Deliverable

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

D3.6. Integration and Testing Report

Revision: 1.0

Authors: Chris Hughes (USAL), Peter tho Pesch (IRT), Mario Montagud (i2CAT), Marc Brelot (MSE), Romain Bouqueau (MSE) and Enric Torres (ANG)

Delivery date: 12-11-2018

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 761974		
Dissemination Level		
P	Public	P
C	Confidential, only for members of the consortium and the Commission Services	

Abstract:
This document describes the integration and testing plan for the ImAc project. We describe a methodology for integration and system testing based on standard software engineering aproaches. This allows us to examine the different components of the ImAc platform (as described in the Technical Architecture - D3.1) and describe each of these components are integrated. We also define a testing strategy based on the user requirements gathered in D2.3.

REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	01-12-2017	Chris Hughes	USAL	Template and ToC
0.2	21-06-2018	Chris Hughes	USAL	Written Methodology and Testing Plan
0.3	04-08-2018	Chris Hughes	USAL	Added a struture for the test results
1.0	12-11-2018	Chris Hughes	USAL	Version 1

EXECUTIVE SUMMARY

This document is published in two iterations to match the development cycle of the ImAc project. In this document we describe the integration and testing plan for the ImAc project. We describe a methodology for integration and system testing based on standard software engineering aproaches.

The system components are defined in the Technical Architecture (D3.1) and this document describes how each of these components are integrated as well as the current status of this development. This document also defines a testing strategy based on the user requirements gathered (D2.3).

Chapter 1 provides an overview of this document, describes the objectives and scope of the integration and testing report, and details how it fits into the larger ImAc project.

Chapter 2 describes our approach to testing, discuses the theory of system integration testing and provides the methodology that we employ for testing the ImAc platform.

Chapter 3 describes specific Integration points within the ImAc project the the activities that they relate to as well as an Integration and Testing plan which identifys current constraints within the system and the strategy for testing these points.

Chapter 4 revisits the key size and performance requirements defined in the system architecture and eveluates the success of the ImAc system.

Chapter 5 provides the results of acceptance testing of the ImAc platform in order to meet the proove that the system meets the user requirements defined in D2.3.

Chapter 6 concludes the document with a summary of the current status of the software and provides a pathway for testing in the second iteration.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Chris	Hughes	USAL	c.j.hughes@salford.ac.uk
Mario	Montagud	i2CAT	mario.montagud@i2cat.net
Marc Romain	Brelot Bouqueau	MSE	marc.brelot@gpac-licensing.com
Enric	Torres	ANG	enric@anglatecnic.com
Peter	tho Pesch	IRT	thopesch@irt.de

CONTENTS

LIST OF ACRONYMS

Acronym	Description
AD	Audio Description
AST	Audio Subtitles
ST	Subtitles
SL	Sign Language
HUR	Home User Requirement
PUR	Professional User Requirement
HMD	Head Mounted Display
FOV	Field of View
CDN	Content Delivery Network
ISP	Internet Service Provider

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 accessible content management, packaging and distribution, customisation of the experience, and display of immersive and adapted content. The design will be fed by requirements gathered in WP2 (T2.2. and T2.3). This includes:

- To design the architecture of a robust platform capable to integrate all the components developed in the project.
- To design and implement the content management component facilitating access to multiple content formats and its storage.
- To adapt and integrate a real-time process to effectively encode multiple streams from inclusive content (i.e. subtitles, audio description and sign language) into full omnidirectional video.
- To design and implement a delivery chain that can process the input from Production (especially for the Accessibility and Immersive sides) and make them available accurately to the player using standard technologies.
- To design and implement a player (including clients and libraries) required to display omnidirectional video across devices (TV, second screen and HMD) maintaining coherence, synchronization, and enabling interaction and personalisation features.
- Validate development in semi-open pilots and large-scale pilots.
- Disseminate and communicate the WP outcomes among other researchers and industry stakeholders.

This document provides the integration and testing report, which describes the verification that the software meets the goals of the technical Architecture (D3.1)

1.2. Scope of this document

This document provides a detailed description of the overal testing process of the ImAc system. This is described in 4 main sections:

- Chapter 2: Approach, Theory and Methodology
- Chapter 3: Integration
- Chapter 4: Acceptance Testing

1.3. Status of this document

This is the first version of D3.6 with delivery foreseen in M10. A revised version of this document will be delivered in M20.

1.4. Relation with other ImAc activities

The PERT diagram illustrates the relation between D3.6 and the other ImAc activities. The Architecture design is built based upon the findings of D3.1 Platform Architecture and it feeds into the implementation of both the Accessibility Services and the Immersive Platform.

