placed.

Note that a Node.js server together with the associated code will be also necessary for the voice control and Node.js features.

6.3. How to Access the installed ImAc player

Up to date, various version of the player have been released, and are detailed in Annex I, with their associated access URLs and available features.

The test or development version of the player can be accessed via the following URL: https://imac.gpac-licensing.com/playertest/

The stable version of the player can be accessed via the following URL: https://imac.gpac-licensing.com/player/

7. USER MANUAL

This section describes how to use the ImAc player, after having successfully installed the server and client components to make it running, or by directly accessing the URL on which the player is accessible.

Note that the player has significantly evolved since the submission of the first iteration of the deliverable, as detailed in Annex I. In order to be able of visually checking the progress made, the User Manual of that version of the player is provided in Annex IV. It has not been included in this section to ease its reading and comprehension. Finally, a factsheet summarizing the key technical specifications and features of the player is provided in Annex V. This factsheet is being used in the dissemination and exploitation actions conducted in the project.

A demo video of the player can be watched at the Youtube channel of the project: https://www.youtube.com/watch?v=cHTodj3pCJM. This video also provides a summary about how to user the player and its available features. In addition, the technical and functional features of the player can be found in some scientific publications, like [3], [7], [8] and [9].

7.1. Access the Player

The player is web-based, so it can be accessed by just typing an URL into a web browser. At the time of submission of this deliverable, the most recent release of the public version of the player is available here: https://imac.gpac-licensing.com/player/

The list of the released versions of the ImAc player up to date, with their features, contents and purpose, is provided in Annex I.

7.2. Initial Screen: Selection of Contents and Settings

When typing the URL of the ImAc player, an initial page listing the available contents is shown, as can be seen in Figure 18. This landing page has a modern, intuitive and accessible UI. It includes the list of available videos and information about them, such as their title, cover, thumbnail, duration, language of the main audio and the available access services. All this information is retrieved via the metadata generated at the production side, when creating and preparing the contents, as described in Annex II. Information about the project is also provided in the footer (e.g. logo, webpage, partners, Social Media channels…). This initial page has been called the *ImAc portal*.

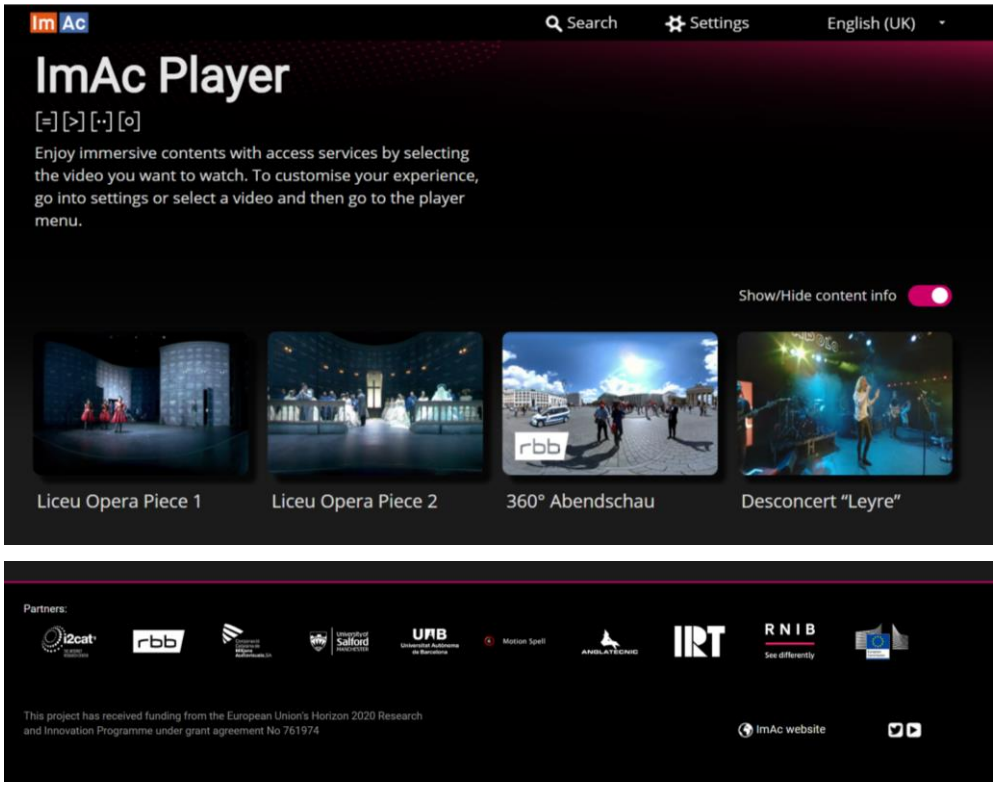


Figure 18- ImAc portal: initial Screen listing the contents and with initial settings

Through this initial screen the users can select the desired language for the UI, as well as some initial settings that can be changed afterwards in the player menu during media consumption.

By clicking on the Settings menu, the user can select / activate General Settings for the player, Settings common for the Access Services, and specific Setting for each Access Service (see Figure 19). A brief text is added to indicate the effect / purpose of each settings option.

In the General Settings sub-menu, the users can set / select:

- The menu type (Figure 20): The previous version of the player included two different player menus (see Annex IV). Based on the results from pilot 1, a new single menu has been designed in this evolved version of the player, but it includes an enlargement feature to improve accessibility, in a similar manner to the enhanced-accessibility (aka low-sighted) UI available in the previous version of the player. This way, the usability of the player improves by just having one single UI, while meeting the requirements for having enlarged UI control and text. This settings option allows the user to select their preferred menu type where the features and controls will be discussed later.
- The pointer size: In VR environments, the UI controls can be typically selected / set via a pointer. One requirement was the possibility of personalizing the size of this pointer, and this can be done in this settings option, by choosing between three pre-defined sizes: Small, Medium, and Large.

- Voice Control (Figure 21): through this settings option the voice control feature can be Activated / Deactivated. When activated, the Device ID to be connected to is requested via a text box.
- Save and Erase the User's Profile: This settings option allows the users to save the selected / activated settings, so they can be loaded in future usages of the player. It also allows erasing these settings.

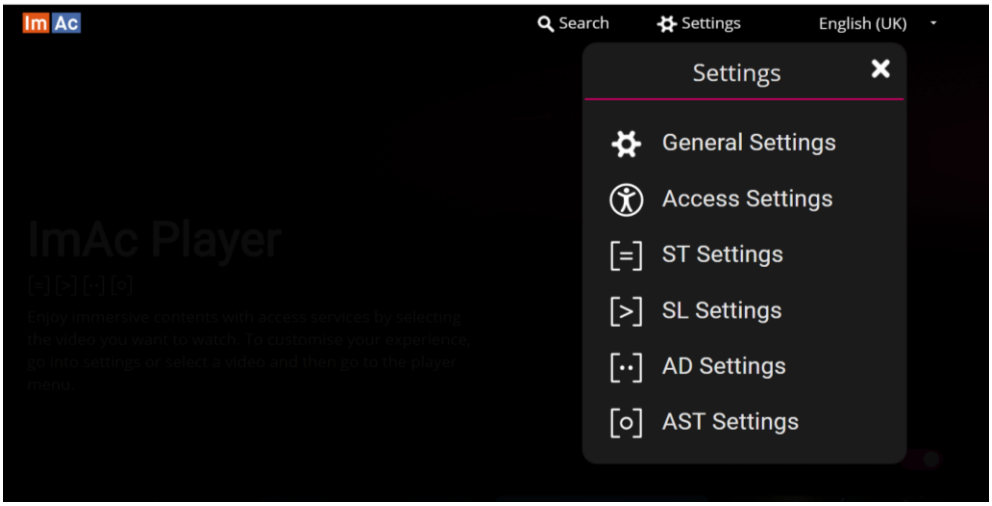


Figure 19- ImAc portal: Settings menu

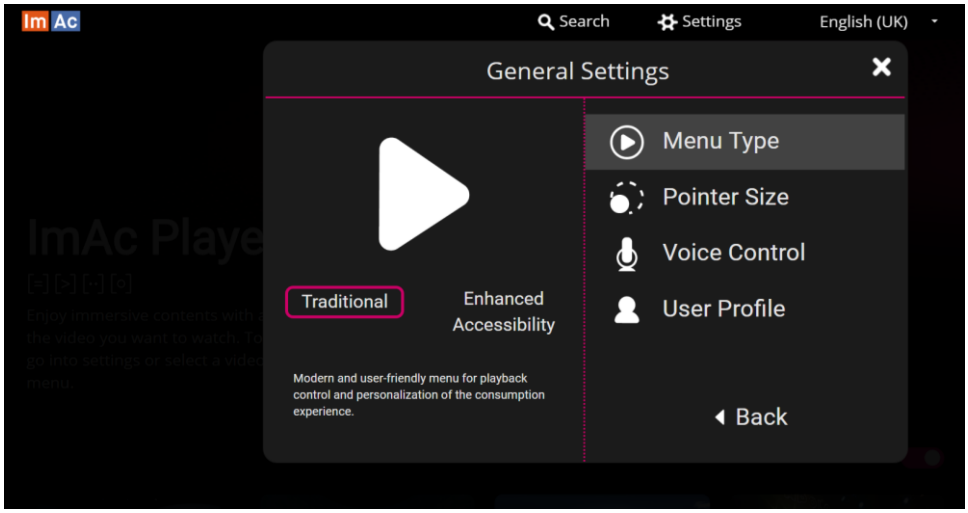


Figure 20- ImAc portal: General Settings Sub-Menu

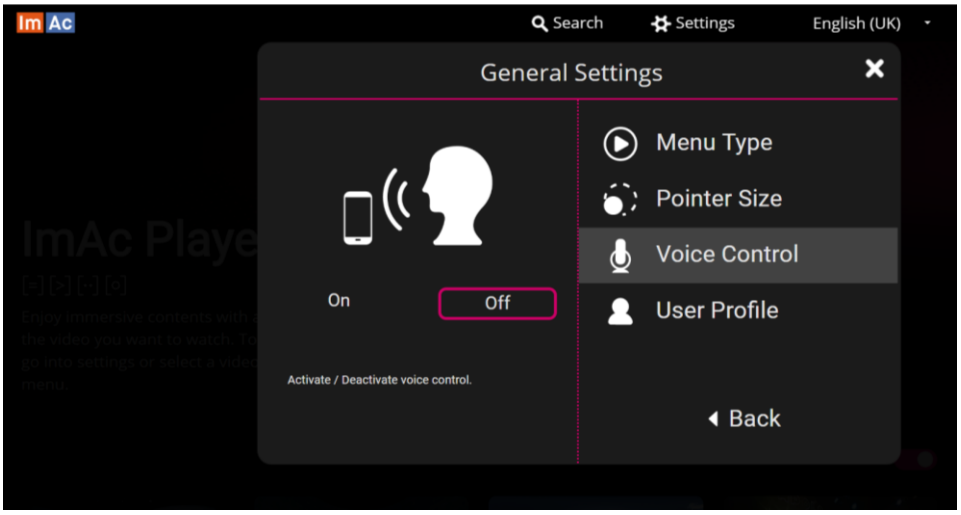


Figure 21- ImAc portal: Voice Control Settings

In the Access Settings sub-menu, the users can set / select:

- The language for the Access Services.
- The indicator (Figure 22): It is the visual guiding method to indicate where the target speaker is in the 360º area when ST and/or SL are activated. The available options are: arrows, radar or none.
- The Size of the Safe Area (Figure 23): It allows choosing the preferred size of the Safe Area, which is a sub-region of the FoV where to place the visual elements on screen in order to provide a comfortable viewing experience. The sizes of the available options have been derived from evaluation conducted in the pre-pilot tests.

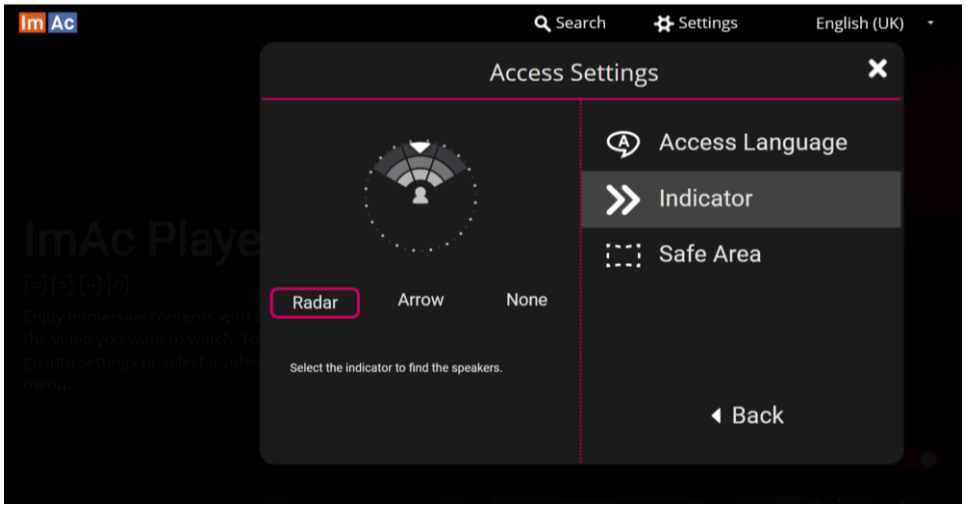


Figure 22- ImAc portal: Indicator Settings

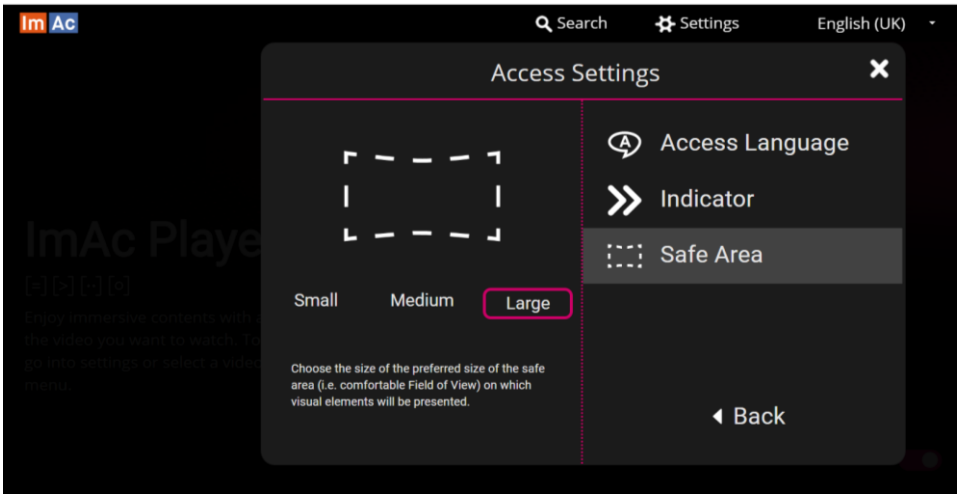


Figure 23- ImAc portal: Safe Area Settings

In the ST Settings sub-menu (Figure 24), the users can set / select:

- The preferred subtitles size.
- The preferred option for the subtitles background: outline, or a semi-transparent background box.
- The preferred position: top, bottom.
- Easy-to-Read or Traditional ST.

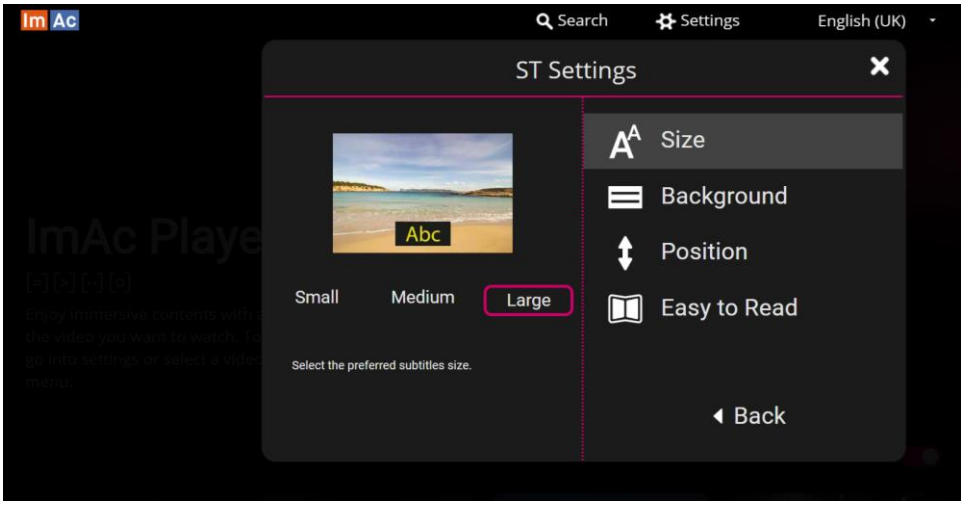


Figure 24- ImAc portal: ST Settings

In the SL Settings sub-menu (Figure 25), the users can set / select:

- The preferred size for the SL video window
- The preferred position: right, left.

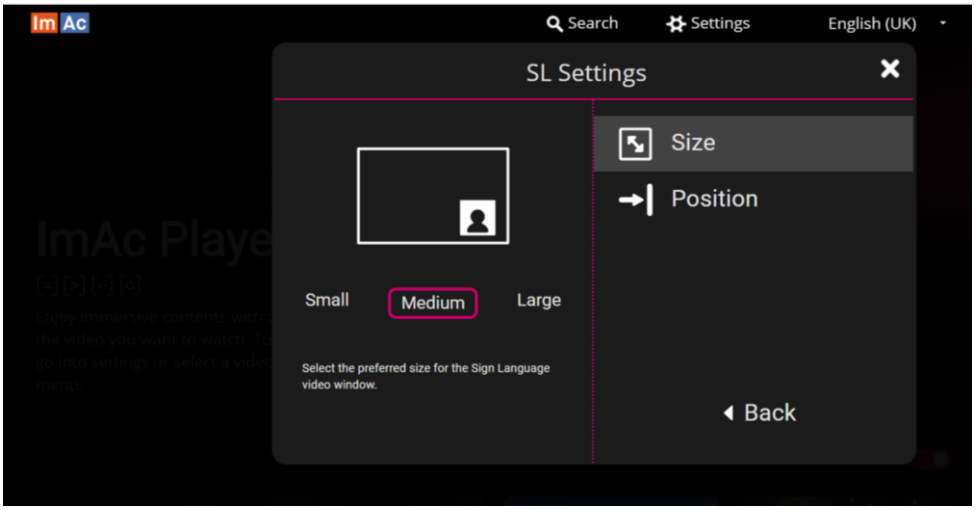


Figure 25- ImAc portal: SL Settings

In the AD Settings sub-menu (Figure 26), the users can set / select:

- The preferred Presentation Mode: Classic (Voice of God, or positioned in-head), Static (like a friend whispering in your ear) and Dynamic (audio comes from where the speaker or action is).
- The preferred relative volume level compared to the main audio track: Low, Medium or High.

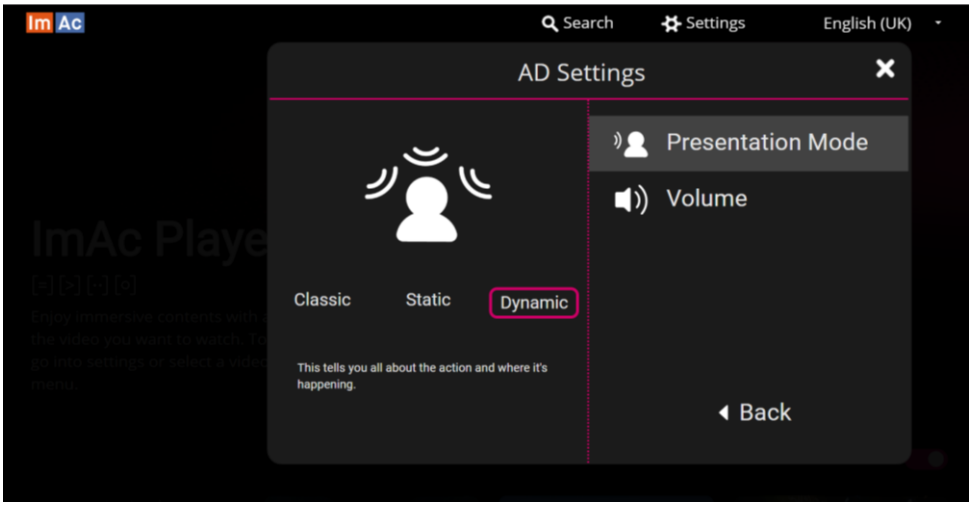


Figure 26- ImAc portal: AD Settings

In the AST Settings sub-menu (Figure 27), the users can set / select:

- The preferred Presentation Mode: Classic (Voice of God, or positioned in-head) or Dynamic (audio comes from where the speaker or action is).
- The preferred volume level for the AST track.
- Easy-to-Read or Traditional AST.

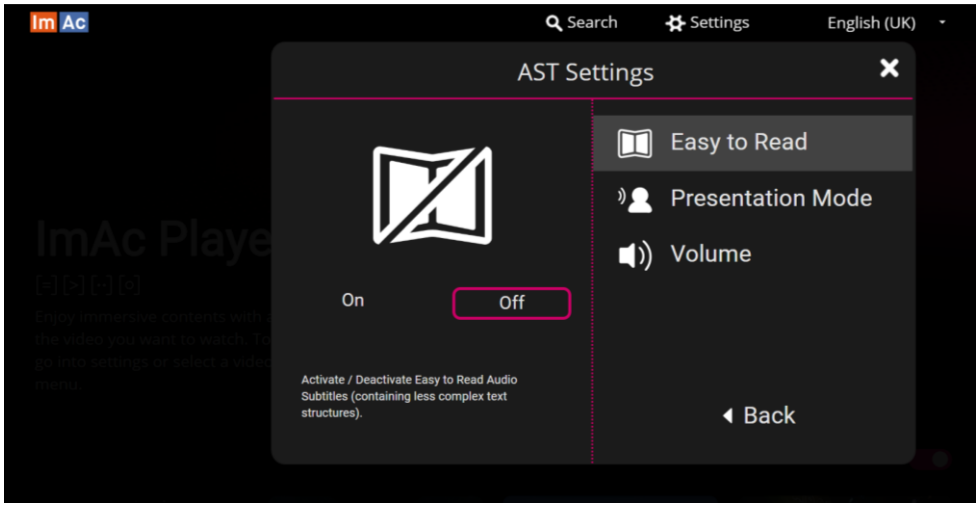


Figure 27- ImAc portal: AST Settings

Likewise, the Search menu allows users to search for specific contents by their title and/or by filtering the displayed contents by their available Access Service and/or Language (Figure 28).

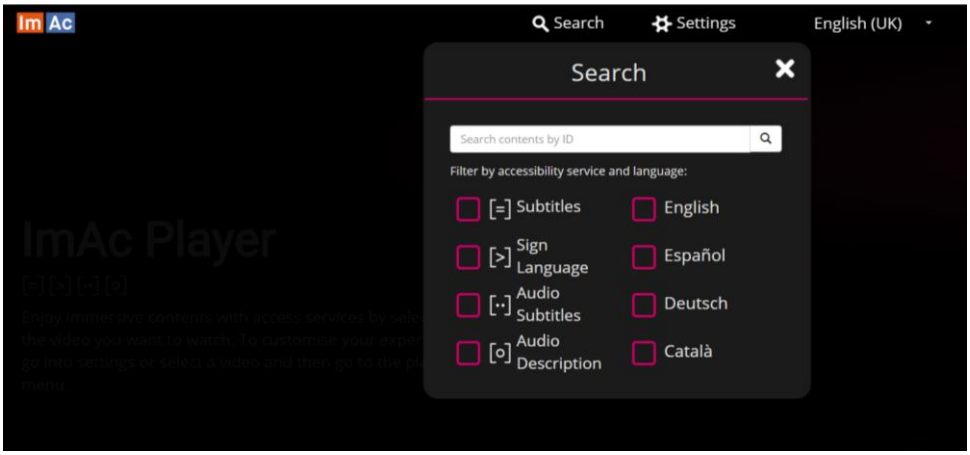


Figure 28- ImAc portal: Search and Filtering Features

Once clicking on the thumbnail of a specific clip (Figure 18), then its video cover is shown together with more information about this selected clip (duration, synopsis, available access services…), as shown in Figure 29. If the user wishes to consume the selected content, it can be loaded by clicking on any of the two available Play buttons (next to the synopsis and as an overlay to the thumbnail). If the Play button is clicked the player will be start. If a VR-enabled device is being used, a menu will appear asking if the player has to be run in tablet mode or in VR mode. This initial decision will have an impact on the positioning of the UI elements on the player, which are either attached to the screen (tablet mode) or to scene (VR mode). After the player starts a banner will be shown indicating how to open the UI (see Figure 30). A cookie with a validity of one month has been added to not have to shown this banner multiple times for each user and usages of the player. It means that the banner will not be shown after the first time, until the cookie is erased after one month.

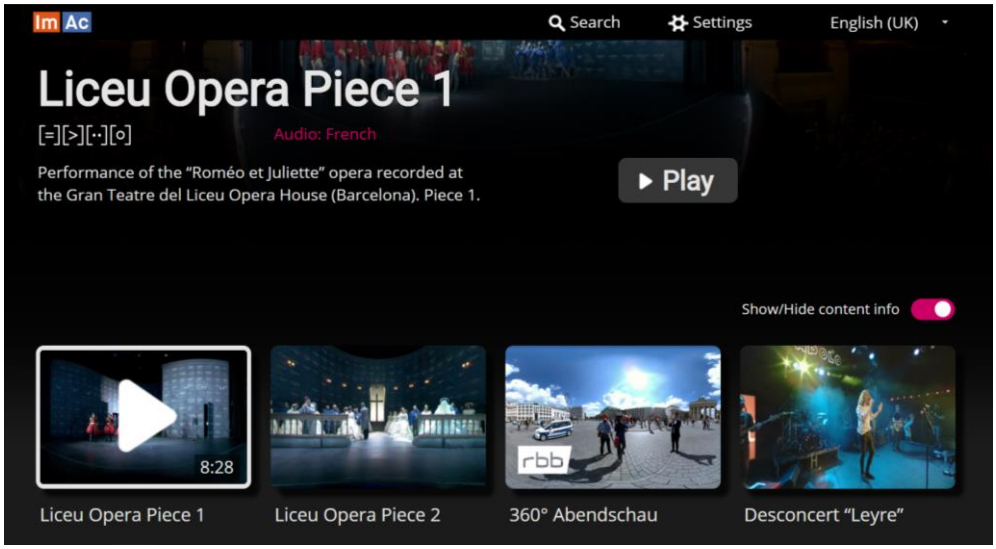


Figure 29- ImAc portal: Selected content

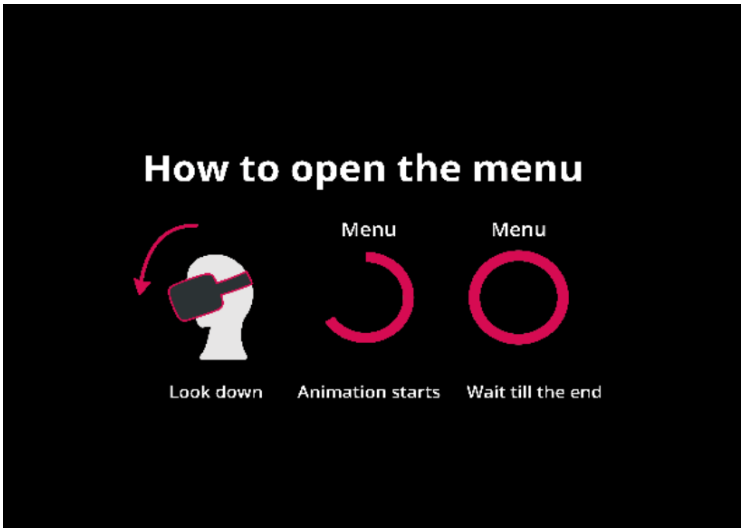


Figure 30- Banner indicating how the UI can be opened when starting the ImAc player

7.3. UIs and Features

The designed UI takes into account the sensorial capacities of the target users, and the requirements and preferences identified in the focus groups (gathered in WP2, and detailed in D2.2). Likewise, it takes into account the particularities of each device and platform, together with the physical limitations of each screen, in order to be easily interpreted/used by all users, regardless of their sensorial abilities and the device being used. This may involve that multiple pathways for media control and interaction need to be provided.

Two different UIs were designed for the player version used in pilot 1 (see Annex IV), based on the requirements collected from the user-centric activities conducted in the first year of the project and on the two design proposals by the broadcasters of the consortium. However, there were multiple reasons to design evolved or refined versions of both UI variants. On the one hand, feedback from T3.4 indicated that the number of clicks to activate / deactivate access services (issue in both menus) should be minimized. On the other hand, the results from pilot 1 concluded that: 1) the toggle button to activate / deactivate access services in these UIs was not so intuitive by users (tested in the traditional UI, but also applies to the low-sighted); 2) usability of the traditional UI was not good in small screens (e.g. smartphones); 3) clustering some common settings for the access services (e.g. language, indicators, safe area...) would increase simplicity; 4) having two distinct UIs may result in a less intuitive and simple usage of the player.

Accordingly, it was decided to design and implement a single menu with a bigger size variant to make the usage of the player simpler and more intuitive, while still meeting all requirements (e.g. enlargement of the UI controls), based on the lessons learned in the first year of the project. Different options for menus, colours, fonts and practices in VR for a good usability and accessibility were investigated.

While watching the video, the menu can be opened by looking down for a period of time (Figure 30), by performing consecutive clicks via the mouse/keyboard/touchpad, or by performing a long click on the VR controllers. This allows having a clean screen without any visual element while watching VR contents, which can have a negative impact on the user's immersion. The menu can be closed by clicking on the exit control (top right), as in typical menus and screens.

Figure 31 shows how the newly designed player menu looks. It includes the necessary controls to activate / deactivate the required accessibility services (with just one click) and to set the different options and features to meet the requirements gathered in WP2. The icons proposed by the DR (Danish Broadcaster) for the accessibility services have been adopted (see Figure 32): https://www.dr.dk/om-dr/about-dr/smart-icons-design-common-european-standardization

Figure 31- Player menu

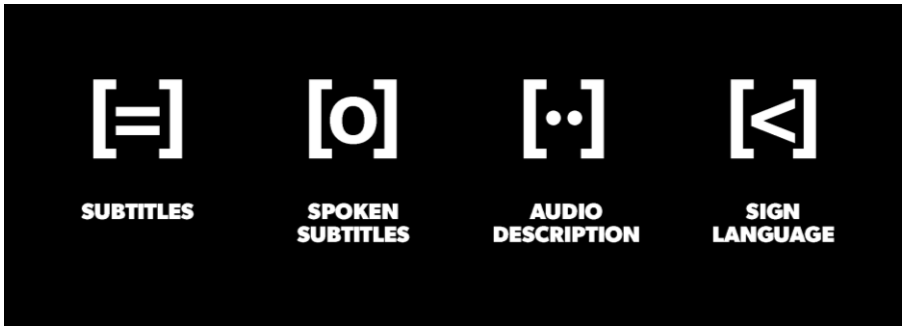


Figure 32- Accessibility Icons adopted in the player

7.4. Playback Controls and Volume Settings

The player includes the following playback controls: play, pause, seek forward and seek backward (Figure 31). The seek steps have been set to 5s. In addition, the traditional UI includes a progress bar that allows a customized seeking by clicking on any point of the bar (Figure 31). When clicking on any of the graphical playback controls, their size is reduced and their colour is changed to yellow (for high contrast) for a short period of time to provide visual feedback about their execution. This can be appreciated in Figure 33 for the “play” button, when compared to Figure 31. Specifically, when clicking on the “play” control, the “pause” control will be shown, and vice versa (see Figures 31 and 33). At any moment, the current playback time and the duration of the video are shown at the edges of the progress bar (see Figure 31).

The UI also includes volume setting controls (see Figure 31). In particular, it allows increasing and decreasing the volume level, in steps of 10% of the maximum volume. When clicking on them, the current volume level will be shortly displayed (see Figure 34). In addition, by clicking on the loudspeaker button, a strike will be added to this icon, indicating that the “mute” mode has been set (see Figure 35).

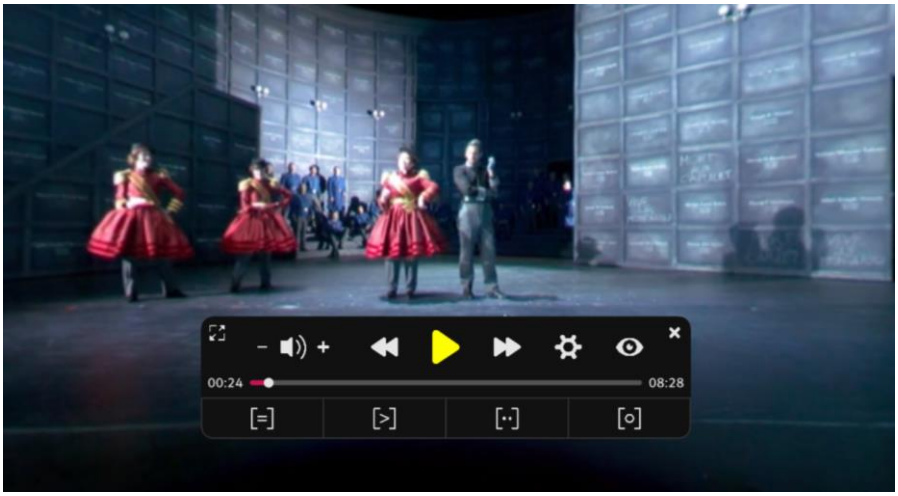


Figure 33- Player menu: Visual Feedback to the execution of commands

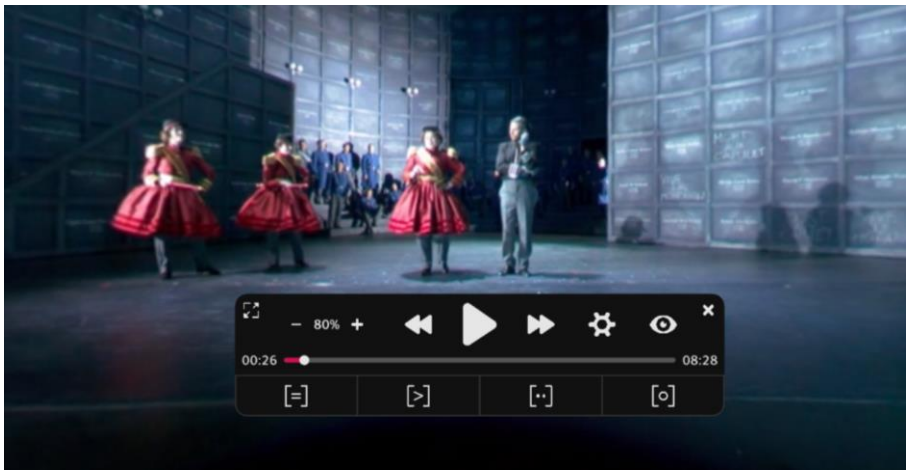


Figure 34- Volume setting controls: current level

Figure 35- Volume setting controls: mute

7.5. (Personalized) Presentation of Subtitles (ST)

The menu enables the activation and de-activation and setting each one of the accessibility services, namely: Subtitles (ST); Sign Language (SL); Audio Description (AD); and Audio Subtitles (AST). Subtitles, in the same manner as all the other accessibility services, can be activated / de-activated by clicking on the controls with the accessibility icons. When activated, this icon become thicker and has magenta colour (see Figures 36), thus also providing visual feedback on the current status.

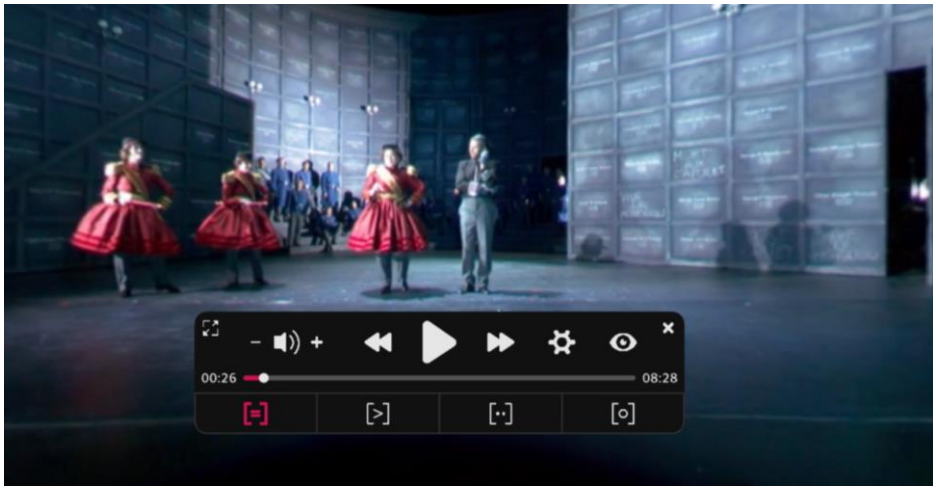


Figure 36- Visual feedback on the activation / de-activation of access services

Based on the requirements gathered in WP2, the subtitles will be always visible and centred in the FoV, regardless of whether the related scenes/objects in the 360º area are visible or not (see D2.2). However, different guiding mechanisms are provided to help the users in finding the associated speakers (explained later). Note that other presentation modes have been explored and have been implemented in the player, like rendering subtitles evenly spaced

every 120º in the 360º environment, and rendering the subtitles attached to the speaker by using always-visible indicators. However, always-visible subtitles have been proven to be preferred by users. Details about how these implementations look like and the obtained results can be checked in WP5 deliverables.

Likewise, the player supports the presentation of subtitles in different colours, based on metadata generated at the production side. This contributes to a better identification of different speakers.

Next the different available options for presentation of subtitles are described.

7.5.1. Selection of ST Track

If subtitle tracks in different languages are available, users will be able to select between them. For this, the preferred language (Figure 37) needs to be selected by clicking on the Accessibility list item in the General Settings menu, which is opened by clicking on the wheel / setting control in the menu (Figure 38). This setting is common to all access services.

Likewise, if available, the selection of Easy-to-Read ST can be done by accessing to the Subtitles item in the General Settings menu (Figure 39)

Note that the setting options will only appear if the access service(s) they refer to are available for the selected content. In addition, the previously selected / activated options, via the player menu or via the ImAc portal are highlighted in yellow colour (thus also providing high contrast).