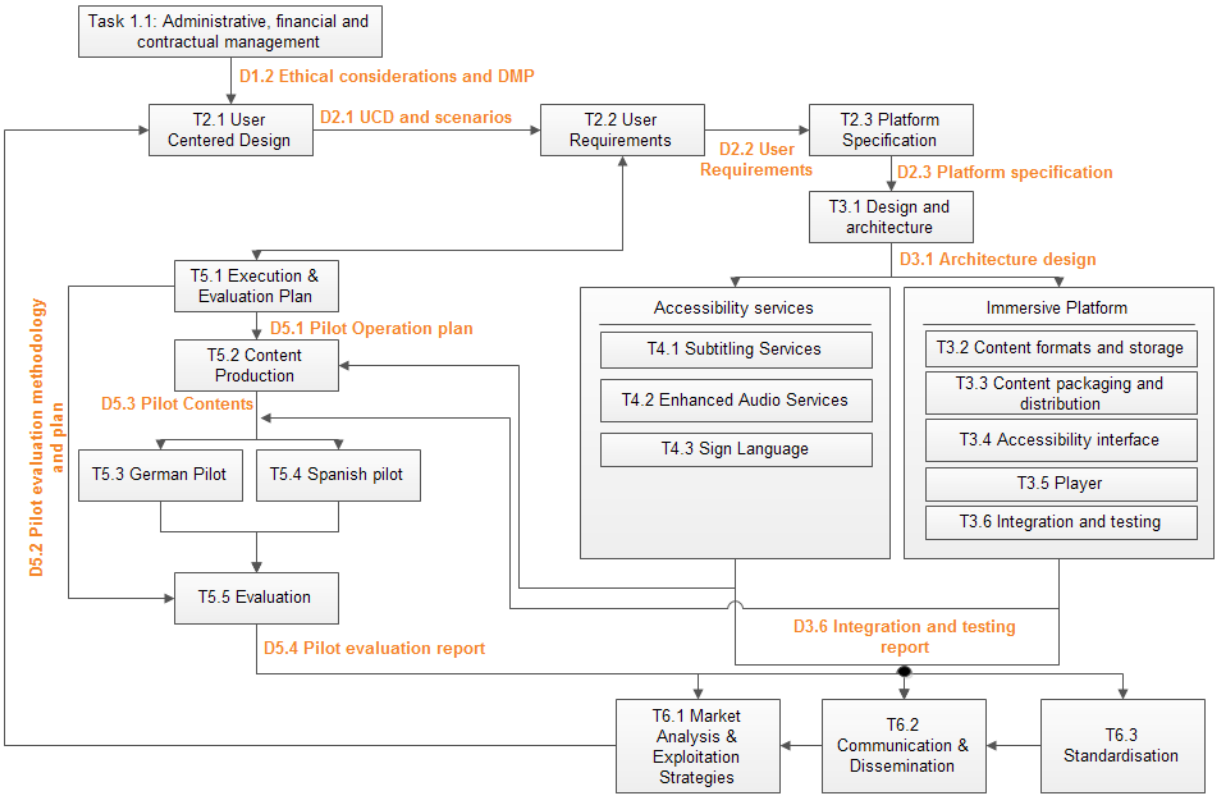

Figure 1 - PERT Diagram illustrating the relationship between D3.1 and other ImAc activities.

2. Approach, Theory and Methodology

2.1. Continuous Delivery

The ImAc project is following a process of continuous integration. This is a common approach when multiple developers from different physical sites regularily integrate their code into a shared repository. This enables a single deployment point where integration can be tested and verified on a regular basis. This allows errors to be identified quickly and easily fixed as changes between versions are typically small and therefore allows the specific changes that have introduced errors to be easily identified.

The success of continious integration relies heavily on version control as each component must be tested with a known version and any identified errors must be repeatable with the specific software versions.

Between the code being developed for each component and deployment our testing approach follows a four step process as shown in figure 1.

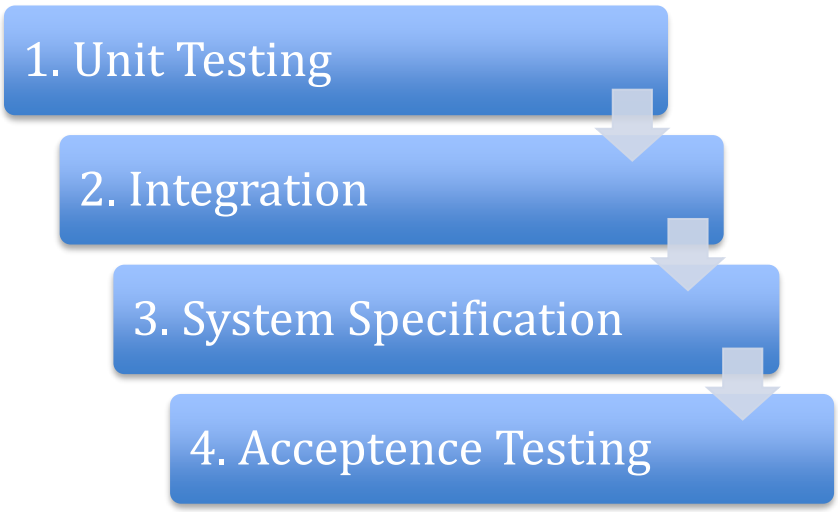

Figure 1 - Continuous Delivery

2.2. Unit Testing

After development each software component is broken down into the smallest testable component, or unit. Each of these is then tested independently to validate each unit of software performs as designed. Each development partner has been responsible for the unit testing of their own software components in isolation and must therefore be satisfied that the software behaves as expected before being integrated with the ImAc platform.

2.3. Integration

Next each component is commited to the shared project repository. This allows each individual unit to be combined and tested as a group. The purpose of this level of testing it to expose the faults in the interaction between each integrated unit, particularily where units are being developed by individual sites.

2.4. System Specification

The entire system is then tested against the key sizing and timing requirements which were defined in the Technical Architecture (D3.1). This enables the systems compliance with the specified requirements to be evaluated.

2.5. Acceptance Testing

Finally the software is tested for acceptability. This is done by evaluating the software against the requirements (D2.3) and asseses whether the software is acceptable to the user by satisfying their requirements and prooving that the system is acceptable for delivery.

3. INTEGRATION

Figure 2 identifies the key integration points within the Imac Project.

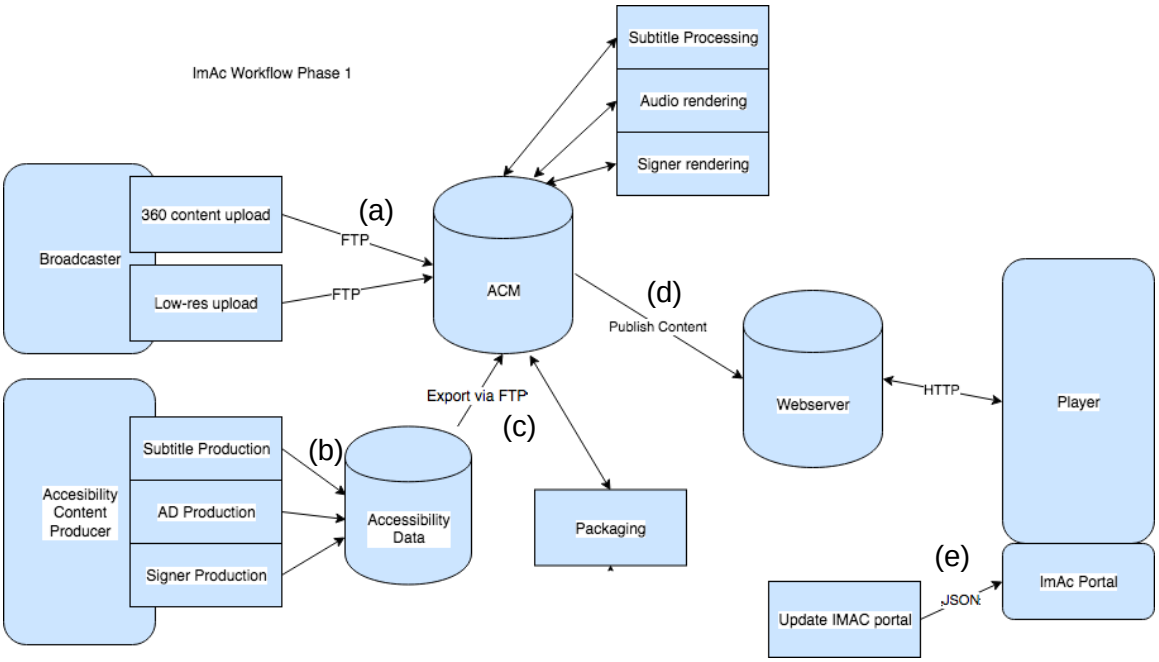

Figure 2 – Key Integration points withing phase 1 of the ImAc project

3.1. 360-degree content upload

Integration Point	Between the Broadcaster and ACM as shown in figure 2(a)
Current Interface (Phase 1)	Broadcaster (TVC, RBB) uploads HQ content in FTP. This FTP has a folder named “Input”. The name of the file uploaded is taken as the ID of the content and this ID will be used through the workflow.
Result	• HQ content is available on FTP server. • Program ID is defined (by the file name). • The upload is triggered by the upload of a signature file.
Planned Interface (Phase 2)	A web client in the ACM
Testing Details	Server SFTP: imac.gpac-licensing.com User: imac Pwd: jyfRwHyqZIDy7B HQ content includes 360 video and one (or more) main audio mixes. Supported format(s) video: H.264 360º Equirectangular 4K or superior resolution

	Supported format(s) audio: AAC or WAV with FOA (= First Order Ambisonic), Binaural (stereo),

3.2. Preview (Low Resolution) Upload

Integration Point	Between the Broadcaster and ACM as shown in figure 2(a)
Current Interface (Phase 1)	Broadcaster (TVC, RBB) uploads a low-res version of the content in ACM. The name of the file uploaded is the same as chosen in #1, but with the suffix "_lowres".
Result	• Low-res content is available in ACM. • Program in ACM is linked to HQ content on ftp
Planned Interface (Phase 2)	Automatically produced by the ACM
Testing Details	lowres content includes 360 video and one main audio mixes. Format video: mp4/h264 Format audio: aac, stereo

3.3. Accesibility Content Production

3.3.1. Subtitle Production

Integration Point	Between the Accesibility Content Producer and the Accessibility Content Database as shown in figure 2(b)
Current Interface (Phase 1)	Multiple steps: 1) Broadcaster (TVC, RBB) assigns subtitling work to one or more subtitlers from the CM interface (from the subtitling edit dialogue of the asset that appears on the right side of the screen when clicking the icon  on the asset). 2) Subtitlers access to the ED interface (straight after logging to ACM when they only have subtitler permission or by clicking the ED icon from the CM interface when they also have permission on the CM interface). 3) Subtitler clicks on one of his/her pending subtitling works. 4) The web subtitling editor opens with the subtitling work and the corresponding video. The subtitler can create the subtitling and click the save button to save it to the ACM.
Result	• Subtitles are available in ACM.
Planned Interface (Phase 2)	As Phase 1

Testing Details	ACM access: http://imac.gpac-licensing.com/acm/cm/assets.php# user: test pw: test

3.3.2. AD Production

Integration Point	Between the Accesibility Content Producer and the Accessibility Content Database as shown in figure 2(b)
Current Interface (Phase 1)	Multiple steps: 1) Broadcaster (TVC, RBB) assigns AD work to audio descriptors from the CM interface. 2) Audio descriptors access to the ED interface (straight after logging to ACM when they only have audio descriptor permission or by clicking the ED icon from the CM interface when they also have permission on the CM interface). 3) Audio descriptor clicks on one of his/her pending AD works. 4) The web AD editor opens with the AD work and the corresponding video. 5) The audio descriptor can create the AD and click the save button to save it to the ACM.
Result	• AD assets are available in ACM.
Planned Interface (Phase 2)	As Phase 1

3.3.3. Signer Production

Integration Point	Between the Accesibility Content Producer and the Accessibility Content Database as shown in figure 2(b)
Current Interface (Phase 1)	Multiple steps: 1) Broadcaster (TVC, RBB) assigns SL work to signer from the CM interface. 2) Signer access to the ED interface (straight after logging to ACM when they only have signer permission or by clicking the ED icon from the CM interface when they also have permission on the CM interface). 3) Signer clicks on one of his/her pending SL works. 4) The web SL editor opens with the SL work and the corresponding video. 5) The signer can create the SL and click the save button to save it to the ACM.
Result	• Signer file package is available in ACM.

Planned Interface (Phase 2)	As Phase 1
Testing Details	ACM access: http://imac.gpac-licensing.com/acm/cm/assets.php# user: test pw: test

3.4. Accesibility data export to FTP

Integration Point	Between the Accesibility Database and the ACM as shown in figure 2(c)
Current Interface (Phase 1)	When all accessibility content was created, Broadcaster (TVC, RBB) goes to ACM and triggers export of all accessibility data to the FTP. ACM will take care of copying files and trigger conversion processes. There is a folder named “output” where all files shall be copied into. 1) Subtitle files are manually copied to the FTP. 2) For AD see 3.4.2 3) For Signer see 3.4.3
Result	• Accessibility files are available on FTP. • Metadata about accessibility files is made available for packaging (to MSE)
Planned Interface (Phase 2)	As Phase 1

3.4.1. Subtitle Processing

Currently this is kept as a blank placeholder as there is no subtitle processing module.

3.4.2. Audio Rendering

Integration Point	Between the Accesibility Database and the ACM as shown in figure 2(c)
Current Interface (Phase 1)	Multiple steps: 1) ACM (ANGLA) triggers audio rendering from ACM. 2) Audio renderer (IRT) processes audio.

	3) Audio renderer copies pre-mixed audio streams (also main mix without AD) to FTP. There is a folder named “output” where the audio files shall be copied into.
Result	• pre-mixed AD audio streams are available on FTP.
Planned Interface (Phase 2)	As Phase 1
Testing Details	Rendering can be triggered with HTTP-request. Renderer will get all AD audio snippets with accompanying metadata and the main audio mix in a zip-package. Json-file is used for setting renderer parameters (incl. output path, naming convention)

3.4.3. Signer Rendering / Processing

Integration Point	Between the Accesibility Database and the ACM as shown in figure 2(c)
Current Interface (Phase 1)	The signer video is manualy uploaded to the server via FTP.
Result	• signer stream(s) is/are available on FTP in folder “output”.
Planned Interface (Phase 2)	The signer content will be automatically uploaded from the ACM

3.5. Publish Content

Integration Point	Between the ACM and content webserver as shown in figure 2(d)
Current Interface (Phase 1)	Broadcaster (TVC/RBB) writes an email to MSE, asking to publish the content.
Result	• publishing process is triggered.

Planned Interface (Phase 2)	This will be an automated process initiated from the ACM

3.6. Packaging

Integration Point	Between the Packager and ACM as shown in figure 2(c)
Current Interface (Phase 1)	From the metadata and assets provided, the packager (MSE) will manually segment and package all files and create an MPD file.
Result	• all files are packaged, MPD is created. Files are on the server, ready for playback.
Planned Interface (Phase 2)	This will be an automated process initiated from the ACM

3.7. Update ImAc portal

Integration Point	Between the ACM and the ImAc portal as shown in figure 2(e)
Current Interface (Phase 1)	MSE sends a json file to the IMAC portal including links to MPD and all required metadata.
Result	• program is published and can be played back via the IMAC-portal.
Planned Interface (Phase 2)	This process will be automated in phase 2

3.8. Integration Summary

During the first phase of ImAc development the focus has been on fully developing the key components, such as the player, the ACM and subtitle editor. As a result many of the behind the scenes processes have remained a manual process, such as publishing content, packaging and updating the ImAc portal.