Figure 37- Selection of Language for the Access Service

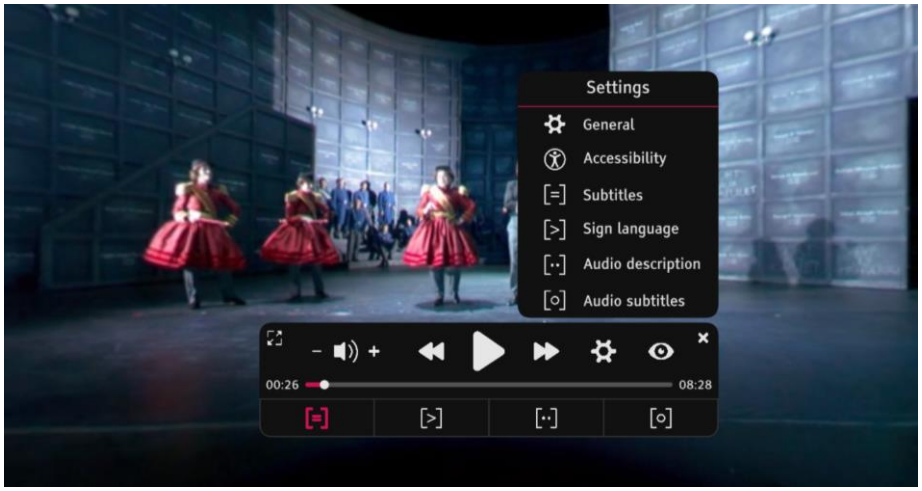


Figure 38- General Settings sub-menu

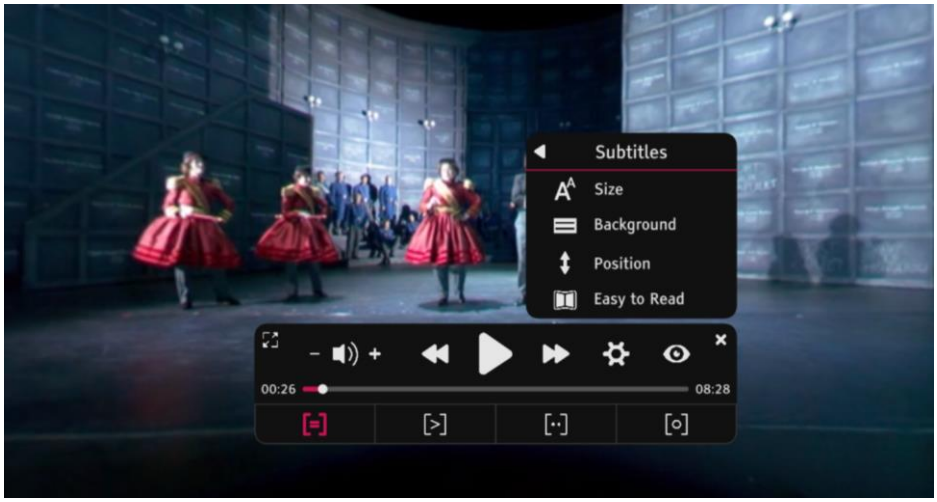


Figure 39- ST Setting sub-menu

7.5.2. Dynamic Positioning

Regarding their position, subtitles are typically visible in the user's field of view in the middle slightly below eye line, two-lined. The ImAc player follows this approach, but it also allows dynamically setting the position of subtitles for their presentation at the top. In particular, users are able to select between two pre-defined positions in the viewing field, as already described when presenting the ImAc portal:

- Bottom  (Figure 40) - Top  (Figure 41)

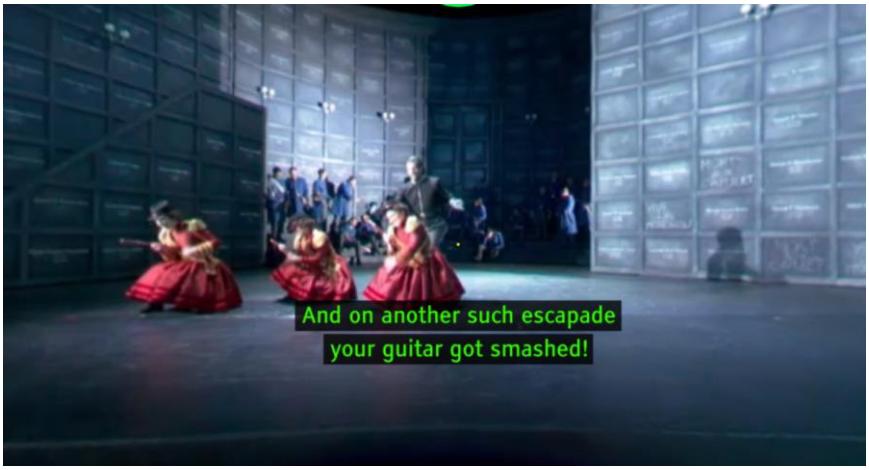


Figure 40- Subtitles placed at the bottom-centered position of the FoV / Safe Area

Figure 41- Subtitles placed at the top-centered position of the FoV / Safe Area

7.5.3. Background

Users are able to select between two different backgrounds for the subtitle area: 1) semi-transparent box, with 80% of opacity (see Figure 40); 2) outline, with 2px for each font size (see Figure 41).

7.5.4. Subtitle Size

The TV broadcast subtitle services are still based on the Teletext limitations, as well as on viewer behaviour, including their reading speed. That implies that the number of characters is limited to about 37 characters per line to avoid line breaks, and it is suggested to use not more than 2 lines for a subtitle frame. ImAc is making use of these recommendations also in the VR world. These recommendations, together with the available area for display graphical elements (i.e. the Comfortable FoV (CFoV) or Safe Area, see next item), determine the maximum size for subtitles. In addition to that, and based on requirements from WP2, three

different sizes can be enabled for the presentation of subtitles (see Figure 39): large (based on the maximum size), medium, and small.

In addition, the player adopts the responsible subtitles approach described in D3.1, which allows automatically re-blocking subtitles based on the available container area.

7.5.5. Safe area or Comfortable Field of View (CFoV)

The most appropriate size for the safe area where to present the accessibility contents and UI controls were investigated in the pre-pilot tests. Different percentages of both 1:1 and 16:9 ratios were considered, being the latter ones more adequate (see Figure 42). The EBU guideline EBU R95 (https://tech.ebu.ch/docs/r/r095.pdf) that determines the safe area for 16:9 screens in a High Definition format was taken into account in these tests.

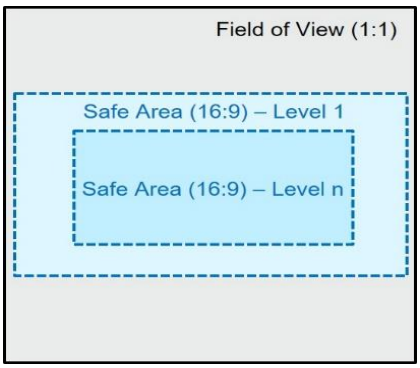


Figure 42- Levels of the Safe Area to determine the most Comfortable viewing experience

It was found that there were no clear preferences between three percentages of the 16:9 ratio (60%, 70% and 80%) and that the HMD in use has a significant impact, as their FoVs may differ. Similarly, these tests were also performed for SL and the preferred levels were 50%, 60% and 70%. Therefore, it was decided to let the users choosing between two common levels (i.e. 60% and 70% of the FoV) for ST and SL, as these services could be activated at the same time (see Figure 43). When each option is selected, the edge of the area in shown by using a dotted yellow line for a short period (see Figure 44). This was added to help the users to better understand this functionality.

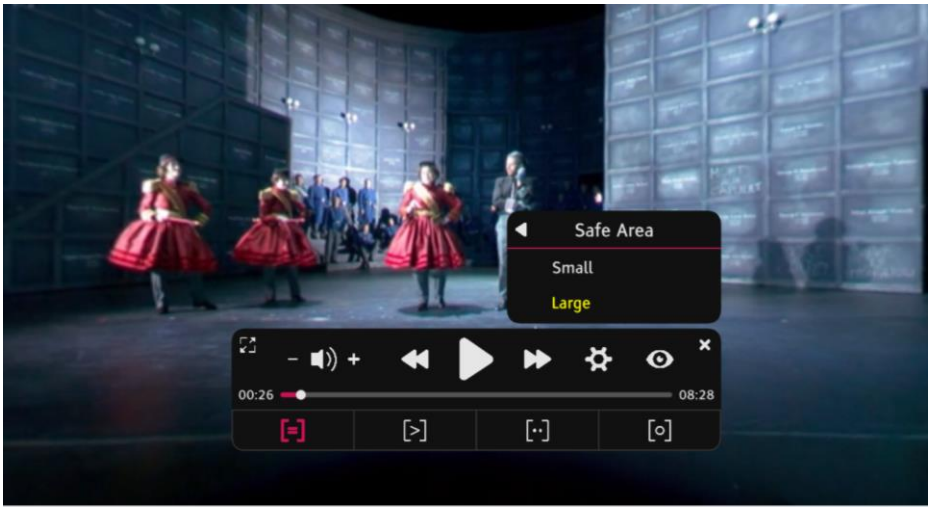


Figure 43- Menu options to set the size of the safe area or CFoV

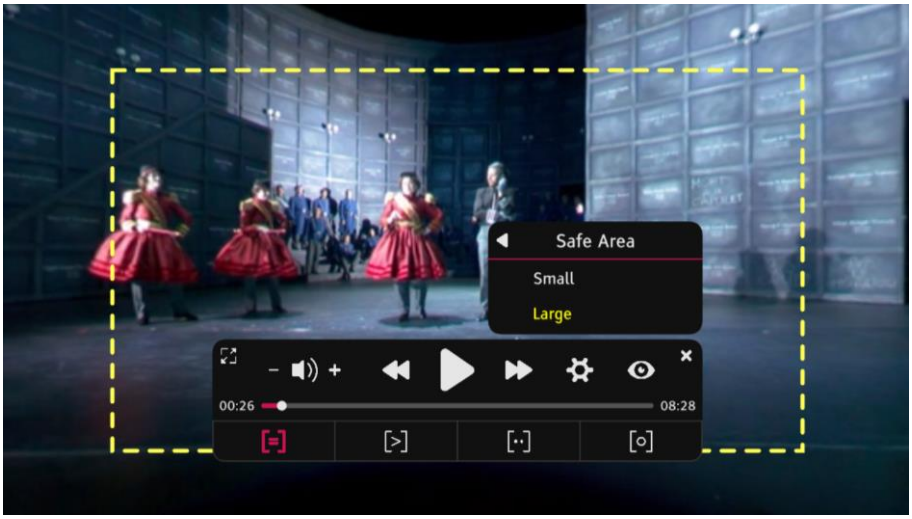


Figure 44- Visual feedback (yellow dotted line) once setting the size of the safe area or CFoV

7.5.6. Guiding Methods

As discussed, subtitles are always visible and centred in the FoV in the ImAc player. In order to assist users in identifying the associated speaker/s, different guiding mechanisms have been implemented (see also Figure 22 for the ImAc portal):

a) None: no guiding mechanism is used.

b) Arrows (see Figure 45): arrows on the right / left part of the subtitle frame indicate that the associated speaker is on the right / left of the current FoV. When the speaker in the FoV, the arrows disappear. The design of the arrows has slightly changed compared to the version described in the first iteration of this deliverable (see Figure 46, left side: previous design at the top, current design at the bottom), and a flashing effect has been added to them too, based on the feedback received from the pilot 1 actions.

c) Radar (see Figure 46): a radar indicates where the speaker are in the 360º area, by representing the speaker with the same colour as the subtitle frames, and representing the current FoV as a portion of the 360º area. The design of the radar has slightly changed compared to the version described in the first iteration of this deliverable (see Figure 46, right side: previous design on the left, current design on the right), based on the feedback received from the pilot 1 actions.

c) Auto-Positioning: it consists of automatically adjusting the FoV based on the speakers' positions. When using this presentation mode, the subtitle editor can explicitly indicate for each subtitle frame being created / edited if the auto-positioning mechanism must be applied for that frame or not.

It is assumed that users can have preferences regarding the usage of these presentation modes, and that their appropriateness will be also determined by the type of video being consumed. Therefore, it was decided to include the three of them in the player, and let the user choose the preferred one in each situation.

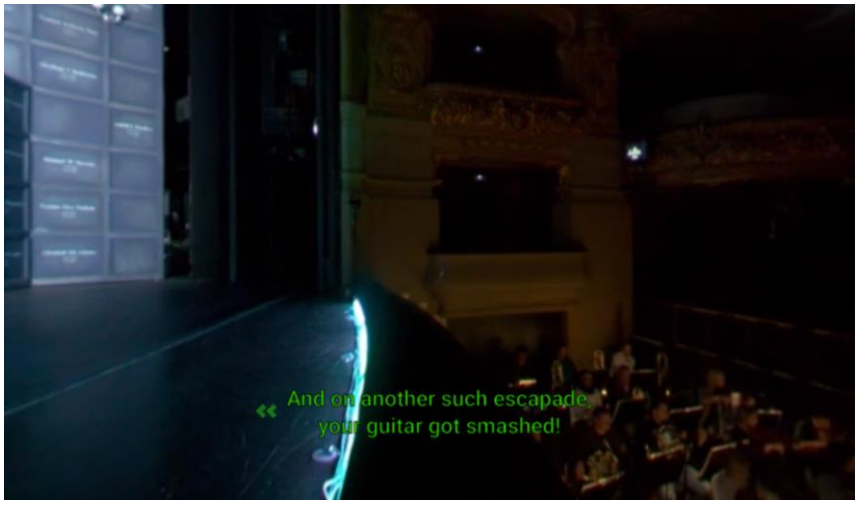


Figure 45- Arrows as a guiding method to inform about the position of the speaker

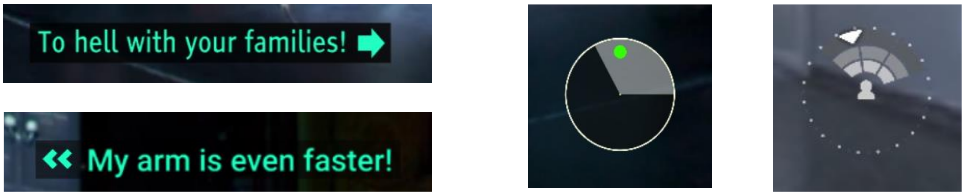


Figure 46- Design improvement for the visual guiding methods

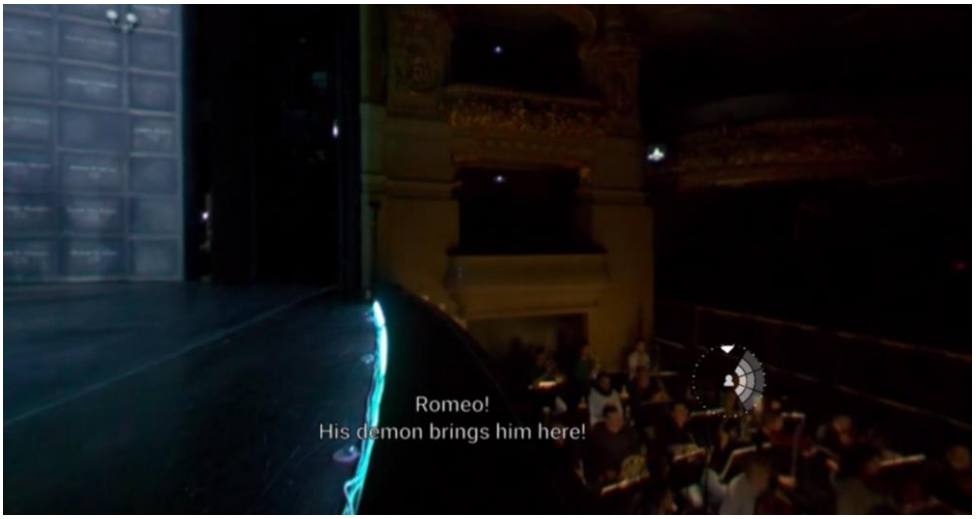


Figure 47- Radar as the guiding method to inform about the position of the speaker

7.6. Presentation of Audio Subtitles (AST)

The menu options and settings for AST presentation have been already introduced when describing the ImAc portal, and can be also set via the player menu by accessing the AST Settings item in the General Settings menu (see Figure 38).

7.7. Presentation of Audio Description (AD)

The menu options and settings for AD presentation have been already introduced when describing the ImAc portal, and can be also set via the player menu by accessing the AD Settings item in the General Settings menu (see Figure 38).

7.8. Presentation of Sign Language (SL)

The menu options and settings for SL presentation have been already introduced when describing the ImAc portal, and can be also set via the player menu by accessing the SL Settings item in the General Settings menu (see Figure 38). In such a case, the setting for the Safe Area and Indicators, introduced for ST, also apply.

Figure 48 shows an example of the presentation of a SL video with arrows as a guiding mechanism. In all presentation modes, the SL video is always kept in the same position of the FoV. In this context, the benefits of two addition presentation features are being evaluated. The first one consists of including text and/or emojis below the SL video window to better identify the active speaker. The first one consists of having the possibility of dynamically hiding / showing the SL video depending on if there is signer activity or not. Prototypes with these

features have been implemented, so if they are proven to provide benefits, they will be adopted in a future release of the player.

Figure 48- Presentation of SL video with arrows as a guiding mechanism

7.9. General Settings

Apart from the settings and personalization options for each access service (ST, AST, AD, SL) and the ones common for the access services, already introduced, the player includes a menu for general settings (Figure 49). Next, the available features and options are detailed.

7.9.1. Language Change

Although the (initial) language for the UI can be selected in the initial webpage (ImAc portal), it can be also dynamically set during media consumption in the menu settings (see Figure 49, accessed through the screen in Figure 38). The available languages are: Catalan, Spanish, German and English.

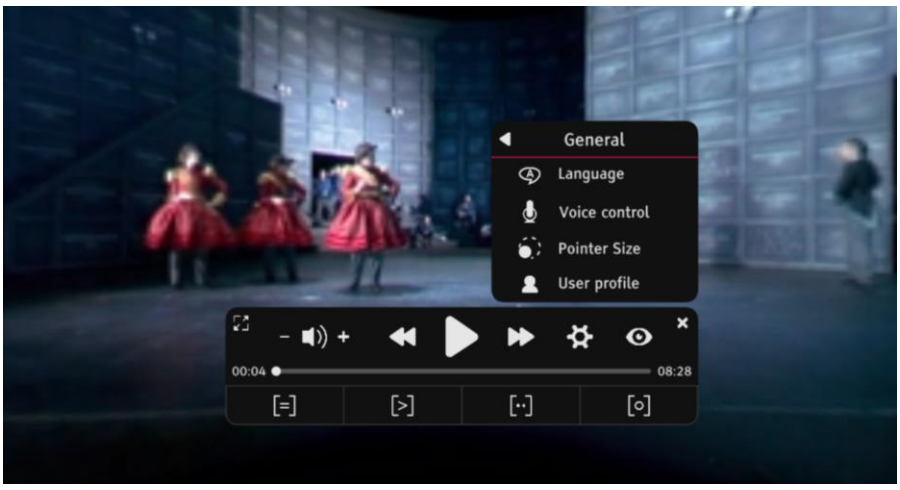


Figure 49- UI Language Setting

7.9.2. Voice Control

Although the voice control feature can be enabled via the ImAc portal, it can be also enabled and disabled in the menu settings (see Figure 49).

7.9.3. Pointer Size

Although the Pointer Size feature can be enabled via the ImAc portal, it can be also enabled and disabled in the menu settings (see Figure 49).

7.9.4. User's Profile / Preferences Saving

In the menu settings, it is also possible to store the user's settings / preferences (see Figure 49), so they will be enabled in future media sessions by default. It is done by saving the configuration using browser cookies. It is also envisioned to define different User's Profiles and to be able to select them in the menu settings in future stages of the project.

7.10. Assistive Mechanisms

Different mechanisms to assist the users in a more effective and comfortable usage of the ImAc player are supported, which are briefly explained next.

7.10.1. Enlarged Version of the UI

As described before, the player includes an enlarged version of the UI. Although the UI type to be displayed can be selected in the ImAc portal, it can be also dynamically changed during consumption via the menu, by clicking on icon at the top left of the menu that has an enlargement action form (see Figure 38). The result can be seen in Figure 50.

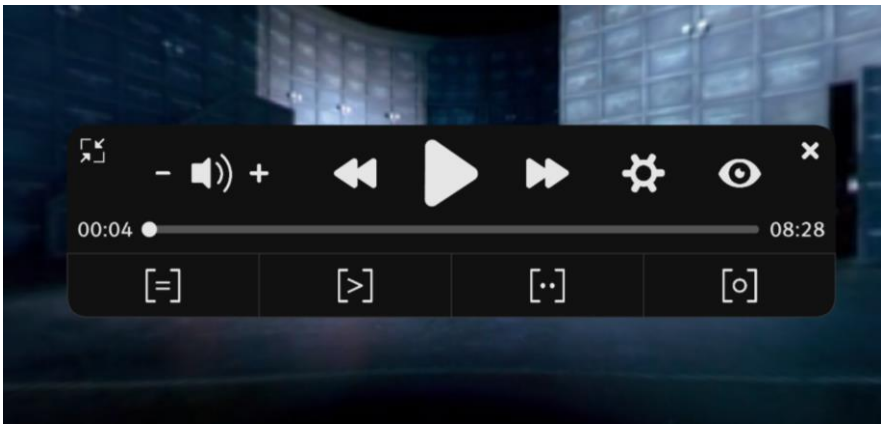


Figure 50- Enlarged version of the UI for better accessibility

Switching back to the traditional (lower size) version of UI can be done by clicking on the icon at the same position, which in this case has a compacting action form.