This approach has allowed the first prototype of the platform to be demonstrated and piloted in order to achieve valuable user feedback. During the second phase of the project the user requirements will be revisited, in order to satisfy the feedback from the first user studies. A more complete platform will be developed integrating and each of these manual processes will be replaced with automated processes.

4. SYSTEM SPECIFICATION

In the architecture design a list of key size and timing requirements where specified. The ImAc platform has been implemented to meet the following system specific requirements:

P1	Production Editors
P2	Accessibility Content Manager
P3	Content Packager and Distribution
P4	Player

4.1. Production Editors

Key sizing and timing requirements for the production editors:

P1.1	Unlimited number of simultaneous web editors in remote stations (it depends only on the hardware capabilities of the server and authorized instances, not limited by software).
P1.2	Instant playback for previewing/verification, only a small buffer on the player when additional video has to be downloaded.
P1.3	Significant delay on the HMD verification playback, all the processes from the content packager have to be triggered.

4.2. Accesibility Content Manager

Key sizing and timing requirements for the accessibility content manager:

P2.1	File and database capacity only limited by hard drive size of the server, not by software.
P2.2	Subtitling files upload/download: no limitation in size (they are manageable files) and fast to transfer.
P2.3	Audio description file upload/download: size only limited by Apache configuration (they can have considerable size), and some time is required.
P2.4	Sign language file upload/download: size limited only by Apache configuration (they are big files), and some time without closing the web browser is required.
P2.5	Video file upload/download: size limited by Apache configuration (they are quite big files), and quite a lot of time without closing the web browser is required.

P2.6	Metadata viewing/editing: fast and without limitation (only authorized users).

4.3. Content Packaging and Distribution

Key sizing and timing requirements for the content packager and distribution:

P3.1	Encoding: One simultaneous stream for main content.
P3.2	Encoding: One simultaneous stream for enhancement.
P3.3	Packaging: One simultaneous stream.
P3.4	Delivery: CDN- unlimited number of players
P3.5	Delivery: Broadcast TS- unlimited number of players

4.4. Player

Key sizing and timing requirements for the player and shared sessions:

P4.1	There is no reason to have multiple active players per consumer device. The software architecture and design of the ImAc player guarantee a proper presentation of the immersive and accessibility contents in a single player on current consumer devices (e.g. smartphones, tablets, HMDs.). It is true even considering the use of web-based technologies, and the limited hardware and software resources of the current consumer devices.
P4.2	The designed solutions to achieve a proper association between the involved devices and a synchronized playback in the session in local scenarios are scalable and lightweight. Thus, the number of simultaneous devices presenting ImAc media contents in local scenarios will be mostly limited by the available bandwidth (e.g. determined by the contract with the Internet Service Provider (ISP).
P4.3	The total number of involved players, in all the active distributed scenarios, will be determined by the scalability of the service provider / broadcaster resources (e.g., Content Manager, CDNs, number of servers...). A proper dimensioning of these resources, together will proper strategies to distribute/balance them will maximize the scalability in terms of number of concurrently ImAc players and sessions.

5. ACCEPTANCE TESTING

Each of the components developed in phase 1, have been tested against the origional requirements, as defined in D2.2. The following table shows which of the components from the user requirements where implemented in Phase 1, those which have not yet been implemented and those which are currently a manual process.

Production Editors	S1	Production of New Subtitles	Implemented in Phase 1
	S2	Subtitle Verification and correction	Not implemented in Phase 1
	S3	Production of Audio Description	Not implemented in Phase 1
	S4	Audio Description Verification and Correction	Not implemented in Phase 1
	S5	Production of Sign Language	Not implemented in Phase 1
	S6	Sign Language Verification and Correction	Not implemented in Phase 1
	S7	Object Based Audio Editing	Not implemented in Phase 1
Accessibility Content Manager	S8	Assignment of videos for the accessibility content production	Implemented in Phase 1
	S9	Getting an Accessibility Content file	Implemented in Phase 1
	S10	Generating Different audio formats with an audio renderer	Manual Process
Content Packaging and Distribution	S11	Preparation of Contents	Manual Process
	S12	Distribution	Manual Process
Player	S13	Consumption of media contents	Implemented in Phase 1

5.1. Acceptence Test summary

Test ID	Test Case Description
S1.1	User accesses the production web page via the web browser.

S1.2	User enters username and password.
S1.3	A window with the list of videos to be subtitled by the user appears.
S1.4	User selects the video to be subtitled and presses the edit button.
S1.5	The web-subtitling editor opens with the video in the preview window and with no subtitles.
S1.6	User creates a new subtitle with all the associated metadata (timecode, character, angle of view, etc.). For that user can use the video player buttons and mouse to move around the video in order to find the entry and exit points of the subtitle, find the character (by video panning) and listen to the corresponding character.
S1.7	The subtitle is simulated over the preview video window.
S1.8	User moves to the next empty subtitle by pressing the corresponding key.
S1.9	User repeats the same process with the rest of the subtitles.
S1.10	User checks the subtitling restrictions by pressing the corresponding button.
S1.11	If a subtitle doesn't comply with the restrictions, a message appears for the user to correct it before continuing with the rest of the checking process.
S1.12	User corrects the subtitle to comply with the restrictions.
S1.13	User repeats the same process until all subtitle items are compliant. In this last case, the simulation button is pressed.
S1.14	User presses the simulation button to verify the final result.
S1.15	User uses the preview video player buttons to move around and see the result accordingly. For instance if the user presses the play button the subtitles will be displayed and deleted over the preview video in their respective times along with the video playback so as to simulate the final result.
S1.16	User can also press the HMD simulation by pressing the corresponding button.
S1.17	User puts on the HMD to watch the final result.
S1.18	User can move up and down the subtitles if specific ones need to be corrected.
S1.19	The subtitling is saved automatically in the Accessibility Content Manager
S2.1	User accesses the production web page via the web browser.
S2.2	User enters username and password.
S2.3	A window with the list of videos to be verified by the user appears.
S2.4	User selects the video to be verified and presses the edit button.
S2.5	The web-subtitling editor opens with the video in the preview window and with the subtitling.
S2.6	User checks the subtitling restrictions by pressing the corresponding button.
S2.7	If a subtitle doesn't comply with the subtitling restrictions a message appears for the user to correct it before continuing with the rest of the checking process.
S2.8	User corrects the subtitle to comply with the subtitling restrictions.
S2.9	User repeats the same process until there's no subtitle that does not comply. In this last case the simulation button activates.
S2.10	User presses the simulation button to verify the final result.