When using the enlarge version of the UI, and menus or sub-menus are selected, they are opened in the centre of the screen, as the unique displayed sub-menus with enlarged size (thus without the main menu items), with the corresponding navigation controls, as can be seen in Figures 51 and 52.

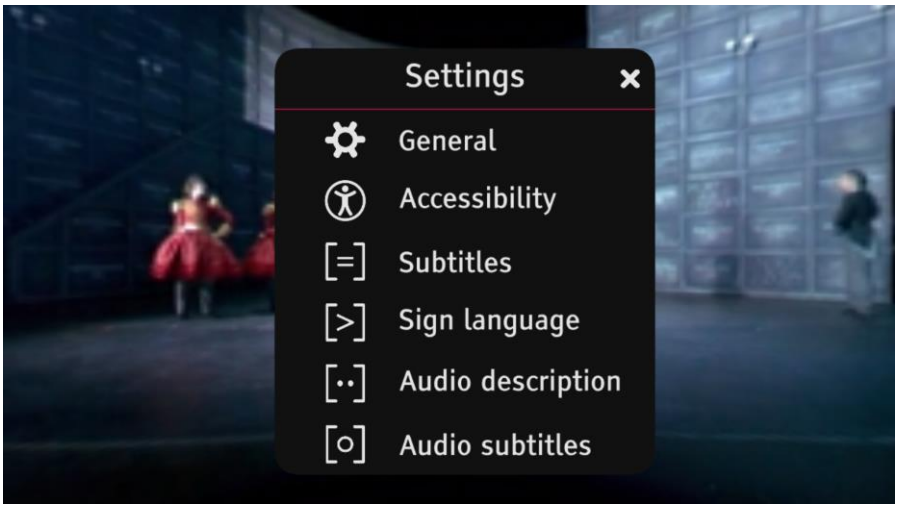


Figure 51- Enlarged version of the UI: General Settings menu

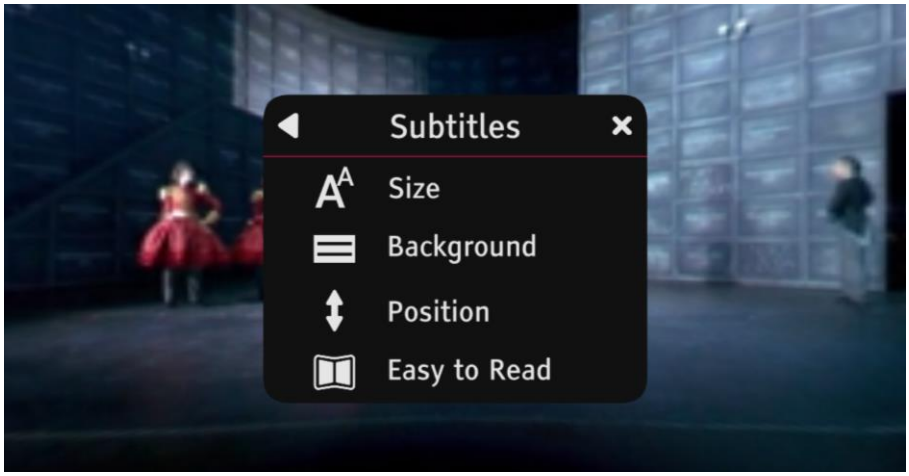


Figure 52- Enlarged version of the UI: ST Settings sub-menu

7.10.2. Visual Indicators

Visual Indicators or Guiding Methods, such as a radar or arrows, are added to help the users in finding the target speaker in the 360º area when ST and SL services are activated. They have been presented in Section 7.5.6.

7.10.3. Voice Control

The voice control feature allows users to interact with the player by executing voice commands, and getting spoken feedback to their execution. This is especially beneficial for low-sighted and blind users.

7.11. Interaction with the Player

The ImAc player supports different interaction modes. When using a PC for the consumption of videos, interaction with the player is done via the mouse and/or the keyboard. When using a tablet or smartphone, interaction with the player is done via the touch screen, although it is also possible to navigate around the 360º area by using the gyroscope sensor. When using a VR-enabled device, the head movement sensors are used to navigate around the 360º area and to move the cursor. The controllers of the VR devices can also be used to move the cursor, while their touchpad and buttons can be used to select and/or navigate between menus of the UI. Finally, voice control is also provided, as described above.

7.12. Multi-Screen Scenarios

When using the web-based version of the multi-screen scenarios, an association mechanism between the companion screens and the main screen will be provided (e.g. via a QR or pin code). The current implementation associates all the consumption devices that are connected via the same network and that have chosen the same languages.

When using the HbbTV version of the multi-screen scenarios, a notification will appear in the main screen, and the users will need to perform any action, like pressing the “Red Button” on the remote to launch the multi-screen app on their companion screens, which need to have connected to the same WiFi as the connected TV.

The concrete solutions and steps to enjoy these scenarios will be detailed once completing their implementation, planned for upcoming months.

8. CONCLUSIONS

The end-to-end ImAc platform is comprised of different parts where production, editing, management, preparation, delivery and consumption of contents take place. This deliverable has provided a comprehensive overview of the ImAc player, by explaining its different pieces of software, how to configure and use it, how the different features are provided/enabled, and the different screens, UIs and controls, and interaction modalities of the player, providing the targeted functionalities and personalization options.

The ImAc player is a core component of the ImAc platform, as it is the interface through which end-users will consume the available immersive and accessibility contents in an interactive and customized manner.

The development of the player has been driven by the user-centric methodology followed in the project, which has provided an extensive list of requirements to be met. It has been quite challenging to provide efficient solutions for all these requirements, especially given the diversity and combination of media contents being considered, the heterogeneity of the media consumption devices to be supported, together with their limited capabilities and resources, and the diverse needs and/or preference of the target end-users, etc. The UI design and features provided by the player have been refined based on the results from pilot 1 actions and extended to accommodate both the new requirements derived from the pilot 1 actions and those already envisioned to be implemented for pilot 2. At the time of writing this deliverable, the results from the pre-pilot phase 2 seem promising, and no major changes are expected to be requested towards pilot 2. However, it was been decided to enlarge the WP3 cycle in order to be able to take into account the feedback and results from pilot 2 actions in order to refine the player, including its graphical design and the features it provides, with the final goal of an improved immersion and accessibility.

Finally, it is also noteworthy to reflect that the features to monitor and register the users' activity and behaviour, e.g. via Google Analytics, have not been fully implemented yet at the time of writing this deliverable, but will be provided on time for the execution of pilot 2 actions. Google Analytics tools and services are quite common nowadays, and documentation on how to integrate them with media services and applications can be found quite easily, so readers can get an idea of the necessary steps for their integration.

9. REFERENCES

[1] M. Montagud, F. Boronat, J. Belda, D. Marfil, "Use of Web Components to Develop Interactive, Customizable and Multi-Device Video Consumption Platforms", Communications in Computer and Information Science (CCIS), volume 605, Springer-Verlag, April 2016.

[2] D. Gómez, J. A. Núñez, I. Fraile, M. Montagud, S. Fernández, “TiCMP: A lightweight and efficient Tiled Cubemap projection strategy for Immersive Videos in Web-based players”, ACM NOSSDAV’18, Amsterdam, June 2018.

[3] M. Montagud, I. Fraile, J. A. Núñez, S. Fernández, “ImAc: Enabling Immersive, Accessible and Personalized Media Experiences”, ACM TVX 2018, Seoul (South Korea), June 2018.

[4] I. Fraile, D. Gómez, J. A. Núñez, M. Montagud, S. Fernández, “Personalized and Immersive Presentation of Video, Audio and Subtitles in 360º Environments: An Opera Use Case”, ACM TVX 2018, Seoul (South Korea), June 2018.

[5] F. Boronat, D. Marfil, M. Montagud, J. Pastor, “HbbTV-Compliant Platform for Hybrid Media Delivery and Synchronization on Single and Multi-Device Scenarios”, IEEE Transactions on Broadcasting, 64(3), pp. 721-746, 2018

[6] M. Montagud, P. Cesar, J. Jansen, F. Boronat (Eds.), “Handbook on Multimedia Synchronization” (23 chapters), Springer-Verlag, ISBN 978-3-319-65840-7, 2018.

[7] M. Montagud, E. Meyerson, I. Fraile, S. Fernández, "Modular Testbed for KPI Monitoring in Omnidirectional Video Streaming Scenarios", The Eleventh International Conference on Advances in Multimedia (MMEDIA 2019), València (Spain), March 2019.

[8] B. Agulló, M. Montagud, I. Fraile, “Making interaction with virtual reality accessible: rendering and guiding methods for subtitles", AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing), Accepted for Publication in 2019.

[9] M. Montagud, I. Fraile, E. Meyerson, M. Genís, S. Fernández, “ImAc Player: Enabling a Personalized Consumption of Accessible Immersive Content”, ACM TVX 2019, June, Manchester (UK).

ANNEX I – RELEASED VERSIONS OF THE IMAC PLAYER

The following table lists and details the features of the released versions of the ImAc player at the date of submission of this deliverable.

Table I.I. Released Versions of the ImAc player

Version (Release Date)	URL	Created for…	Contents	Features [Types of Changes]
v1 (18/7/18) (Improvements released on 24/7/18, and on 10/8/18)	http://84.88.32.46/imacplayerv1/	Initial Testing / Feedback and for IFA demo	video 1: Rapzember video 2: Opera scene 1 video 3: Abendschau (RBB Video) video 4: I, Philip	[Added] Support for 360º video [Added] (Only) Low-sighted UI [Added] Activation and Deactivation of Accessibility Services via toggle buttons [Added] Menu is open via a (green) button in a fixed position [Added] Playback controls with visual feedback [Added] Presentation of Subtitles: with language and personalization settings (position, background…) [Added] Subtitles with guiding mechanisms, but using a static compass [Added] Three sizes for the Comfortable Field of View (CFoV)

v1.5 (24/7/18)	http://84.88.32.46/imac2	Pre-pilot tests, for SL and ST Preparation of different test conditions with different options for the Comfortable Field of Views (CFoV) and guiding mechanisms for ST and SL. Video snippets with integrated still images to guide user through the test conditions	video 1: Rapzember video 2: Opera scene 1	[Fixed] Some fixed bugs from v1 [Added] VR and tablet modes [Added] Settings on initial screen: Select content Select language Select device
v2 (10/8/18) (Improvements released on 24/8/18 and on 7/9/18)	http://84.88.32.46/imacplayerv2/	Testing improvements and new features	video 1: Rapzember video 2: Opera scene 1 video 3: Abendschau (RBB Video)	[Fixed] Some fixed bugs from v1.5 [Added] Support for DASH [Added] Support for HMD controllers [Added] First version of the Traditional UI (open the menu via an always-visible wheel settings icon) [Changed] Improvements to the "Low-Sighted" UI [Changed] Open the Menu by looking down for a period [Added] Multi-language support [Added] Support for SL video, with all personalization settings and guiding mechanisms (for Opera 1 scene)

v3 (10/8/18)	http://84.88.32.46/imacplayerv3/	Showing to ARTE the "I Philip" video integrated into the ImAc player, to ask for permission for its use in the project	video 4: I, Philip	Same as v2
Test Radar (18/9/18)	http://84.88.32.46/imacradar/	Test with first version of radar, before Pilot 1	video 1: Rapzember video 2: Opera scene 1 video 3: Abendschau (RBB Video) video 4: I, Philip	[Changed] Dynamic Radar (instead of a static compass)
Pilot 1 - Part 1 (21/9/18)	http://84.88.32.46/imacpilot1_part1/	Pilot Phase 1 Video snippets with different test conditions, using arrow and radar as the guiding mechanism for subtitles	video 1: I, Philip video 2: Opera scene 1 video 3: Opera scene 2	[Fixed] Some fixed bugs from v2/v3 (from Issue tracking table) [Changed] Refined Radar [Changed] Support for subtitles frames without an associated angle [Changed] Improvements to the Traditional UI (opened by looking down) [Added] Immediate feedback once updating controls / settings in the traditional UI
Pilot 1 - Part 2 (21/9/18)	http://84.88.32.46/imacpilot1_part2/	Pilot Phase 2	video 1: Opera scene 1 video 2: Desconcert 1 video 3: Desconcert 2	Same as for Pilot 1 - Part 1

		Usability tests for the traditional UI on tablet and on HMD		
v4 (10/10/18)	http://84.88.32.46/imacplayerv4/	Testing Improvements and New Features	video 1: Opera scene 1 video 2: Opera scene 2 video 3: Desconcert 1 video 4: Desconcert 2 video 5: Abendschau (RBB Video)	[Fixed] Some fixed bugs from Pilot 1 version (from Issue tracking table) [Added] New initial page with all available videos (except "I Phillip") and with the option to choose the UI and the language. [Added] Added browser "Back" button to go back to the initial webpage from the player webpage, without having to load again the webpage. [Added] Generation of a dedicated URL for the video player with a specific video (useful for embedding purposes). [Added] Changes in the controls and settings (position, size, presentation modes…) are reflected immediately, even though the menu is opened.
v5 (20/11/18)	http://84.88.32.46/imacplayerv5/	Testing Improvements and New Features – Presentation at ICT 2018	video 1: Opera scene 1 video 2: Opera scene 2 video 3: Desconcert 1 video 4: Desconcert 2 video 5: Abendschau (RBB Video)	[Fixed] Some fixed bugs from v4 (from Issue tracking table) [Added] New software architecture for interacting with the UIs (higher performance efficiency) [Added] Only allow selection of available languages for accessibility contents.

				[Added] Dynamic radar for SL videos. [Added] A new settings option to be able to switch between traditional and low-sighted (enhanced-accessibility) UI during the consumption experience. [Added] The player shows a dotted line for the display area when a new area is selected (to better understand this feature) [Added] A banner appears when the player starts, explaining how the menu can be opened. [Added] Progress bar with seeking option.
v5.1	http://84.88.32.46/UAB-test2/	Testing subtitle rendering and presentation modes	video 0: Acclimation video 1: Ryot – Holy Land 4 video 2: Ryot – Holy Land 5 video 3: Ryot – Guelaguetza	[Added/Test] Subtitles attached to the video (for comparison) [Changed/Test] Lower transparency to the background box [Changed/Test] Transition speed for auto-positioning adjusted based on state-of-the-art studies
v5.2 (11/2/19)	http://84.88.32.46/UAB-test3	Testing subtitle rendering and presentation modes	Not Important, this release was to test new features	[Changed/Test] Opening the menu via a "long click" on the controller [Added] Navigate around the 360º area, by using the mouse and keyboard on the PC. [Changed/Test] Subtitlles are always shown horizontal (i.e. no inclination based on the user's perspective)

v5.3 (25/2/19)	http://84.88.32.46/UAB-test4	Testing subtitle rendering and presentation modes (v2)	video 0: Acclimation video 1: Ryot – Holy Land 1 video 2: Ryot – Holy Land 2 video 3: I Philip	[Changed/Test] New version of the arrow with transparency effect [Changed/Test] New version of the arrow
v6.0 (13/2/19)	Test version in GPAC Server: https://imac.gpac-licensing.com/playertest/ Stable version in GPAC Server: https://imac.gpac-licensing.com/player/	Moved to GPAC Server.	All videos available for /player All videos, except "I Philip" for /playertest	[Added] AD functionalities (language, presentation mode, volume) [Added] Add: AST functionalities (language, easy to read, volume) [Added] A testbed to monitor and register Key Performance Indicators (KPIs) has been integrated [Changed] The ST are signalized within the MPD at the AdaptationSet level [Changed] The SL video are signalized within the MPD at the AdaptationSet level [Changed] The AD tracks are signalized within the MPD at the AdaptationSet level [Changed] The AST tracks are signalized within the MPD at the AdaptationSet level [Added] Support for multiple periods in the MPD [Added] Preview Function [Added] Multi-Options Menu is closed once clicking outside of it

				[Changed] The personalized positioning of the SL video has been changed from Top/Bottom to Right/Left, based on requirements [Changed] When using HMD, the menu is opened in front of the user [Changed] The SL video is positioned at top, when the ST are positioned at top [Fixed] Error when the JSON file has an empty SL value [Fixed] Auto-play of the SL video when the Enhanced-Accessibility menu is opened [Fixed] Default options of SL video not highlighted in yellow [Fixed] Incorrect values for the SL video area [Fixed] Performance issue with the Oculus Rift controller.
v7 (9/5/19)	Test release: http://84.88.32.46/imacplayerv7/ Test version in GPAC Server: https://imac.gpac-licensing.com/playertest/ Stable version in GPAC Server:	New design of the ImAc portal and player	All videos available for /player All videos, except "I Philip" for /playertest	[Added] New responsive and more attractive design of the ImAc portal and player UI [Add] New metadata about the contents is added to the JSON file and the portal [Added] Multi-language support and selection [Added] Search / filtering features in the portal [Added] Initial settings and saving them in the portal [Added] Conection to the voice control server (via http://84.88.32.46/imacplayerv7/)

	https://imac.gpac-licensing.com/player/			[Added] Toggle button to allow showing / hiding the content info. Hiding the content info can be helpful in low size screens, being able to watch the catalogue of contents in a more comfortable manner [Added] Links to the partners websites, to the project website and its Social Media channels. [Added] Personalization options for the size of the SL video [Changed] New banner to indicate how to open the menu. [Changed] The menu can also be opened by performing (5) consecutive clicks on the controller in use. [Changed] The player can be controlled via voice (if connected to the voice server) [Changed] Integration of the new menu, including its enhanced-accessibility version. [Changed] The initial settings in the ImAc portal are also highlighted in yellow when opening the menu. [Changed] The player is paused when the pre-view feature is opened [Changed] Support for all test conditions (rendering/presentation modes) considered in the pre-pilots (phase 2), if desired for any content. [Changed] Detects the available access services for each video, and only shows its associated controls

				[Known Issue] Subtitles are occluded under the player menu. [Known Issue] The menu bar does not work in drag&drop mode, just by clicking on the target position. [Known Issue] No button for going back to the poral while playing is available (but the back button of the browser can be used)
V7.1 (3/5/19 for German version) (10/5/19 for Catalan version)	[German] ST: https://imac.gpac-licensing.com/pilot-test1/ ST UI test: https://imac.gpac-licensing.com/test-ui-ger/ SL: https://imac.gpac-licensing.com/pilot-test2/ https://imac.gpac-licensing.com/pilot-test2v2/	URL for the ST and SL pre-pilot tests, in German and in Catalan	I Philip for ST tests All contents for ST UI test	[New] Presentation modes for subtitles considered for the pre-pilot tests phase 2

	[Catalan] ST: https://imac.gpac-licensing.com/pilot-test1cat/ ST UI test: https://imac.gpac-licensing.com/test-ui-cat/			
V7.2 (22/5/19 for German version) (4/6/19 for Catalan version)	[English] AD Presentation Modes + Audio Introduction: https://imac.gpac-licensing.com/test-ad-2-eng/ AD + Interaction (via menu and voice control): http://84.88.32.46/test-ad-1-eng AST: https://imac.gpac-	URL for the AD and AST pre-pilot tests, in English and in Catalan	AD: video 1: Ryot – Holy Land 1 video 2: Ryot – Holy Land 2 video 3: Ryot – Holy Land 3 [with the 3 AD presentation modes] video 4: Ryot – Holy Land 4 AST: video 5: Opera clop 1	[Added] Contents with the considered presentation modes for AD and AST

	licensing.com/test-ast-eng/ [Catalan] AD Presentation Modes + Audio Introduction https://imac.gpac-licensing.com/test-ad-2-cat/ AD + Interaction (via menu and voice control): http://84.88.32.46/test-ad-1-cat AST: https://imac.gpac-licensing.com/test-ast-cat/			
V7.3 (6/6/19)	http://84.88.32.46/pac3/	URL for web-based multi-screen demo	video 1: Opera clop 1 video 2: I Philip	[Added] First released implementation of the multi-screen demo for fully web based scenarios

ANNEX II – CONTENT FORMATS AND METADATA MODEL

This Annex firstly provides in different tables the content formats, resolutions and versions for each media asset being considered in ImAc. Then, it details the metadata model that has been defined to provide the information about the available contents in the ImAc portal. Finally, it details the proposed extensions to, and format of, the MPD to signalize the considered access services.