S2.11	User uses the preview video player buttons to move around and see the result accordingly. For instance if the user presses the play button the subtitles will be shown and deleted over the preview video in their respective times along with the video playback so as to simulate the final result.
S2.12	If the subtitling and the video are not in sync, user presses the offset button and enters the right entry time (TCin) that is required for the current subtitle. The same offset is applied to the rest of subtitles times.
S2.13	User presses the simulation button again to verify the final result.
S2.14	User can also press the HMD simulation by pressing the corresponding button.
S2.15	User puts on the HMD to watch the final result.
S2.16	User can move up and down the subtitles if specific ones need to be corrected.
S2.17	The subtitling is updated automatically in the Accessibility Content Manager database.
S3.1	User accesses the production web page via the web browser.
S3.2	User enters username and password.
S3.3	A window with the list of videos to be audio described by the user appears.
S3.4	User selects the video to be audio described and presses the edit button.
S3.5	The web audio description editor opens with the video in the preview window and with no audio description.
S3.6	User creates a new audio description segment with all the associated metadata (TCs, character, angle of view, etc.). The user can use the video player buttons and mouse to move around the video in order to find the entry and exit points of the audio description, find the character (by video panning) and watching the corresponding scene.
S3.7	User moves to the next empty audio description segment by pressing the corresponding key.
S3.8	User repeats the same process with the rest of the audio description segments.
S3.9	User checks the audio description restrictions by pressing the corresponding button.
S3.10	If an audio description segment doesn't comply with the restrictions a message appears for the user to correct it before continuing with the rest of the checking process.
S3.11	User corrects the audio description segment to comply with the restrictions.
S3.12	User repeats the same process until there's no audio description segment that does not comply. In this last case the simulation button activates.
S3.13	User presses the simulation button to verify the final result.
S3.14	User uses the preview video player buttons to move around and see the result accordingly. For instance if the user presses the play button the audio description segments will be playback mixed with the preview video audio in their respective times along with the video playback so as to simulate the final result.
S3.15	User can also press the HMD simulation by pressing the corresponding button.
S3.16	User puts on the HMD device to watch the final result.
S3.17	User can move up and down the audio description segment if specific ones need to be corrected.
S3.18	The audio description is saved automatically in the Accessibility Content Manager database.

S3.19	User might export a json file for description of AD position/gain for the obA Editor together with raw AD audio files.
S4.1	User accesses the production web page via the web browser.
S4.2	User enters username and password.
S4.3	A window with the list of videos to be verified by the user appears.
S4.4	User selects the video to be verified and presses the edit button.
S4.5	The web editor opens with the video in the preview window and with the audio description.
S4.6	User checks the audio descriptions restrictions by pressing the corresponding button.
S4.7	If an audio description doesn't comply with the audio description restrictions a message appears for the user to correct it before continuing with the rest of the checking process.
S4.8	User corrects the audio description to comply with the audio description restrictions.
S4.9	User repeats the same process until there's no audio description that does not comply. In this last case the simulation button activates.
S4.10	User presses the simulation button to verify the final result.
S4.11	User uses the preview video player buttons to move around and see the result accordingly. For instance, if the user presses the play button the audio description will be played and deleted over the preview video in their respective times along with the video playback so as to simulate the final result.
S4.12	If the audio description and the video are not in sync, user presses the offset button and enters the right entry time (TCin) that is required for the current audio description. The same offset is applied to the rest of audio description times.
S4.13	User presses the simulation button again to verify the final result.
S4.14	User can also press the HMD simulation by pressing the corresponding button.
S4.15	User puts on the HMD device to watch the final result.
S4.16	User can move up and down the subtitles if specific ones need to be corrected.
S4.17	The audio description is updated automatically in the Accessibility Content Manager database.
S5.1	User accesses the production web page via the web browser.
S5.2	User enters username and password.
S5.3	A window with the list of videos to be signed by the user appears.
S5.4	User selects the video to be signed and presses the edit button.
S5.5	The web sign language editor opens with the video in the preview window and with no sign language.
S5.6	User creates a new interpreter segment with all the associated metadata (TCs, character, angle of view, etc.). For that user can use the video player buttons and mouse to move around the video in order to find the entry and exit points of the subtitle, find the character (by video panning) and listen to the corresponding character.
S5.7	The interpreter video is shown in an independent window next to the preview video player window.
S5.8	User moves to the next empty interpreter segment by pressing the corresponding key
S5.9	User repeats the same process with the rest of the interpreter segments.

S5.10	User checks the sign language restrictions by pressing the corresponding button.
S5.11	If an interpreter segment doesn't comply with the restrictions, a message appears for the user to correct it before continuing with the rest of the checking process.
S5.12	User corrects the interpreter segment to comply with the restrictions.
S5.13	User repeats the same process until there's no interpreter segment that does not comply. In this last case the simulation button activates.
S5.14	User presses the simulation button to verify the final result.
S5.15	User uses the preview video player buttons to move around and see the result accordingly. For instance if the user presses the play button the interpreter video will be played back in their respective times in its window next to the video playback so as to simulate the final result.
S5.16	User can also press the HMD simulation by pressing the corresponding button.
S5.17	User puts on the HMD device to watch the final result.
S5.18	User can move up and down the interpreter segment if specific ones need to be corrected.
S5.19	The sign language is saved automatically in the Accessibility Content Manager database.
S6.1	User accesses the production web page via the web browser.
S6.2	User enters username and password.
S6.3	A window with the list of videos to be verified by the user appears.
S6.4	User selects the video to be verified and presses the edit button.
S6.5	The web editor opens with the video in the preview window and with the subtitling.
S6.6	User checks the sign language restrictions by pressing the corresponding button.
S6.7	If a sign language doesn't comply with the sign language restrictions a message appears for the user to correct it before continuing with the rest of the checking process.
S6.8	User corrects the sign language to comply with the sign language restrictions.
S6.9	User repeats the same process until there's no sign language that does not comply. In this last case the simulation button activates.
S6.10	User presses the simulation button to verify the final result.
S6.11	User uses the preview video player buttons to move around and see the result accordingly. For instance if the user presses the play button the sign language will be shown and deleted over the preview video in their respective times along with the video playback so as to simulate the final result.
S6.12	If the sign language and the video are not in sync, user presses the offset button and enters the right entry time (TCin) that is required for the current subtitle. The same offset is applied to the rest of sign language times.
S6.13	User presses the simulation button again to verify the final result.
S6.14	User can also press the HMD simulation by pressing the corresponding button.
S6.15	User puts on the HMD to watch the final result.
S6.16	User can move up and down the sign language segment if specific ones need to be corrected.
S6.17	The sign language is updated automatically in the Accessibility Content Manager database.
S7.1	User opens the object-based audio editor.
S7.2	User opens object-based audio scene or creates a new scene.