Table II.I. Video Assets being considered in ImAc

No	Media Asset	Projection type	Encoding parameters	Comments	Cardinality
1	Main 360 video	Equirectangular (or Cubemap)	AVC, MP4, fps and bitrate configurable	Quality Level 1 (720p = 1280 x 720)	1
2	Main 360 video	Equirectangular (or Cubemap)	"	Quality Level 2 (1080p = 1920 x 1080 pixels)	0..1
3	Main 360 video	Equirectangular (or Cubemap)	"	Quality Level 3 (1440p = 2560 x 1440 pixels)	0..1
…	Main 360 video	Equirectangular (or Cubemap)	"	Quality Level 4 (4K or 2160p = 3840 x 2160 pixels)	0..1
	Signer stream continuous	Plain, 2D	4:2:0 chroma sampling, 8 bit colors, and use of High level 3.1 profile, progressive	Only one, continuous or multi-period signer can be present.	0..1
	Signer stream multi-period 1	Plain, 2D	"	Only one, continuous or multi-period signer can be present.	0..1
	Signer stream multi-period 2	Plain, 2D	"		0..1
	... Signer stream multi-period n	Plain, 2D	"		0..1

Table II.II. Audio Assets being considered in ImAc

No	Format	Audio mix	Encoding parameters	AD Description	Cardinality
1	Stereo	Main mix	AAC, 2 channels	-	0..1
2	Stereo	Main mix + AD	"	Pos: center, volume: low	0..1
3	Stereo	Main mix + AD	"	Pos: center, volume: medium	0..1
4	Stereo	Main mix + AD	"	Pos: center, volume: high	0..1
5	FOA	Main mix	AAC, 4 channels	-	0..1
6	FOA	Main mix + AD	"	Pos: Classic, volume: low	0..1
7	FOA	Main mix + AD	"	Pos: Classic, volume: medium	0..1
8	FOA	Main mix + AD	"	Pos: Classic, volume: high	0..1
9	FOA	Main mix + AD	"	Pos: Static, volume: low	0..1
10	FOA	Main mix + AD	"	Pos: Static, volume: medium	0..1
11	FOA	Main mix + AD	"	Pos: Static, volume: high	0..1
12	FOA	Main mix + AD	"	Pos: Dynamic, volume: low	0..1
13	FOA	Main mix + AD	"	Pos: Dynamic, volume: medium	0..1
14	FOA	Main mix + AD	"	Pos: Dynamic, volume: high	0..1
15	5.1	Main mix	..	..	0..1

16	5.1	Main mix + AD	..	..	0..1
17	Stereo	AST	AAC, 2 channels	Parallel audio track, so no fixed volume level applies	0..1
18	FOA	AST	AAC, 4 channels	Parallel audio track, so no fixed volume level applies. Pos: Static	0..1
19	FOA	AST	AAC, 4 channels	Parallel audio track, so no fixed volume level applies. Pos: Dynamic	0..1

Table II.III. Subtitle Assets being considered in ImAc

No	**Format**	**Language**	**Encoding**	**Target**	**Cardinality**
1	IMSC	Spanish	Plain XML file	Hard of hearing	0..1
2	IMSC	German	Plain XML file	Hard of hearing	0..1
3	IMSC	English	Plain XML file	Hard of hearing	0..1
4	IMSC	Catalan	Plain XML file	Hard of hearing	0..1
5	IMSC	Spanish	Plain XML file	Hard of hearing + Easy to Read	0..1
6	IMSC	German	Plain XML file	Hard of hearing + Easy to Read	0..1
7	IMSC	English	Plain XML file	Hard of hearing + Easy to Read	0..1
8	IMSC	Catalan	Plain XML file	Hard of hearing + Easy to Read	0..1

METADATA MODEL - JSON FILE LISTING / DESCRIBING THE AVAILABLE CONTENTS

The ImAc portal needs to provide a carrousel or catalogue of the available immersive contents (360º clips) together with some extra information about these clips (e.g. language, duration…) and, most importantly, the available access service for each of them. This information is provided via a JavaScript Object Notation (JSON) file, with the following structure:

```
{
 "title": "ImAc contents",
 "contents": [
  // Full entry (metadata) for one clip
  {
   "title": "", // Title of the video clip
   "poster": "", // URL of the poster for the video clip
   "thumbnail": "", // URL of the thumbnail for the video clip
   // Description // Synopsys of the video clip in each language
   "descriptionArray": [
   {
    "de": "",
    "ca": "",
    "es": "",
    "en": ""
   }],

 "access": [
  {
   "ST": ["en", "es", "de", "ca"], // values can be any of these: "en", "es", "de", "ca", for any of the access services
   "SL": [],
   "AD": [],
   "AST": []
  }],
  "language": "", // Language of the Main audio
  "duration": "", // Duration of the Clip
```

```
  "url": "../imac_content/pilot_1/OPERALICEU_01/stream.mpd", // URL of the main MPD for that clip
  "audioChannels": “”, // Type of audio channel: stereo (2), Ambisonics (4)
  "projection_type": “”, // Indicates whether the projection of the video is Equirectangular, or Cubemap (and the specific configuration of faces being defined)
  "stereoscopic": “”, // Indicates whether the video is stereoscopic (and the type) or not.
},

// One entry like the above for each clip
{

},
```

METADATA MODEL – PROPOSED EXTENSIONS TO THE MPD

This section indicates how the different immersive and accessibility media elements and their characteristics / parameters are signalized within the MPD for each clip.

Main Video and Audio Tracks

The availability of the main video and audio tracks is signalized at the AdaptationSet level, as done typically for 360º video and audio tracks.

Regarding the video track, the language is specified via the *lang* attribute at the AdaptationSet level, and the type of asset is identified via the *value* attribute (value=”main”) within the Role element.

Currently, the signalization of parameters for the projection type (Equirectangular or CubeMap, together with specific CubeMap configurations proposed in the project) and the Stereoscopic format are signalized in the JSON file previously described. As future work, their signalization at the MPD level will be proposed for standardization.

Regarding the audio track, the language is specified via the *lang* attribute at the AdaptationSet level, the type of asset is identified via the *value* attribute (value=”audio”) within the Role element, and the specific type of audio track (Stereo or Ambisonics) is specific via the *value* attribute (e.g. value=”2” for stereo, value=”4” for FOA) within the AudioChannelConfiguration element.

As future work, the signalization of OBA and its related parameters, can be proposed for standardization.

```
<?xml version="1.0"?>
<MPD…>
    <ProgramInformation…>
        <Title>manifest.mpd generated by ImAc consortium</Title>
    </ProgramInformation>
    …
    <Period duration=…>
        <!--360º video-->
        <AdaptationSet lang="" maxFrameRate="" maxHeight="" maxWidth="" par="" segmentAlignment=" " startWithSAP="" >
            <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" >
            <!-- one Representation for each video quality -->
            <Representation bandwidth="" codecs=" " frameRate="" id="" mimeType="video/mp4" sar="" height="" width="" >
                <SegmentTemplate duration="" initialization=" " media=”” startNumber="" timescale ="">
            </Representation>
         …
        </AdaptationSet>

        <!—main audio track-->
        <AdaptationSet lang="" segmentAlignment=" " startWithSAP="" >
            <Role schemeIdUri="urn:mpeg:dash:role:2011" value="audio" >
            <!-- one Representation for each audio quality -->
            <SegmentTemplate duration="" initialization=" " media=”” startNumber="" timescale =" >
            <Representation audioSamplingRate="" bandwidth="" codecs="" id="" mimeType="audio/mp4" >
                <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="" />
            </Representation>
        …
        </AdaptationSet>

    </Period>
    …
</MPD>
```

Sign Language Tracks

The availability of the Sign Language video tracks is signalized at the AdaptationSet level, specifying the language via the *lang* attribute and the type of asset via the *value* attribute (value=”sign”) within the Role element. In such a case, the latitude and longitude angles that indicate the speaker’s position within the 360º area for each segment are specified within the SegmentTemplate element. The specific position and size of the sign language window in the 360º scene can be personalized by using controls of the ImAc player.

An important issue arises regarding the dynamic presentation of the Sign Language video tracks in a picture-in-picture mode, as only one video and audio track can be presented simultaneously by dash.js. Therefore, if the Sign Language track is provided as a DASH stream, then: Option 1) the code of dash.js needs to be modified in order to be able to initialize another parallel stream from the SegmentTemplate element; or Option 2) the URL of an MPD for the Sign Language track needs to be provided in order to create another instance of dash.js / video element for this parallel track, thus having internal MPDs within the main MPD. Currently, Option 2 is implemented.

As future work, efficient solutions to represent the angles information, to define non-continuous Sign Language video streams (e.g. as a collection of segments) and to include text information to be displayed providing information about the target speaker (e.g. the name), will be proposed for standardization. In addition, the appropriateness of adopting the discussed Options 1 and 2 needs to be discussed in standardization bodies.

```
<AdaptationSet group="" lang="" maxFrameRate="" maxHeight="" maxWidth="" par="" segmentAlignment=" " startWithSAP="">
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="sign" />

        <Representation bandwidth="" codecs="" frameRate="" height="" id="" mimeType=" " sar="" width="" startWithSAP=>
                Option 1:
                <SegmentTemplate duration="" initialization=" " media=" " startNumber="" timescale="" latitude=”” longitude=”” />
                Option 2:
                <SegmentList timescale="1000" duration="27070" >
                        <SegmentURL media="URL_of_MPD.mpd" />
                </SegmentList>
        </Representation>

</AdaptationSet>
```

Subtitle Tracks

The availability of the subtitle tracks is signalized at the AdaptationSet level, specifying the language via the *lang* attribute and the type of asset via the *value* attributes within the Role and Accessibility elements. Within the Role element, the value type can be *main* for SDH and *alternate* for translation subtitles. When the value is *main*, then the *value="2"* is added to the Accessibility element. No Accessibility element is currently present for the translation subtitles. Likewise, a new *e2r* attribute has been defined to indicate whether the subtitles are of Easy-to-Read modality (modality specific in ImAc) or not.

In such a case, the latitude and longitude angles that indicate the speaker's position within the 360º area, the styling effects and the vertical position (e.g. top / bottom) are specified within the IMSC file. However, all these values can be altered in the player via personalization controls.

As future work, the specified extensions to the IMSC file format and to indicate the availability of Easy-to-Read subtitles will be proposed for standardization.

```
<AdaptationSet contentType="text" mimeType="application/ttml+xml" segmentAlignment="true" lang=" ">
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value=" " e2r=" "/>
        <Accessibility schemeIdUri="urn:tva:metadata:cs:AudioPurposeCS:2007" value=""/>
        <Representation id=" " bandwidth="">
                <BaseURL>baseURL.xml</BaseURL>
        </Representation>
</AdaptationSet>
```

Audio Description Tracks

The availability of the Audio Description tracks is signalized via a combination of two methods: 1) the *value* attribute within the Accessibility element, which is set to *1* (if no Audio Description track is available, then the Accessibility element is not present); and 2) the *value* attribute within the Role element, which is set to *alternate* (value also typically used for clean audio). The audio language is indicated via the *lang* attribute with the AdaptationSet element (one element per language). The type of audio is indicated in the AudioChannelConfiguration element, as for the main audio track. Finally, the Audio Description mode and gain are indicated in the newly proposed *mode* and *gain* attributes in the Representation element. The possible values for *mode* are [Classic, Static, Dynamic] and for *gain* are [Low, Medium, High]. Note that an *imac* namespace could be used for these new attributes (imac:mode, imac:gain).

As future work, the specified extensions to the MPD to signalize the Audio Description presentation modes will be proposed for standardization.

```
<AdaptationSet contentType=”"" lang="">
        <Accessibility schemeIdUri="urn:tva:metadata:cs:AudioPurposeCS:2007" value="1" />
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="alternate"/>
        <Representation mimeType="audio/mp4" audioSamplingRate="48000" bandwidth="320000" id="” mode="" gain="">
                <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="" />
                Option 1 (AAC File)
                <BaseURL>ad_file.aac</BaseURL>
                <SegmentBase indexRange="0-2000"/>
                Option 2 (DASH Stream)
                <SegmentTemplate media=" " timescale="" startNumber="" initialization=""  />
        </Representation>
        …
</AdaptationSet>

</AdaptationSet>
```

Audio Subtitle Tracks

Unlike the Audio Description tracks, that are alternate tracks muxed with the main audio track, the Audio Subtitles tracks are parallel tracks to be played out in combination with the main audio tracks, when enabled. Likewise, Audio Subtitles are not commonly provided as an access services, so no standard signalling methods for them exist yet. In ImAc, Audio Subtitles are provided via Text-To-Speech (TTS) tracks with synthetized and audio/voice, and also considering audio spatiality. Therefore, the proposed signalling for Audio Subtitles is quite similar than for Audio Description, by defining a new value attribute (value=”ast”), and by eliminating the for the *gain* attribute. It is because the volume level can be adjusted as desired for the Audio Subtitles track, as it is not a muxed track with the main audio.

However, a similar issue as for the Sign Language video tracks arises regarding the Audio Subtitles tracks, as only one video and audio track can be presented simultaneously by dash.js. Therefore, if the Audio Subtitles track is provided as a DASH stream (Options 1 and 2 below), then: Option 1) the dash.js code needs to be modified in order to be able to initialize another parallel stream from the SegmentTemplate element; or Option 2) the URL of an MPD for the Audio Subtitles track needs to be provided in order to create another instance of dash.js / audio+video element for this parallel track, thus having internal MPDs within the main MPD. Currently, Option 2 is implemented.

As future work, the specified extensions to the MPD to signalize the Audio Subtitles will be proposed for standardization, paying special attention to the value attributes within the Accessibility and Role elements. Also, the appropriateness of adopting the discussed Options 1 and 2 needs to be discussed in standardization bodies.

```
<AdaptationSet contentType="" lang="">
        <Accessibility schemeIdUri="urn:tva:metadata:cs:AudioPurposeCS:2007" value="ast" />
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="ast"/>
        <Representation mimeType="audio/mp4" audioSamplingRate="48000" bandwidth="320000" id="” mode="" gain="">
                <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="" />
                Option 0 (AAC File)
                <BaseURL>ast_file.aac</BaseURL>
                <SegmentBase indexRange="0-2000"/>
                Option 1 (inline DASH Stream)
                <SegmentTemplate media=" " timescale="" startNumber="" initialization="" />
                Option 2 (MPD to external DASH stream)
                <SegmentList timescale="1000" duration="27070" >
                        <SegmentURL media="URL_of_MPD.mpd" />
```

```
                </SegmentList>

            </Representation>
            …
</AdaptationSet>
```

ANNEX III – VOICE COMMANDS AND RESPONSES IN EACH LANGUAGE

The table below provides the lists of intents, voice commands and responses used in the voice control system for the pre-pilot tests, for each language.

Intent	English Command	English Response	Spanish Command	Spanish Response	Catalan Command	Catalan Response
play	ImAc play	Playing video	ImAc reproduce / Imac play	reproduciendo vídeo	ImAc reprodueix / ImAc play	reproducció de vídeo
	play		reproduce / play		reprodueix / play	
	ImAc play video		ImAc reproduce el vídeo		ImAc reprodueix el vídeo	
	play video		reproduce el vídeo		reprodueix el vídeo	
pause	ImAc stop	Stopping video	ImAc para / Imac estop	parando vídeo	ImAc atura / ImAc estop	Aturant vídeo
	stop		para / estop		atura / estop	
	stop video		para el vídeo		atura el vídeo	
	ImAc stop video		ImAc para el vídeo		ImAc atura el vídeo	
	ImAc pause		ImAc pausa		ImAc pausa	
	pause		pausa		pausa	
	pause video		pausa el vídeo		pausa el vídeo	

	ImAc pause video		ImAc pausa el vídeo		ImAc pausa el vídeo	
listCatalog	list videos	1 video. Holy Land Episode 4	lista de vídeos	vídeo 1. Holy Land episodio 4	llista de vídeos	
openIntent	alexa open ImAc		alexa abre ImAc		alexa obre ImAc	
	echo open ImAc		echo abre ImAc		echo obre ImAc	
volume_up	make louder	Ok, How about this?	más alto	vale, ¿así mejor?	més alt	Així està bé?
	turn volume up		sube el volumen		apuja el volum	
	ImAc volume up		ImAc sube el volumen		ImAc apuja el volum	
	volume up		sube el volumen		apuja el volum	
volume_down	make quieter	Ok, How about this?	más bajo	vale, ¿así mejor?	més baix	Així està bé?
	turn volume down		baja el volumen		abaixa el volum	
	ImAc volume down		ImAc baja el volumen		ImAc abaixa el volum	
	volume down		baja el volumen		abaixa el volum	

AD_on	turn on audio description	Audio Description on	activa la audiodescripción	audiodescripción activada	activa l'audiodescripció	Audiodescripció activada
	turn audio description on		activa la audiodescripción		activa l'audiodescripció	
	ImAc audio description on		ImAc activa la audiodescripción		ImAc activa l'audiodescripció	
	audio description on		activa la audiodescripción		activa l'audiodescripció	
AD_off	turn off audio description	Audio Description off	desactiva la audiodescripción	audiodescripción desactivada	desactiva l'audiodescripció	Audiodescripció desactivada
	turn audio description off		desactiva la audiodescripción		desactiva l'audiodescripció	
	ImAc audio description off		ImAc desactiva la audiodescripción		ImAc desactiva l'audiodescripció	
	audio description off		desactiva la audiodescripción		desactiva l'audiodescripció	
return	close player	Closing Video and returning to portal	cierra el reproductor	cerrando el vídeo y volviendo al menú principal	tanca el reproductor	Tancant el vídeo i tornant al menú principal
	ImAc close video		ImAc cierra el vídeo		ImAc tanca el vídeo	
	close video		cierra el vídeo		tanca el vídeo	

	ImAc close		ImAc cierra		ImAc tanca	
	close		cierra		tanca	
help	what can i say	You can ask me to play and pause, turn the volume up and down or turn audio description on and off.	¿qué puedo pedirte?	Puedes pedirme que reproduzca o pause el vídeo, que suba o baje el volumen o que active o desactive la audiodescripción.	què et puc demanar?	Em pots demanar: reproduir i pausar, apujar o abaixar el volum i activar o desactivar l'audiodescripció
	help		ayuda		ajuda	
volume_x	volume	Ok, How about this? Volume {volume}	volumen		volum	
	volume {volume]		volumen {volume}		volum {volume}	
	ImAc volume {volume]		ImAc volumen {volume}		ImAc volum {volume}	
forward	seconds	Ok, Skipped forward {seconds] seconds	segundos	vale, hemos avanzado {seconds} segundos	segons	Hem avançat {seconds} segons
	fast forward {seconds] seconds		avanza {seconds} segundos		avança {seconds} segons	

	fast forward {seconds]		avanza {seconds}		avança'n {seconds}	
	ImAc skip forward {seconds]		ImAc avanza {seconds}		ImAc avança {seconds}	
	ImAc skip forward {seconds] seconds		ImAc avanza {seconds} segundos		ImAc avança {seconds} segons	
	skip forward {seconds]		avanza {seconds}		avança'n {seconds}	
	skip forward {seconds] seconds		avanza {seconds} segundos		avança {seconds} segons	
backward	seconds	Ok, Skipped back {seconds] seconds	segundos	vale, hemos rebobinado {seconds} segundos	segons	Hem rebobinat {seconds} segons
	rewind {seconds] seconds		rebobina {seconds} segundos		rebobina {seconds} segons	
	rewind {seconds]		rebobina {seconds}		rebobina'n {seconds}	
	ImAc skip back {seconds]		ImAc rebobina {seconds}		ImAc rebobina {seconds}	
	ImAc skip back {seconds] seconds		ImAc rebobina {seconds} segundos		ImAc rebobina {seconds} segons	

	skip back {seconds]		rebobina {seconds}		rebobina'n {seconds}	
	skip back {seconds] seconds		rebobina {seconds} segundos		rebobina {seconds} segons	
open	video	Opening Video {video]	vídeo	abriendo el vídeo {video}	vídeo	Obrint el vídeo
	ImAc open {video]		ImAc abre {video}		ImAc obre {video}	
	open {video]		abre {video}		obre {video}	
	open video {video]		abre el vídeo {video}		obre el vídeo {video}	
	ImAc open video {video]		ImAc abre el vídeo {video}		ImAc obre el vídeo {video}	
menu_open	open ImAc menu	Opening Menu	abre el menú de ImAc	abriendo el menú	Obre el menú d'ImAc	Obrint el menú
	open menu		abre el menú		obre el menú	
	ImAc open menu		ImAc abre el menú		ImAc obre el menú	
menu_close	close ImAc menu	Closeing menu	cierra el menú de ImAc	cerrando el menú	Tanca el menú d'ImAc	Tancant el menú
	ImAc close menu		ImAc cierra el menú		ImAc tanca el menú	

	close menu		cierra el menú		Tanca el menú	
didnotunderstand	*	Sorry, I can't understand the command. Please say again.		Lo siento, pero no te he entendido bien. ¿Puedes repetirlo?		Perdona, no t'he entès. Ho pots repetir?