S7.3	User adds object-based audio objects to the scene (if necessary).
S7.4	User edits audio objects (if necessary).
S7.5	User manually edits or add tracks.
S7.5.1	*User downloads AD track(s) from the Content Manager.*
S7.5.2	*User imports AD track (Result from AD editor, with spatial metadata included).*
S7.5.3	*User exports object-based audio scene.*
S7.6	The spatial metadata is transmitted via a json file.
S7.6.1	*User exports json configuration of AD objects from AD Editor.*
S7.6.2	*User exports raw audio files of AD tracks from AD Editor.*
S7.6.3	*User exports obA Scene file (ADM file) from obA Editor without the AD.*
S7.6.4	*User merges exports from AD Editor and obA Scene into one scene with standalone software.*
S8.1	User accesses the accessibility content manager web page via the web browser.
S8.2	User enters username and password.
S8.3	A window with the list of assets with its corresponding accessibility files appears.
S8.4	User presses the new asset to upload the video to be subtitled, audio described or signed.
S8.5	User selects the new asset, edits metadata and presses the assign production button (subtitling, audio description or sign language).
S8.6	User assigns the corresponding accessibility content production to an operator (professional user that will have to produce the accessibility content).
S8.7	User repeats the process for the rest of videos.
S9.1	User accesses the accessibility content manager web page via the web browser.
S9.2	User enters username and password.
S9.3	A window with the list of assets with its corresponding accessibility files appears.
S9.4	User presses the find button and enters the criteria for the search.
S9.5	A list of assets that comply with the criteria is displayed.
S9.6	User selects the asset and executes the download of the corresponding accessibility content file (subtitling, audio description or sign language).
S10.1	User loads object-based audio scene.
S10.2	User loads settings for rendering process and export.
S10.3	User edits settings for rendering process and export (if necessary).
S10.4	User defines export formats (first order ambisonics, binaural, stereo,...).
S10.5	User starts rendering process.
S11.1	Subtitling, audio description and sign language within the corresponding audio-visual contents are packaged
S11.2	Distribution of the packaged contents to the client players.
S12.1	**VoD: the content will be prepared as files on Content Delivery Network (CDN) storage ready to be distributed on a video on demand request.**

S12.2	**VoD with TV linear distribution: the content will be distributed to be consumed through companion screen synchronously to a linear TV content. In this case, a server may be required to package live or, as a more likely scenario, the content can be pre-packaged to be broadcast at a given time. Then the content is pushed to the CDN (may require authentication tokens which depend on the CDN of the broadcaster).**
S13.1	Scenario 1: User accesses to a website in which a list of accessible 360º video contents (i.e., immersive contents enriched with accessibility assets) is available.
S13.2	Scenario 2: User is watching traditional TV, while it receives a notification about the availability of related immersive and accessible 360º video contents.
S13.3	User could use HMDs, smartphones and tablets for the playback of the immersive + accessible contents in both scenarios. In scenario 1, the use of PCs (laptops, desktops…) is also possible
S13.4	User presses the proper button/link to start the playback of the immersive + accessible contents. The playback can also automatically start in scenario 2.
S13.5	User can enable the accessibility services, which allows an adaptive and personalized presentation of the accessibility assets (subtitles, audio subtitles, sign language, and audio description).
S13.6	User can dynamically activate/deactivate the presentation of the available accessibility assets, as well as to set the available personalisation options for each of them.
S13.7	During playback, user can freely explore the 360º area (e.g., by moving the head, using the touch screen, mouse…), and the presentation of the accessibility assets will be adapted accordingly.
S13.8	In case of multi-screen scenarios, a time-aligned presentation of the media playback across to involved devices will be provided, so coherent sessions can be experienced.
S13.9	When the video playback is finished, user can decide to watch it again, to watch other one, or leave the ImAc accessibility service.

5.2. Acceptence Tests

5.2.1 Production of New Subtitles

Test ID: S1.1

Test Case Description	User accesses the production web page via the web browser.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user starts their web browser and enters the production web page URL ("https://imac.gpac-licensing.com/acm_test/cm/assets.php").

Expected Output	The production web page opens, displaying the ImAc logo, the Accesibility Content Manager title and a username and password entry box.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.2

Test Case Description	User enters username and password.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user enters their username ("test") in the username text input box and password ("test") into the password text input box. The user then either clicks on the 'Log in' button or presses the enter key.
Expected Output	If the username and password are valid, then the user is logged in and taken to the Asset list. If the username and password are invalid a warning is presented to the user and they are taken no further.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.3

Test Case Description	A window with the list of videos to be subtitled by the user appears.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The landing page presented to the user is an 'Asset List'.
Expected Output	On this page the user can see all of the video assets that they have access to, and select individual videos. Each video shows its unique AssetID

	number, and shows whether the asset contains video, subtitles, audio description and sign language.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.4

Test Case Description	User selects the video to be subtitled and presses the edit button.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user clicks on the subtitle button relating to a video asset. The user then clicks on the edit button in order to open the subtitle editor.
Expected Output	Once the user clicks on the subtitle button for a video asset if a subtitle asset exists and is related to the video a dialogue box displays a metadata description of the the subtitle file. If the subtitle file does not yet exist the dialogue box allows the user to create a new subtitle file, or upload an exiting one. Provided a subtitle file exists the edit button in avaliable.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.5

Test Case Description	The web-subtitling editor opens with the video in the preview window and with no subtitles.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user clicks on the subtitle edit button.
Expected	The web based subtitle editor starts. If a subtitle file already exists for the

Output	video asset then it is populated with the existing subtitles. If not a blank subtitle editor is opened.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.6

Test Case Description	User creates a new subtitle with all the associated metadata (timecode, character, angle of view, etc.). For that user can use the video player buttons and mouse to move around the video in order to find the entry and exit points of the subtitle, find the character (by video panning) and listen to the corresponding character.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user is able to interact with the video with the pause, play, frame advance buttons and scrubbing by clicking on a timeline. The 360 degree video can be manipulated in order to explore the location for the subtitle. Text is typed into the subtitle box to create a subtitle. The user can scrub the video to the subtitle start and end point by using the 'In' and 'Out' buttons.
Expected Output	Once the text is typed into the subtitle box it appears on the video. The 'In' and 'Out' buttons update the time for the subtitle and allows the subtitle to be aligned with the video when it I is played. Manipulating the users view on the 360 degree video also updates the desired angle for the subtitle location.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.7

Test Case Description	The subtitle is simulated over the preview video window.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU

	@4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user selects a preview mode, from a radio button list including 'edit', 'forced preview' or 'free preview'. The video preview respects this mode when the video is played back.
Expected Output	In edit mode the user is able to freely move within the subtitles. In 'forced preview' mode the video always rotates to the angle related to each subtitle as it plays. In free preview' mode the video always displays the subtitles at the correct timings, but allows the user to arbitarily rotate the video.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.8

Test Case Description	User moves to the next empty subtitle by pressing the corresponding key.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user either clicks on the 'Insert subtitle' or plesses the 'Page Up' button.
Expected Output	A new blank subtitle in inserted into the file.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.9

Test Case Description	User repeats the same process with the rest of the subtitles.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)

Input	The user repeats S1.6 - S1.8 for an arbitary number of subtitles.
Expected Output	The subtitle editor allows each of the subsequent subtitles to be added to the stack of subtitle.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.10

Test Case Description	User checks the subtitling restrictions by pressing the corresponding button.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user presses a button to button check that the subtitle complies with defined restrictions.
Expected Output	A dialog box appears with metrics relating to the defined restrictions.
Actual Output	The restriction information is presented directly in the editor. When a subtitle is selected the remaining characters and reading speed are presented graphically alongside the subtitle box.
Component Version	Version 27
Result	Pass

Test ID: S1.11

Test Case Description	If a subtitle doesn't comply with the restrictions, a message appears for the user to correct it before continuing with the rest of the checking process.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	Any non complience to a subtitle restriction is highlighted in the editor.
Expected	When a subtitle is either too long, or the reading speed is too fast, the editor

Output	displays a warning. When increasing the letter count, or reading speed the traffic light warning changes from green, through amber to red.
Actual Output	The warning is presented in the form of a traffic light system direclty in the editor pane.
Component Version	Version 27
Result	Pass

Test ID: S1.12

Test Case Description	User corrects the subtitle to comply with the restrictions.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user edits either the character count, start or end time and the warning system guides the user to bring the subtitle properties back inline with the restrictions.
Expected Output	When the user reduces the character count, or reading speed the warning indicator turns green.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.13

Test Case Description	User repeats the same process until all subtitle items are compliant. In this last case, the simulation button is pressed.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	Each subtitle is individually selected.
Expected Output	The warning is displayed for each subtitle and each subtitle can be updated to comply with the restrictions.

Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.14

Test Case Description	User presses the simulation button to verify the final result.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user selects forced or free preview mode.
Expected Output	The video is played back on a preview window.
Actual Output	As expected.
Latest Test	7th November 2018
Result	Pass

Test ID: S1.15

Test Case Description	User uses the preview video player buttons to move around and see the result accordingly. For instance if the user presses the play button the subtitles will be displayed and deleted over the preview video in their respective times along with the video playback so as to simulate the final result.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user watched the video using the preview video player. The video can be played in either forced preview or free preview mode. The video interaction buttons allow the video to be played, paused etc as would be expected from any video player.
Expected Output	The 360 video is played back in the preview window and controlled with the video buttons. The subtitles are displayed correctly. In forced preview mode the video jumps to the correct angle for the subtitle. In free preview mode the

	user can drag the video to change the view and the location aid provides a directional indicator.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.16

Test Case Description	User can also press the HMD simulation by pressing the corresponding button.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user pressed the simulation button.
Expected Output	The video is played back on an HMD.
Actual Output	Not currently implemented.
Component Version	Version 27
Result	N/A

Test ID: S1.17

Test Case Description	User puts on the HMD to watch the final result.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user puts on the HMD.
Expected Output	The video is played back on an HMD.
Actual Output	Not currently implemented.
Component	Version 27

Version	
Result	N/A

Test ID: S1.18

Test Case Description	User can move up and down the subtitles if specific ones need to be corrected.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user either selects a specific subtitle from the list, or pressed, the next or previous button.
Expected Output	The specific subtitle is brought back into focus.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S1.19

Test Case Description	The subtitling is saved automatically in the Accessibility Content Manager
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user leaves the editor, by either closing the window, pressing an exit link with or without pressing save first.
Expected Output	The current subtitle file is saved to the server.
Actual Output	If the browser is forced to close the latest changes are lost. The user has to click on the save button to ensure that the latest changes are stored.
Component Version	Version 27
Result	Pass

5.2.2 Assignment of videos for the accessibility content production

Test ID: S8.1

Test Case Description	User accesses the accessibility content manager web page via the web browser.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user starts their web browser and enters the production web page URL ("https://imac.gpac-licensing.com/acm_test/cm/assets.php").
Expected Output	The production web page opens, displaying the ImAc logo, the Accesibility Content Manager title and a username and password entry box.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S8.2

Test Case Description	User enters username and password.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user enters their username ("test") in the username text input box and password ("test") into the password text input box. The user then either clicks on the 'Log in' button or presses the enter key.
Expected Output	If the username and password are valid, then the user is logged in and taken to the Asset list. If the username and password are invalid a warning is presented to the user and they are taken no further.
Actual Output	As expected.
Latest Test	Version 27
Result	Pass

Test ID: S8.3

Test Case Description	A window with the list of assets with its corresponding accessibility files appears.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user arrives at the ACM landing page.
Expected Output	The landing page contains a graphical representaion of each of the video assets that they have permission to view. This includes an ID, Language, Statusm abd filename. It also shows which accesibility assets have been generated to relate to each asset, and contains links to the editors.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S8.4

Test Case Description	User presses the new asset to upload the video to be subtitled, audio described or signed.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user presses the 'new asset' button.
Expected Output	A dialog appears allowing the user to upload a video file and create the initial meta data such as the title and description.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S8.5

Test Case Description	User selects the new asset, edits metadata and presses the assign production button (subtitling, audio description or sign language).

Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user clicks on either an assignment button (for either subtitling, AD or sign language) or selects the video and selects the appropriate tab on the right hand information pane.
Expected Output	The user is given an otion to eithe upload, or create a new accessibility file if one does not yet exist, or use the online editor to edit the file that does exist.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S8.6

Test Case Description	User assigns the corresponding accessibility content production to an operator (professional user that will have to produce the accessibility content).
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	In the information pane for either of the accesibility files, one or more accesibility editors can be assigned to the asset by clicking on the plus button.
Expected Output	Clicking on the add button provides a list of all avalibale users, any of which can be added to the list uf assigned users. Clicking on the cross allows users to be removed.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S8.7

Test Case Description	User repeats the process for the rest of videos.

Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The process S8.4 - S8.6 can be repreated for any number of files and assets.
Expected Output	The user is able to successfully switch between files and assets.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

5.2.3 Getting an Accessibility Content file

Test ID: S9.1

Test Case Description	User accesses the accessibility content manager web page via the web browser.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user starts their web browser and enters the production web page URL ("https://imac.gpac-licensing.com/acm_test/cm/assets.php").
Expected Output	The production web page opens, displaying the ImAc logo, the Accesibility Content Manager title and a username and password entry box.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S9.2

Test Case Description	User enters username and password.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)

Input	The user enters their username ("test") in the username text input box and password ("test") into the password text input box. The user then either clicks on the 'Log in' button or presses the enter key.
Expected Output	If the username and password are valid, then the user is logged in and taken to the Asset list. If the username and password are invalid a warning is presented to the user and they are taken no further.
Actual Output	As expected.
Latest Test	7th November 2018
Result	Pass

Test ID: S9.3

Test Case Description	A window with the list of assets with its corresponding accessibility files appears.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user arrives at the ACM landing page.
Expected Output	The landing page contains a graphical representaion of each of the video assets that they have permission to view. This includes an ID, Language, Statusm abd filename. It also shows which accesibility assets have been generated to relate to each asset, and contains links to the editors.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S9.4

Test Case Description	User presses the find button and enters the criteria for the search.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user types a search criteria into the search text box at the top of the page

	and presses search.
Expected Output	The user is presented with a reduced list of asssets where the title contains the search string.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S9.5

Test Case Description	A list of assets that comply with the criteria is displayed.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The search landing page is presented.
Expected Output	The user is presented with a reduced list of asssets where the title contains the search string.
Actual Output	As expected.
Component Version	Version 27
Result	Pass

Test ID: S9.6

Test Case Description	User selects the asset and executes the download of the corresponding accessibility content file (subtitling, audio description or sign language).
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	For any selected asset the user is able to download the corosponding accesibility files.
Expected Output	From the information pane the user is able to select and download any of the accesibility files which exist.

Actual Output	As expected.
Component Version	Version 27
Result	Pass

5.2.4 Consumption of media contents

Test ID: S13.1

Test Case Description	Scenario 1: User accesses to a website in which a list of accessible 360º video contents (i.e., immersive contents enriched with accessibility assets) is available.
Setup	Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102)
Input	The user visits the ImAc player launch page (http://84.88.32.46/imacplayerv5/).
Expected Output	The webpage presents the user with a selection of the 360º video assets that they have access to, along with a preview image and title. The user is also presented with a choice of Menu type (Traditional or Enhances-Accesibility), language and subtitle format.
Actual Output	As Expected.
Component Version	Version 5
Result	Pass

Test ID: S13.2

Test Case Description	Scenario 2: User is watching traditional TV, while it receives a notification about the availability of related immersive and accessible 360º video contents.
Setup	HbbTV
Input	The user acceses the ImAc player through a smart TV
Comments	This scenario is not supported by the first phase of the ImAc Project.
Component	Version 5

Version	

Test ID: S13.3

Test Case Description	User could use HMDs, smartphones and tablets for the playback of the immersive + accessible contents in both scenarios. In scenario 1, the use of PCs (laptops, desktops…) is also possible
Setup	1) Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102) 2) Occulus Go
Input	Selecting a video from the ImAc portal on either a desktop computer or HMD opens the video.
Expected Output	The video opens full screen and allows the user to interact with the video, either by looking around (in the case of a HMD) or by clicking and dragging with the mouse (on a desktop Computer). The video should start playing automatically and should behave fully immesrively to the user. For example, if using a HMD there should be no delay or lag when moveing the view.
Actual Output	As expected.
Component Version	Version 5
Result	Pass

Test ID: S13.4

Test Case Description	User presses the proper button/link to start the playback of the immersive + accessible contents. The playback can also automatically start in scenario 2.
Setup	1) Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102) 2) Occulus Go
Input	The user is able to interact with the playing video, through the menu provided. The menu is accesed in a number of ways depending upon whether the user is using the traditional menu or enhanced accesibility menu. Using a HMD the user looks at the menu button for a few seconds to activate the menu.

Expected Output	When looking at the menu buton (or clicking on the menu button on a desktop computer) the menu should be activated and display.
Actual Output	As expected.
Component Version	Version 5
Result	Pass

Test ID: S13.5

Test Case Description	User can enable the accessibility services, which allows an adaptive and personalized presentation of the accessibility assets (subtitles, audio subtitles, sign language, and audio description).
Setup	1) Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102) 2) Occulus Go
Input	The user can navigate the menu in order to enable or disable any of the accesibility services.
Expected Output	Switching any of the accesibility services (subtitles, AD, signer) on or off in the menu should enable or disable its playback in the video, if that service is avalible.
Actual Output	As expected.
Component Version	Version 5
Result	Pass

Test ID: S13.6

Test Case Description	User can dynamically activate/deactivate the presentation of the available accessibility assets, as well as to set the available personalisation options for each of them.
Setup	1) Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102) 2) Occulus Go

Input	The user can navigate the menu in order to enable or disable any of the accesibility services.
Expected Output	Switching any of the accesibility services (subtitles, AD, signer) on or off in the menu should enable or disable its playback in the video, if that service is avalible.
Actual Output	As expected.
Component Version	Version 5
Result	Pass

Test ID: S13.7

Test Case Description	During playback, user can freely explore the 360º area (e.g., by moving the head, using the touch screen, mouse…), and the presentation of the accessibility assets will be adapted accordingly.
Setup	1) Clean PC - (Windows 10, 32Gb Ram, 64Bit Intel Core i7-7700K CPU @4.2Ghz), Google Chrome (version 70.0.3538.102) 2) Occulus Go
Input	When using a HMD the user should be able to look around the scene and feel fully immersed in the content. On a desktp computer the same result should be achievable by clicking and dragging the mouse.
Expected Output	The environment should respond correctly and in real-time, in order to preserve the illusion of immersion and avoid motion sickness.
Actual Output	As expected.
Component Version	Version 5
Result	Pass

6. CONCLUSIONS

This document contains the first iteration of the integration and testing report for the ImAc platform. It aims to evaluate the system as defined in the Technical Architecture (D3.1) and this document describes how each of these components are integrated as well as the current status of this their integration. It also holds the platform accountable to the requirements gathered in (D2.3).

In Chapter 2 we described our approach to the methodology that we employ for testing the ImAc platform. There we described how the each component would be initialy tested independently by each developed in isolation (unit testing). The methodology also describes the System specification which the platform meets, the workflow and integration of each of the components and their current status. Finally we described how the system would be held to account against the initial user requirements in terms of acceptence testing.

In Chapter 3 we described the workflow of the platform and identify the Integration points within the ImAc project. This showed the current status of each component, which components have currently been implemented, which are implemented which manual steps and which components will be developed in the second part of the project.

Chapter 5 provides the results of acceptance testing of the avaliable components from the platform. Although the user requirements have evolved significantly and features have changed our first acceptence testing shows that the components required for the first round of user testing are fit for purpose and work in line with the initial user requirements. Moving into the next phase of the project the current work flow and integration structure will be used to redevelop the Architecture design, and update the testing plan to meet the updated set of user requirements.