ANNEX IV. USER MANUAL OF IMAC PLAYER (D3.5 V1)

This section describes how to use the ImAc player, after having successfully installed the server and client components to make it running, or by directly accessing the URL on which the player is accessible. This Annex refers to the version of the ImAc player available at the time the first iteration of D3.5 was submitted. It was been kept in the deliverable in order to reflect the evolution of the player, but it has been preferred to move it to an annex to ease the reading flow and comprehension of the section 7 of D3.5.

1.1. Access the Player

The player is web-based, so it can be accessed by just typing an URL via a web browser. At the time of submission of this deliverable, the most recent release of the public version of the player is available here: http://84.88.32.46/imacplayerv5/

The list of the released versions of the ImAc player, with their features, contents and purpose, is provided in Annex I. Note that this list includes versions released later than the version described in this Annex.

1.2. Initial Screen: Selection of Contents and Settings

When typing the URL of the ImAc player, a simple initial page listing the available contents will be shown, as can be seen in Figure IV.1. This is initial page was meant to be replaced in future releases with a more intuitive, scalable and accessible screen, which will indicate the available languages and accessibility services for each video, apart from listing them. This initial page has been called the *ImAc portal*.

Through this initial screen the users can also select the menu (UI) type, the language in which it will be shown (although both options can be dynamically set during consumption, as will be shown later), and the contents to consume by clicking on the associated image (which will be replaced by a thumbnail of the video in the next iteration). When selecting the contents, the player will be shown. If a VR-enabled device is being used, a menu will appear asking if the player has to be shown in tablet mode or in VR mode. This initial decision will have an impact on the positioning of the UI elements on the player, either attached to the screen (tablet mode) or to scene (VR mode). Right after, a banner will be shown indicating how to open the UI (see Figure IV.2).

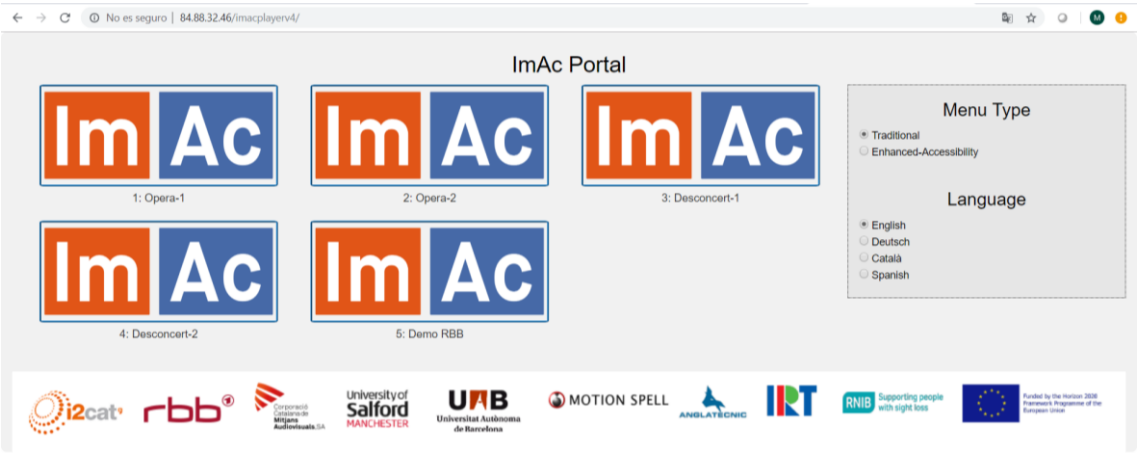


Figure IV.1- ImAc portal: initial Screen listing the contents and with initial settings

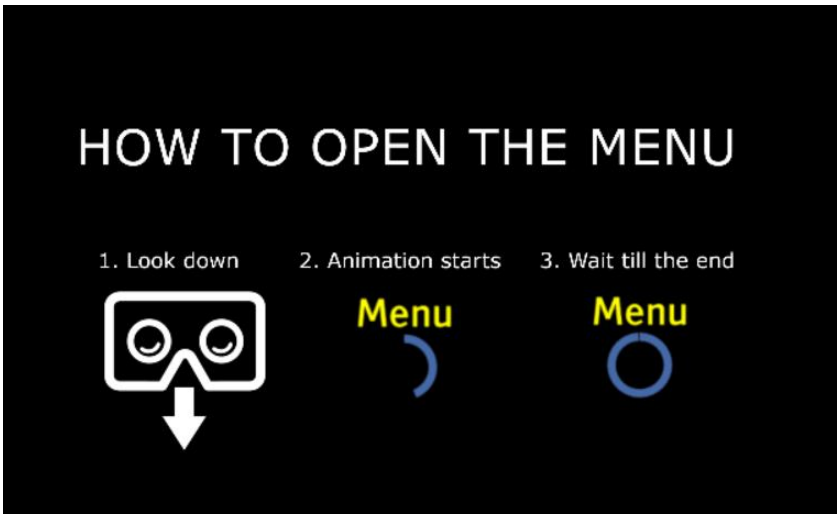


Figure IV.2- Banner indicating how the UI can be opened when starting the ImAc player

1.3. UIs and Features

This UI needs to take into account the sensorial capacities of the target users, and the requirements and preferences identified in the focus groups (gathered in WP2, and detailed in D2.2). Likewise, the UI to be designed needs to take into account the particularities of each platform and physical limitation of each screen, and need to be easily interpreted/used by all users, regardless of their sensorial abilities. In addition, the UI will determine various relevant aspects regarding the presentation of the (immersive and accessibility) contents, such as the "safe area", the comfortable Field-of-View (FoV), the objects' size and position, etc., and may involve that multiple pathways for media control and interaction need to be provided.

As a result of the requirements gathering tasks in WP2 during the first year of the project, it was decided to design and implement two modern, intuitive and user-friendly UIs in the ImAc player. The first one is a traditional UI (see Figure IV.3). The second one is an enhanced-accessibility UI targeted for users with low vision (aka low-sighted UI), which occupies most of the viewing area of the player (see Figure IV.4). Both UIs include the necessary controls to (de-)activate the considered accessibility services and set the different options and features to meet the requirements gathered in WP2. Next sub-sections provide a description of these different features, options and controls in both UIs. More emphasis will be given to explain the different

screens and menu items in the enhanced-accessibility UI, which allows navigating between its screens via arrows in a circular manner, as this UI is more innovative and less conventional than the traditional UI.

Another important issue in VR / 360º players consists of having a clean screen, avoiding the inclusion of intrusive visual elements that can negative impact the user's immersion. Accordingly, the controls of the UIs should not be always visible. To achieve this, it has been decided to implement a specific mechanism to show the UI (e.g. menu options and controls) only when necessary and having the option to close it afterwards. The decision has consisted of looking down for a specific period of time until a spinner control is completely loaded (see Figure IV.5 for the instructions and Figure IV.5 for how the animation looks like). This decision is based on the results of some preliminary tests. Both UIs can be closed by clicking on the exit control (see Figures IV.3 and IV.4), as in typical menus and screens.

The selection of the UI to be shown is indicated in the ImAc portal webpage (Figure IV.1), but a dynamic switching between the traditional and low-sighted UI can be done via voice control or in the settings options when pressing the wheel icon (see Figure IV.5), as explained later. The traditional UI is shown by default, at this stage of the player development.

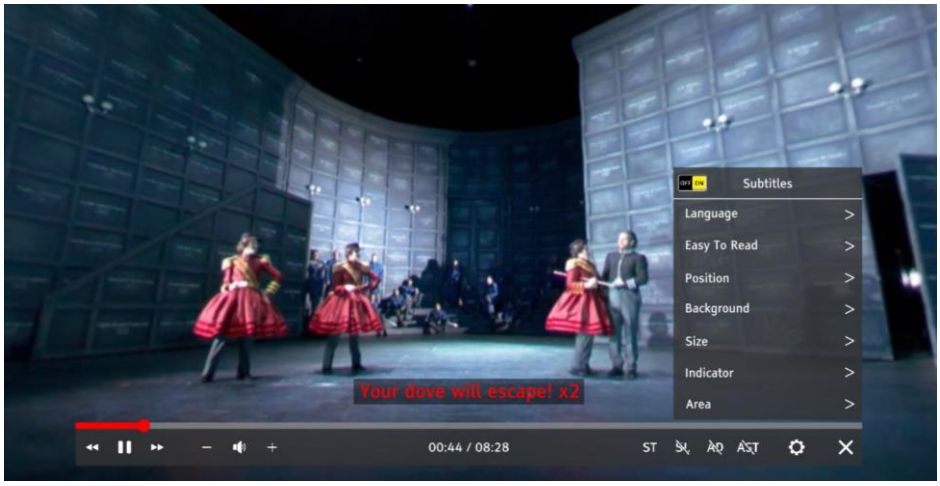


Figure IV.3- Traditional UI of the ImAc player

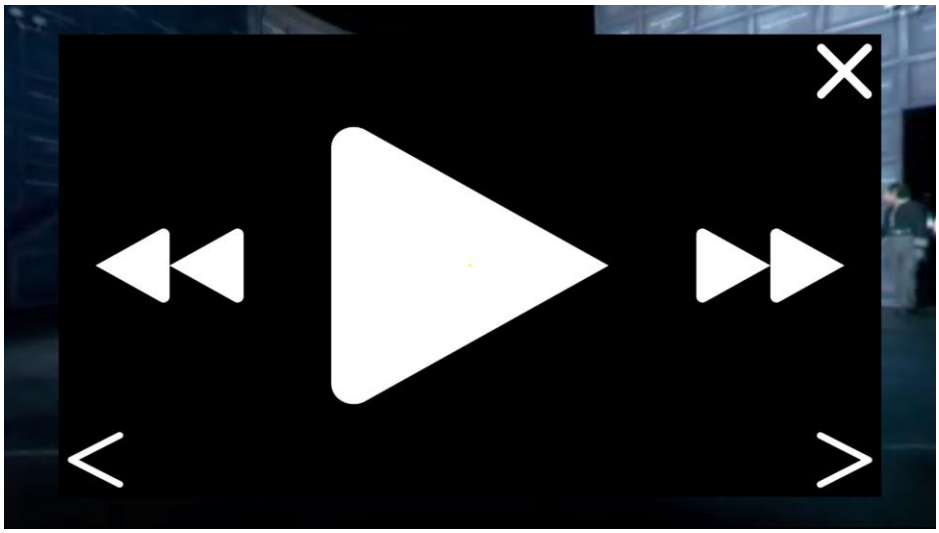


Figure IV.4- Enhanced-Accessibility (or Low-Sighted) UI

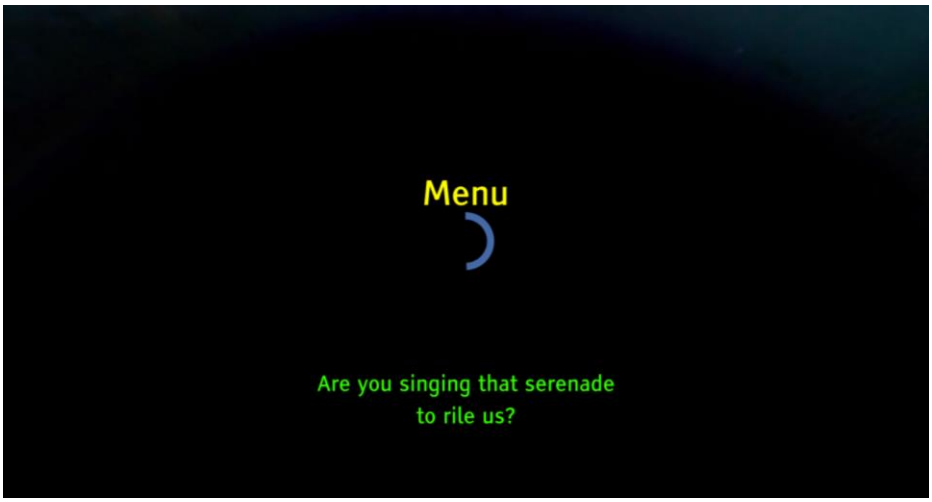


Figure IV.4- Looking down for a period of time to show the UI

1.4. Playback Controls and Volume Settings

The player includes the following playback controls: play, pause, seek forward and seek backward. The seek steps have been set to 5s. These can be seen in Figures IV.2 and IV.3. In addition, the traditional UI includes a progress bar that allows a customized seeking by clicking on any point of the bar (Figures IV.2 and IV.5). When clicking on any of the graphical playback controls, their size is reduced and their colour is changed to yellow (for high contrast) for a short period of time to provide visual feedback about their execution. This can be appreciated in Figure IV.6 for the "play" button, when compared to Figure IV.3. Specifically, when clicking on the "play" control, the "pause" control will be shown, and vice versa (see Figure IV.7). Likewise, the current playback time will be shown for a short period of time to also provide visual feedback about the current timeline (see Figure IV.8).

The enhanced-accessibility UI also includes a screen with volume setting controls (see Figure IV.9). It includes the controls for increasing and decreasing the volume level, in steps of 10% of the maximum volume. When clicking on them, the current volume level will be shortly displayed (see Figure IV.10). In addition, by clicking on the loudspeaker button, a strike will be added to this icon, indicating that the "mute" mode has been set (see Figure IV.11). Exactly the same controls are available in the traditional UI, as in conventional players (see Figures IV.2 and IV.5).

Figure IV.5- Settings Menu option to Switch between UIs

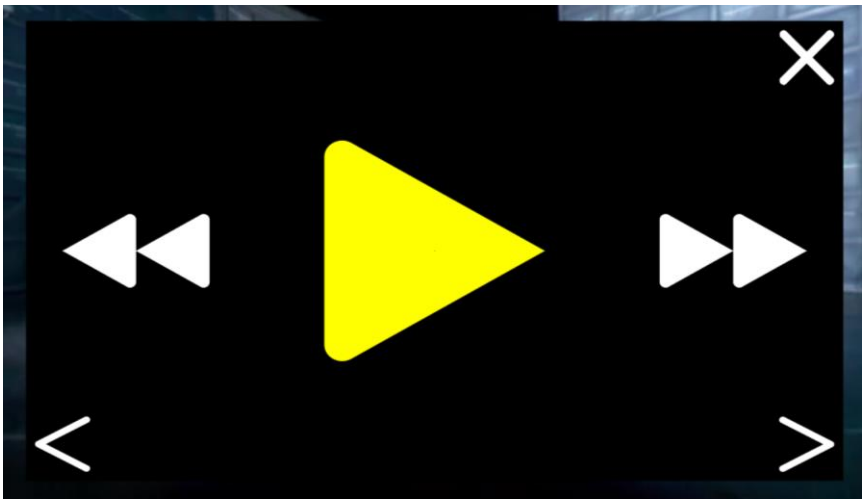


Figure IV.6- Visual feedback when clicking on control of the media player

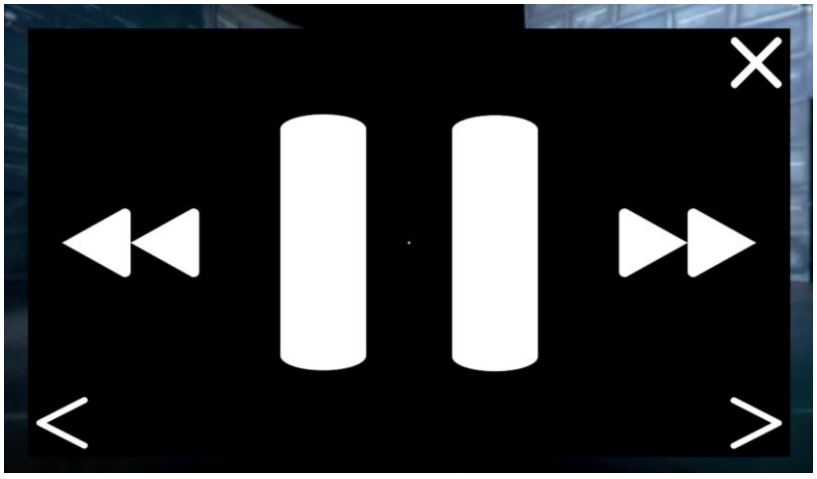


Figure IV.7- Pause button after having clicked on play

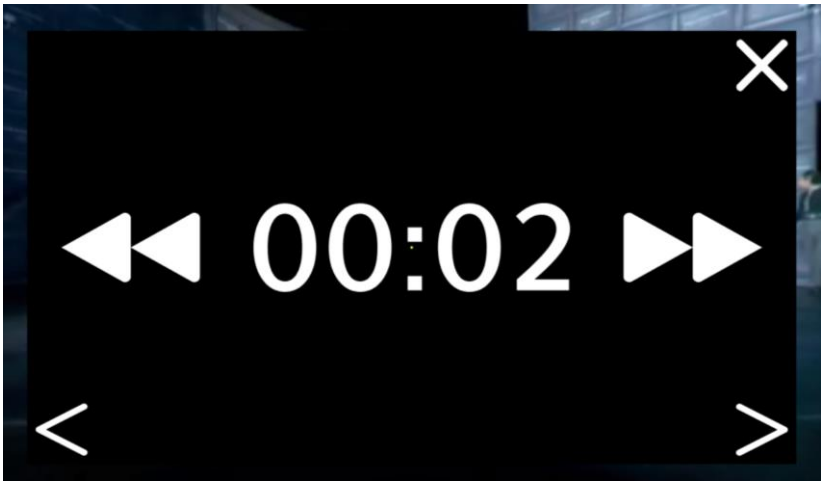


Figure IV.8- Visual feedback of the current playback position after having clicked on play

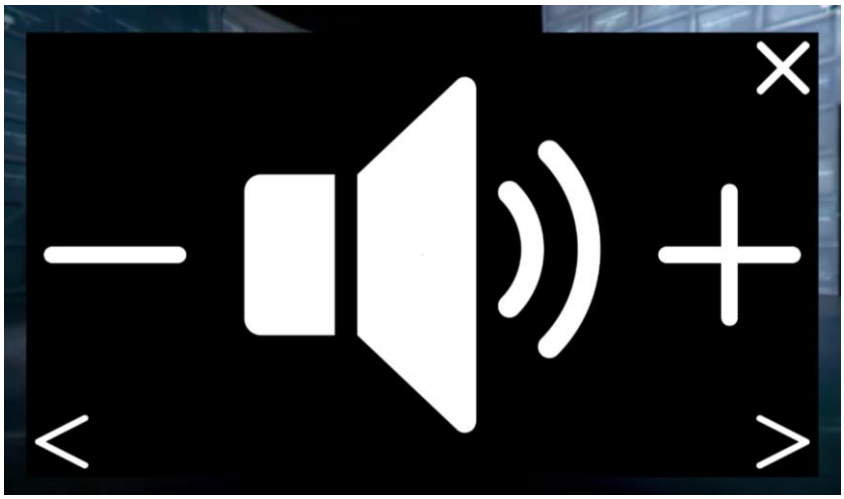


Figure IV.9- Volume setting controls in the low-sighted UI

1.5. (Personalized) Presentation of Subtitles (ST)

The main menu includes a screen (see Figure IV.12) for enabling (switching on / off) and setting each one of the accessibility services, namely: Subtitles (ST); Sign Language (SL); Audio Description (AD); and Audio Subtitles (AST). Subtitles, in the same manner as all the other accessibility services, can be enabled / disabled via a toggle control (see Figures IV.2 and IV.13).

Based on the requirements gathered in WP2, the subtitles will be always visible and centred in the FoV, regardless of whether the related scenes/objects in the 360º area are visible or not (see D2.2). However, different guiding mechanisms are provided to help the users in finding the associated speakers (explained later). Likewise, the player supports the presentation of subtitles in different colours, based on metadata generated at the production side. This contributes to a better identification of different speakers.

Next the different available options for presentation of subtitles are described.

1.5.1. Selection of ST Track

If different subtitle tracks are available, users will be able to select between them. This includes multiple languages (see Figure IV.13), and the selection of easy-to-read subtitles (see Figure IV.14).

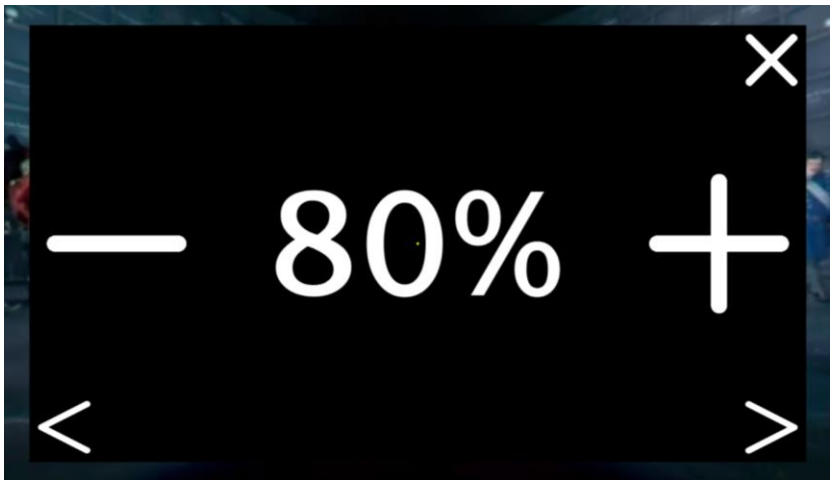


Figure IV.10- Visual feedback when setting the volume level

Figure IV.11- Mute icon shown once clicking on the loudspeaker icon. By clicking on it, the mute mode is disabled.

Figure IV.12- Menu to have access to the accessibility services in the enhance-accessibility UI

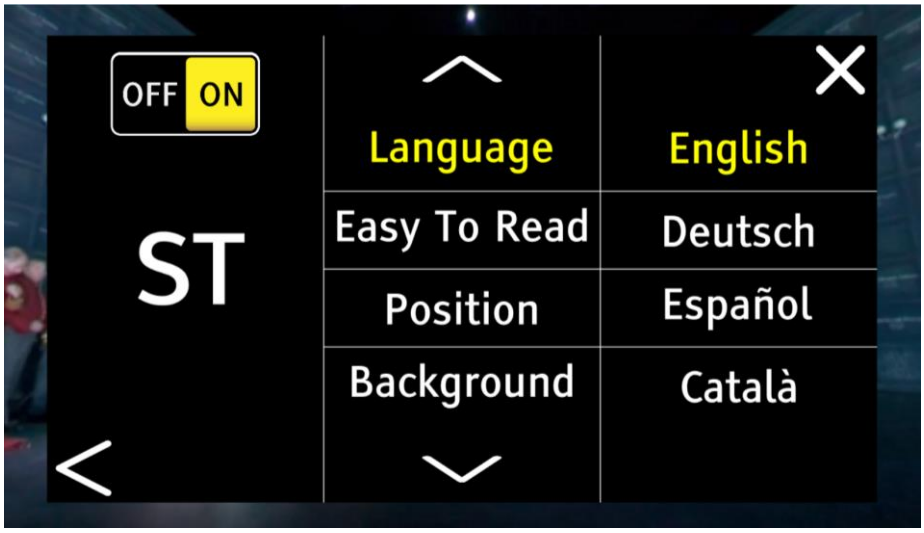


Figure IV.13- Activation / Deactivation of subtitles and personalization options

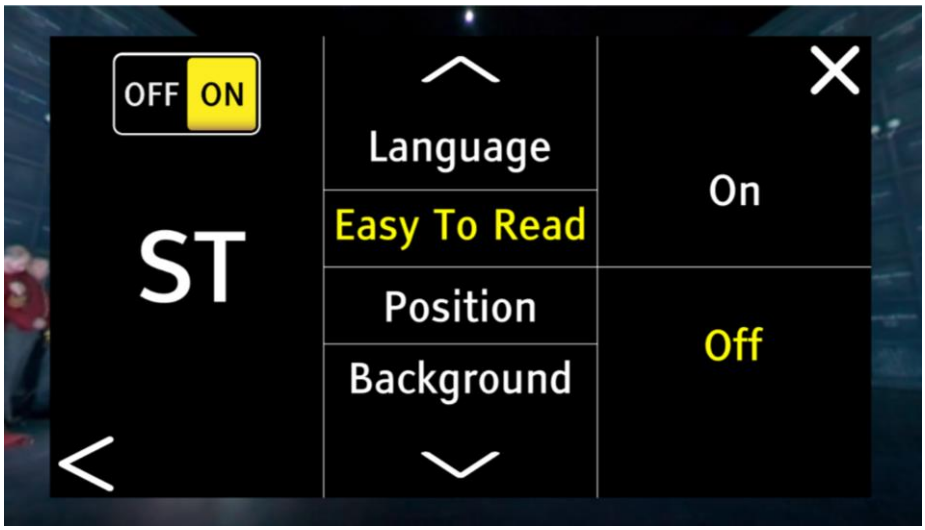


Figure IV.14- Activation / Deactivation of Easy-to-Read Subtitles

1.5.2. Dynamic Positioning

Regarding their position, subtitles are typically visible in the user's field of view in the middle slightly below eye line, two-lined. The ImAc player follows this approach, but it also allows dynamically setting the position of subtitles for their presentation at the top. In particular, user are able to select between two pre-defined positions in the viewing field (see Figures IV.15, IV.16 and IV.17):

- Bottom - Top

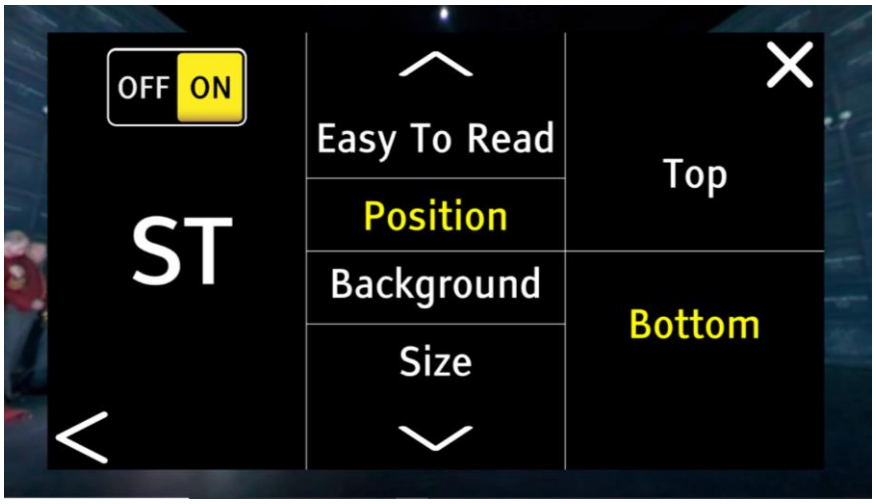


Figure IV.15- Menu options to set the position of subtitles in the Enhanced Accessibility UI

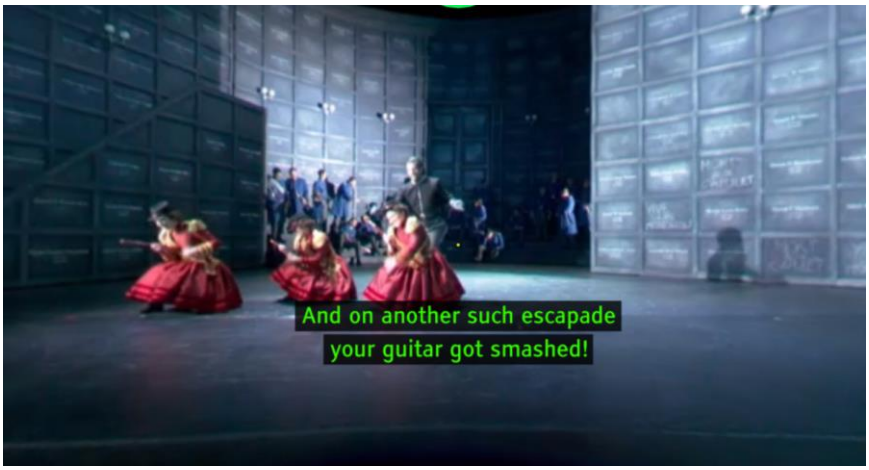


Figure IV.16- Subtitles placed at the bottom-centered position of the CFoV

Figure IV.17- Subtitles placed at the top-centered position of the CFoV

1.5.3. Background

Users are able to select between two different backgrounds for the subtitle area (see Figure IV.15): 1) semi-transparent box, with 80% of opacity (see Figure 26); 2) outline, with 2px for each font size (see Figure IV.17).

1.5.4. Subtitle Size

The TV broadcast subtitle services are still based on the Teletext limitations, as well as on viewer behaviour, including their reading speed. That implies that the number of characters is limited to about 37 characters per line to avoid line breaks, and it is suggested to use not more than 2 lines for a subtitle frame. ImAc is making use of these recommendations also in the VR world. These recommendations, together with the available area for display graphical elements (i.e. the CFoV, see next item), determine the maximum size for subtitles. In addition to that, and based on requirements from WP2, three different sizes can be enabled for the presentation of subtitles (see Figure IV.18): large (based on the maximum size), medium, and small.

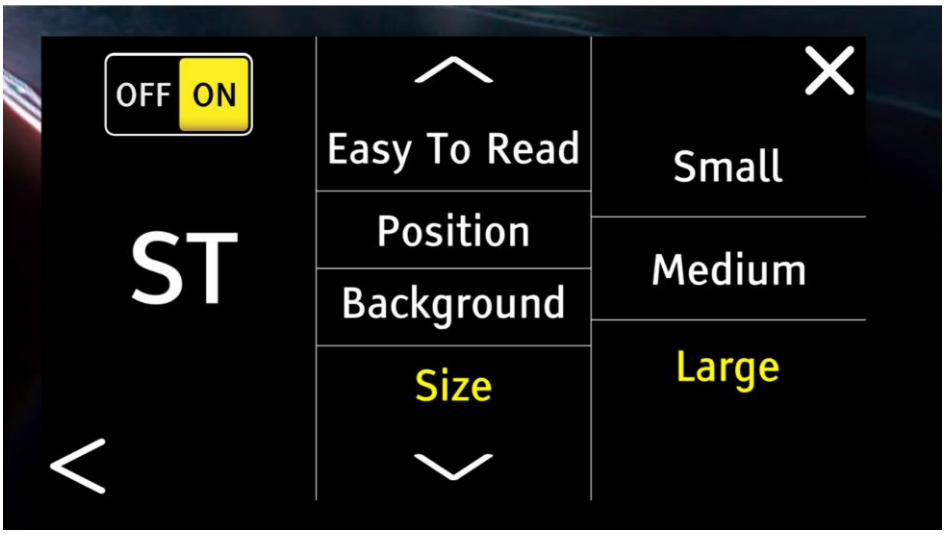


Figure IV.18- Menu options to set the position of subtitles in the Enhanced Accessibility UI

1.5.5. Safe area or Comfortable Field of View

The most appropriate size for the safe area where to present the accessibility contents and UI controls was investigated in the pre-pilot tests. Different percentages of both 1:1 and 16:9 ratios were considered, being the latter ones more adequate. The EBU guideline EBU R95 (https://tech.ebu.ch/docs/r/r095.pdf) that determines the safe area for 16:9 screens in a High Definition format was taken into account in these tests.

It was found that there were no clear preferences between three percentages of the 16:9 ratio (50%, 60% and 70%) and that the HMD in use has a significant impact, as their FoVs may differ. Therefore, it was decided to let the users choosing between three sizes for the display area (see Figure IV.19). When each option is selected, the edge of the area in shown by using a dotted yellow line for a short period (see Figure IV.20). This was added to help the users to better understand this functionality.

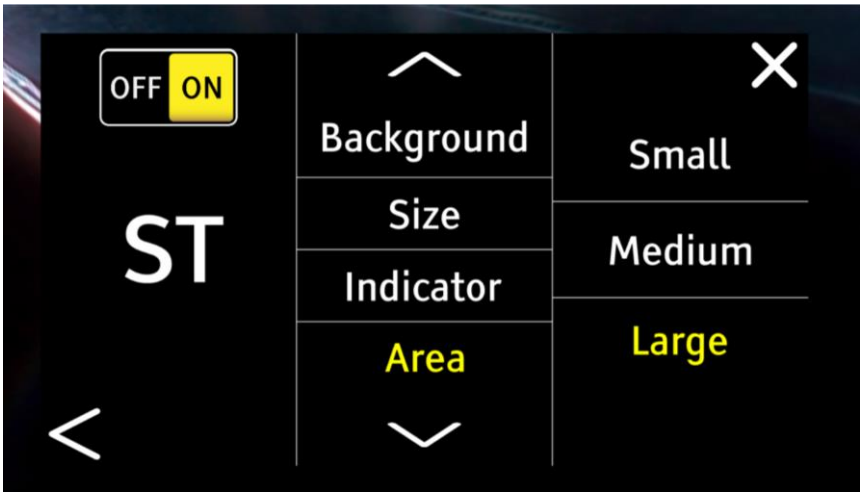


Figure IV.19- Menu options to set the size of the safe area or CFoV for subtitles in the Enhanced Accessibility UI

Figure IV.20- Visual feedback (yellow dotted line) once setting the size of the safe area or CFoV for subtitles

1.5.6. Guiding Mechanisms

As discussed, subtitles are always visible and centred in the FoV in the ImAc player. In order to assist users in identifying the associated speaker/s, different guiding mechanisms have been implemented (see Figure IV.21):

a) None: no guiding mechanism is used.

b) Arrows (see Figure IV.22): arrows on the right / left part of the subtitle frame indicate that the associated speaker is on the right / left of the current FoV. When the speaker in the FoV, the arrows disappear.

c) Radar (see Figure IV.23): a radar indicates where the speaker are in the 360º area, by representing the speaker as dots with the same color as the subtitle frames, and representing the current FoV as a portion of the 360º area;

c) Auto-Positioning: it consists of automatically adjusting the FoV based on the speakers' positions. When using this presentation mode, the subtitle editor can explicitly indicate for each subtitle frame being created / edited if the auto-positioning mechanism must be applied for that frame or not.

It is assumed that users can have preferences regarding the usage of these presentation modes, and that their appropriateness will be also determined by the type of video being consumed. Therefore, it was decided to include the three of them in the player, and let the user choosing the preferred one in each situation.

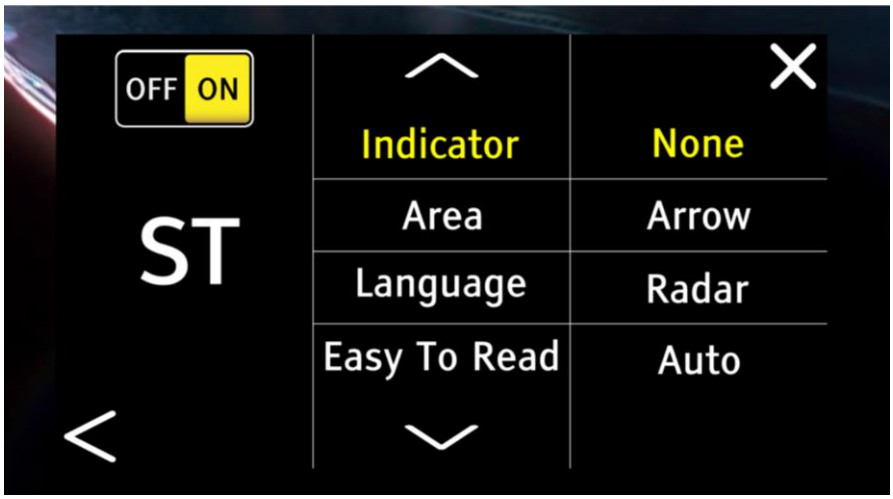


Figure IV.21- Guiding Mechanisms available in the ImAc player

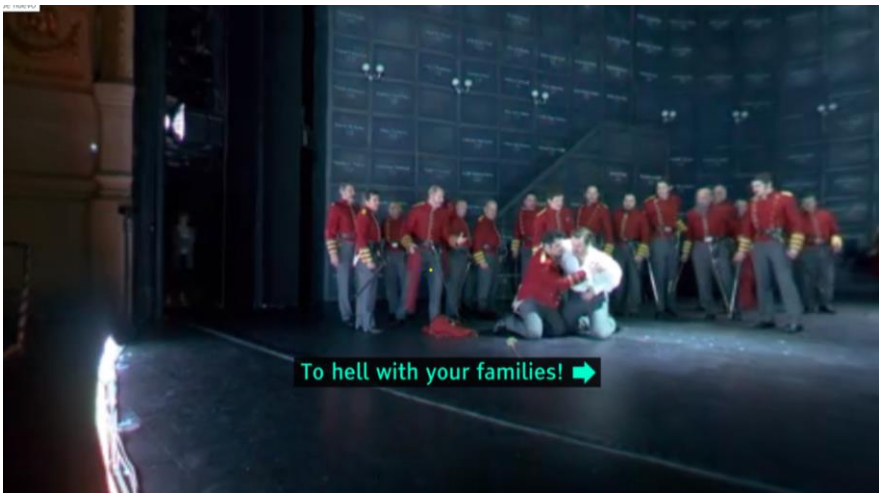


Figure IV.22- Arrows as a guiding mechanism to inform about the position of the speaker

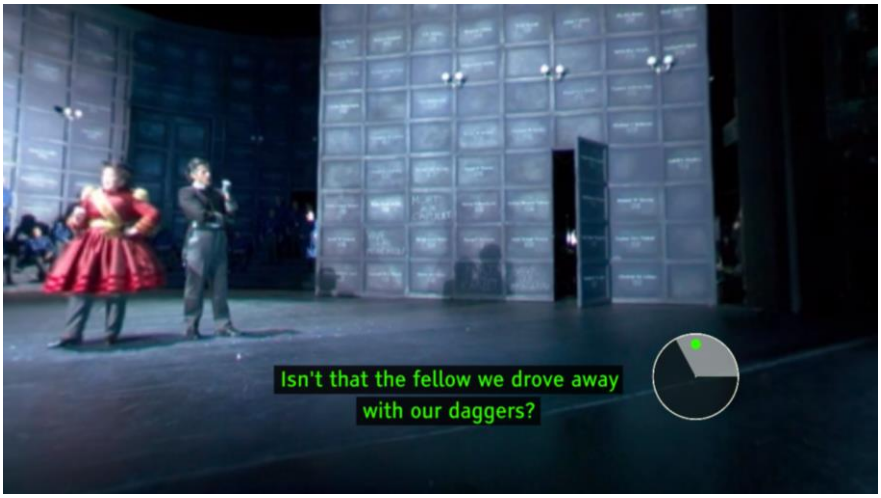


Figure IV.23- Radar as the guiding mechanism to inform about the position of the speaker

1.6. Presentation of Audio Subtitles (AST)

The menu options and settings for AST presentation are not implemented in the current version of the player yet. However, the libraries for presentation of both traditional audio and spatial audio are already developed and integrated into the player. Such options will be implemented once having the AST contents and their metadata ready.

More details about the AST related requirements can be found in D2.2.

1.7. Presentation of Audio Description (AD)

As for AST, the menu options and settings for AD presentation are not implemented in the current version of the player yet. However, the libraries for presentation of both traditional audio and spatial audio are already developed and integrated into the player. Such options will be implemented once having the AD contents and their metadata ready. Indeed, some tests with spatial audio (Ambisonics) have been already successfully conducted.

Based on the requirements gather in WP2, three presentation modes for AD will be implemented:

1) AD centred in the scene (Voice of God)

2) AD anchored in the scene (Friend on Sofa).

3) AD anchored on the event/object being described. (AD on action)

Options to have secondary AD tracks, and to be able to separately set the volume of the AD track will be also available. More details about the AD related requirements can be found in D2.2.

1.8. Presentation of Sign Language (SL)

By clicking on the SL acronym in the main menu (Figures IV.2 and IV.12), the presentation of SL video in a Picture-in-Picture mode can be enable and disabled, via a toggle control (see Figure IV.24). As for subtitles, different personalization options are available for the presentation of SL videos: 1) language (English, German, Spanish and Catalan); 2) position (top, bottom); 3) guiding mechanism (none, arrows, radar, auto-positioning); and 4) display area size (large, medium, small). Figure IV.25 shows an example of the presentation of a SL video with arrows as a guiding mechanism. In all presentation modes, the SL video is always kept in the same position of the FoV.

Figure IV.24- Menu for a personalized presentation of SL video

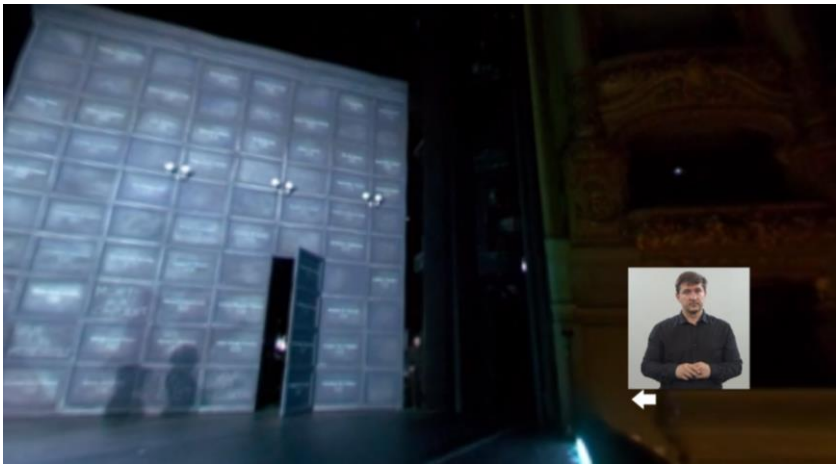


Figure IV.25- Presentation of SL video with arrows as a guiding mechanism

1.9. Voice Control

Apart from graphical menus and controls, voice control has also been identified as a desirable interaction modality for the ImAc player. Voice control is becoming increasingly adopted (e.g. Siri, Google Now, Amazon Echo…), and this is also happening for accessibility services (e.g., Voice Over, Talkback…). Many existing solutions for voice control have been analysed. The list includes:

1) Google Speech API
2) Microsoft Cognitive Services
3) API.AI
4) Project Oxford (Microsoft)
5) Alexa
6) Google Voice Interaction API
7) Wit.ai
8) Api.ai

9) IBM Watson
10) Annyang
11) Web Speech Directly
12) Speechmatics
13) Vocapia Speech to Text API
14) Speech Engine_IFLYTEK CO., LTD.
15) UWP Speech
16) CMU Sphinx
17) Kaldi

From this list, the following web-based solutions have been tested: Annyang, Web Speech, and Pocket Sphinx: https://imac.crazysandbox.co.uk/voice/

After these initial tests, the World Wide Web Consortium (W3C) Web Speech API was the chosen solution for voice control. This API has two parts: SpeechRecognition, for Asynchronous Speech Recognition; and SpeechSynthesis, for Text-to-Speech synthesis. In particular, Annyang, an API built on top of the Web Speech API, has been used to implement the recognition of commands. It supports all the languages considered in ImAc, and it has been provide a satisfactory performance in preliminary tests. The integration of the Web Speech API into the ImAc player is sketched in Figure IV.26. An example of how the association between the spoken words to be detected and the associated commands or player functions is shown below:

```
var commands = {
   'play': function() {speak("play"); videoPlay();},
   'pause': function() {speak("pause"); videoPause();},
   'bigger': function() {speak("bigger subtitles"); adjustLPL(-1);},
   'smaller': function() {speak("smaller subtitles"); adjustLPL(1);},
   'forward': function() {speak("move forward"); videoSeek(5)},
   'back': function() {speak("move backward"); videoSeek(-5)},
};

// Add our commands to annyang
annyang.addCommands(commands);

// Start listening. You can call this here, or attach this call to an
event, button, etc.
annyang.start
```

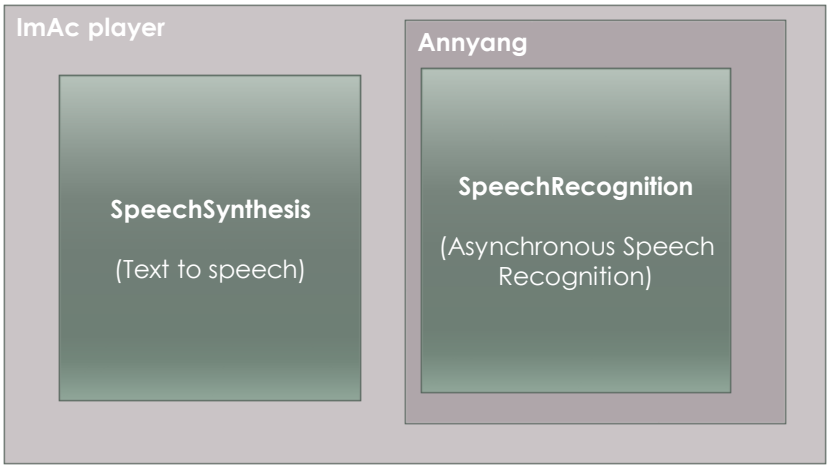


Figure IV.26- Integration of Voice Control into the ImAc player

Table IV.1 provides a list of the voice commands to be supported. From Table IV.1, a first iteration with a selection of common commands has been integrated and tested. However, it has not been publicly shared yet, as it requires the availability of a secured HTTP (HTTPS) server. The list of command words, for each one of the considered languages, together with the associated JavaScript (JS) command / function names for this first iteration, are indicated in Table IV.2.

Voice Command	Action	State
Library	Switches to library mode	From any mode
Player	Switches to player mode	From any mode
Previous	Previous (Item or Chapter)	With content playing
Next	Next (Item or Chapter)	With content playing
Show Menu	Toggle menu bar in library	In any mode
Play / Pause	Play/Pause Toggle	With content queued up
Stop	Stops Playback	With content playing
Seek back 10	Skip Backward 10s	With content playing
Seek back 20	Skip Backward 20s	With content playing
Seek back 30	Skip Backward 30s	With content playing
Seek forward 10	Skip Forward 10s	With content playing
Seek forward 20	Skip Forward 20s	With content playing
Seek forward 30	Skip Forward 30s	With content playing
Subtitle on/off	Subtitle On/Off Toggle	Playing content which has subtitles
Audio Subtitling on/off	Subtitling On/Off Toggle	Playing content which has subtitles
Audio description on/off	AD On/Off Toggle	Playing content which has AD
Signer on/off	Signer On/Off Toggle	Playing content which has signer
Speed normal	Play Speed Normal	With content playing
Speed Slow	Play Speed Slow	With content playing
Speed Fast	Play Speed Fast	With content playing

Fullscreen	Switch Fullscreen	With content playing
Mute	Mutes sound	In any mode
Volume Up	Increases sound volume	In any mode
Volume Down	Decrease sound volume	In any mode
Menu	Context menu shortcut	In any mode
Personalization Options	Context menu shortcut	In any mode

Table IV.1 - Voice commands supported by the ImAc Player

English	**Catalan**	**Spanish**	**German**	***JS command / function name***
Volume Up	Apujar Volum	Subir Volumen	Lauter	getChangeVolumeFunc(true)
Volume Down	Abaixar Volum	Bajar Volumen	Leiser	getChangeVolumeFunc(false)
Play			Wiedergabe / Fortsetzen	getPlayPauseFunc(true)
Pause	Pausar	Pausar	Pause	getPlayPauseFunc(false)
Seek Forward	Avançar	Avanzar	Weiter	getSeekFunc(true)
Seek Back	Retrocedir	Retroceder	Zurück	getSeekFunc(false)
Subtitles On	Activar subíitols	Activar subtítulos	Untertitel an	getOnOffFunc('subtitlesOffButton')
Subtitles Off	Desactivar subtítols	Desactivar subtítulos	Untertitel aus	getOnOffFunc('subtitlesOnButton')
Open Menu	Obrir Menú	Abrir Menu	Öffne Menü	getOpenMenuFunc()
Close Menu	Tancar Menú	Cerrar Menu	Schließe Menü	getCloseTradMenuFunc()

Table IV.2 - Selected Voice commands for the first iteration of the Voice Control Mechanisms

Regarding voice synthesis, the objective is to acknowledge the execution of commands with an audible response, by making use of the SpeechSynthesis part of the Web Speech API. Guidelines

to provide the spoken feedback are being explored in order to adopt the most appropriate approach. Next, a piece of code for the implementation of the voice synthesis is provided:

```
function speak(message)
{
  if (!window.speechSynthesis) {
     warnNoSpeechSynthesisSupport();
     return;
  }
  var utterance = new SpeechSynthesisUtterance(message);
  window.speechSynthesis.speak(utterance);
}
```

A functional demo with a partial implementation of the voice control features (but not integrated in the full-fledged ImAc player yet) is available here: https://imac.crazysandbox.co.uk/responsive/responsive360_voice.php

1.10. Player Settings

Apart from the settings and personalization options for each accessibility service (ST, AT, AD, SL), the player also includes a menu for general settings (see Figures IV.5 and IV.27). Next, the available features and options are detailed.

1.10.1. Language Change

Although the (initial) language for the UI can be selected in the initial webpage (ImAc portal), it can be also dynamically set during media consumption in the menu settings (see Figure IV.28). The available languages are: Catalan, Spanish, German and English.

1.10.2. Menu / UI Type

Although the (initial) UI type to be displayed can be selected in the initial webpage (ImAc portal), it can be also dynamically changed in the menu settings (see Figure IV.5 for the traditional UI, although the same options are available via the Enhanced-Accessibility UI, as shown in Figure IV.28).

1.10.3. Voice Control

Although the voice control feature can be enabled when starting the ImAc player, it can be also enabled and disabled in the menu settings (see Figure IV.28).

1.10.4. User's Profile / Preferences Saving

In the menu settings, it is also possible to store the user's settings / preferences (see Figure IV.28), so they will be enabled in future media sessions by default. It is done by saving the configuration using browser cookies. It is also envisioned to define different User's Profiles and to be able to select them in the menu settings in future stages of the project.

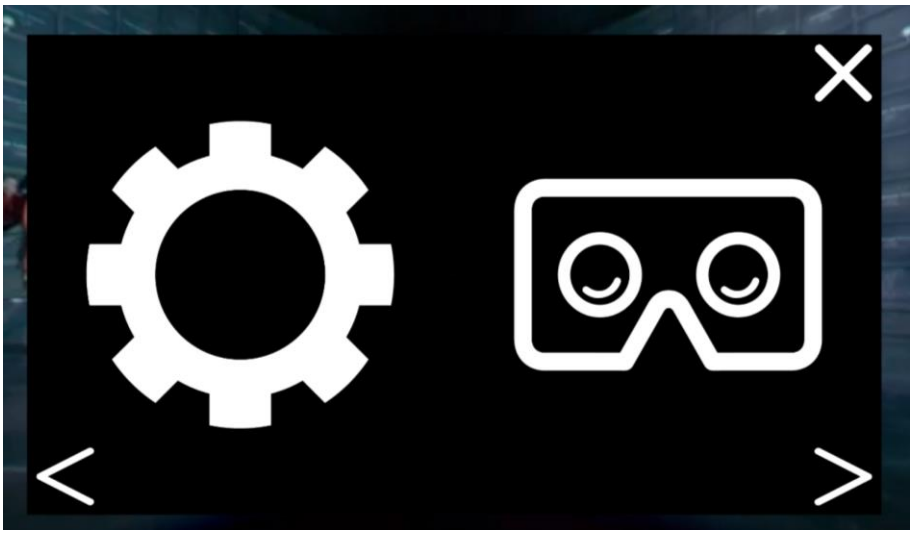


Figure IV.27- Menu screen for the general settings

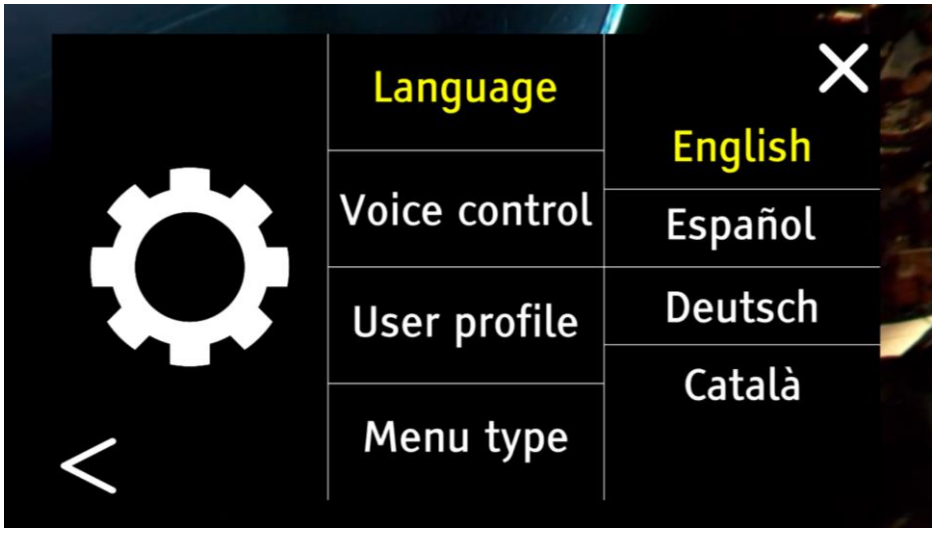


Figure IV.28- Dynamic selection of the UI language in the settings menu

ANNEX V. IMAC PLAYER FACTSHEET

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 761974.

ImAc Player

by i2cat

Accessibility-enabled VR360 web player. It enables the consumption of 360º video and spatial audio, augmented with a hyper-personalized presentation of access services (subtitles, audio description and sign language interpreting).

Features

Use of web-based components:

- Universal (cross-platform, cross-browser and cross-network) support.
- No need for installations / updates at the client side.
- Built on top of dash.js, the reference player for Moving Picture Experts Group (MPEG) Dynamic Adaptive Streaming over HTTP (DASH) standard.

Supported media formats:

- Codec Agnostic (browser-capable codecs).
- Traditional 2D video and 360º video (in both Equirectangular and CubeMap projections). 3D Stereoscopic will be supported.
- Traditional 2D audio (including uncompressed WAV) and 3D spatial audio (First and Higher Order Ambisonics).
- Internet Media Subtitles and Captions (IMSC) Subtitle file format. It is a profile of Timed Text Mark-up Language (TTML), standardised by W3C.

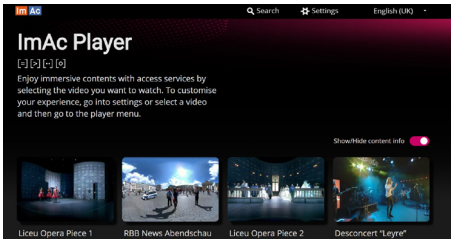
Supported consumption devices (and interaction modalities):

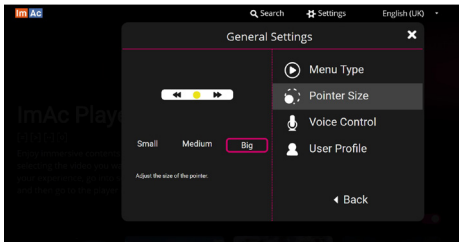
- PC and laptops.
  - » Use of mouse and keyboard for navigation and interaction.
- Mobile devices (including tablets, smartphones).
  - » Use of touch screen, gyroscope and physical buttons for navigation and interaction.
- VR devices (including Head Mounted Devices or HMDs).
  - » Use of HMD buttons, movement trackers and controllers for navigation and interaction.

Portal

Landing page for:

- Language selection.
- Settings.
- Search and filtering features.
- Catalogue of videos, indicating the available access services and languages.
- Video selection.

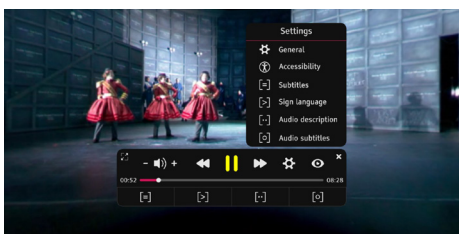




User Interface (UI)

- Responsive design.
- Adapted to VR environments.
- Visual feedback on the execution of commands.
- Enlargement features.
- Preview feature.

- Universal Icons for Accessibility:
  [=] [o] [··] [<]
- Open UI by looking down, by performing consecutive clicks, or via voice control.



i2cat · rbb · Corporació Catalana de Mitjans Audiovisuals, SA · University of Salford Manchester · UAB Universitat Autònoma de Barcelona · MOTION SPELL · ANGLATECNIC · IRT · RNIB See differently

Access Services

Supported features for **Subtitles (ST)**:

- Two rendering modes: subtitles attached to the speaker / position of the scene (world-referenced) or always visible at the bottom center of the Field of View (FoV) (user-referenced).
- Personalized presentation: size, position and language.
- Responsive subtitles.
- Easy-to-Read Subtitles.
- Integration of Non-Speech Info (like text descriptors, emojis…).
- Integration of styling effects for speaker's representation (color, formats, voice-off…).

Supported features for **Sign Language (SL)**:

- Personalized presentation: size, position and language
- Non-Continuous video streams (allowing automatically showing/hiding the video window based on the signer's activity).

Supported features for **Audio Description (AD)**:

- Three audio placement modes and narratives.
  - » Classic: Positioned in-head.
  - » Static: Like a Friend whispering in your ear.
  - » Dynamic: Coming from the direction where speaker / action is.
- Simultaneous Secondary Audio Description tracks (for specific scenes, actions or objects).
- Independent volume settings for the AD and main audio tracks.

Supported features for **Audio Subtitles (AST)**:

- Two audio placement modes.
  - » Classic: Positioned in-head.
  - » Dynamic: Coming from the direction where speaker / action is..
- Independent volume settings for the AST and main audio tracks.
- Easy-to-Read Subtitles.

Assistive Technologies

- **Guiding Methods** to indicate where the target speaker/action is in the 360º area: arrows, radar, auto-positioning mode, and use of spatial audio.
- **Zoom Features.**
- **Voice Control** (Voice Recognition and Spoken Feedback to the Execution of Commands).
  - » Development of a Relay Server to connect with external voice recognition systems, like Alexa (Echo Dot) & Amazon Web Services and Google Home.

Open-Source

The source code will be soon released. Stay tuned (http://www.imac-project.eu/)!

Standardization

The player features are being proposed for standardization in diverse bodies / organizations.

Multi-Screen Scenarios

Technology to enable multi-screen scenarios, composed of one main screen and one or multiple companion screens (e.g., tablet, smartphone, HMDs) providing extra related contents in an interactive, personalized and synchronized manner.

- Support for discovery, association, launching, interaction and media synchronization features.
- Support for both Full web-based and Hybrid Broadcast Broadband TV (HbbTV) scenarios.

Access URLs

Current version: https://imac.gpac-licensing.com/player/

Contact media.team@i2cat.net to have access to other dev releases with extra research-oriented features.

Video

Scan the QR code to watch a video showing the features of the ImAc player.