

Deliverable

Project Acronym:	IMAC
Grant Agreement number:	761974
Project Title:	<i>Immersive Accessibility</i>



D3.2 – Accessibility Content Manager

Revision: 2.4

Authors: Enric Torres Feixas & Kimiasadat Mirehbar (Anglatècnic)

Delivery date: M22 (09-07-2019)

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

Dissemination Level

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

Abstract:

This document is meant to address the concepts, progress made during the project, vital relevant information, installation and access guide and a user manual related to ImAc Accessibility Content Manager (ACM).

REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	26-06-2018	Enric Torres	ANGLA	Template and ToC
0.2	27-06-2018	Enric Torres	ANGLA	First version
0.3	27-06-2018	Enric Torres	ANGLA	Second version
0.4	28-06-2018	Enric Torres	ANGLA	Final version
1.0	27-11-2018	Kimia Mirehbar	ANGLA	Final version
1.1	06-03-2019	Kimia Mirehbar	ANGLA	Template and TOC – second iteration
2.0	30-05-2019	Kimia Mirehbar	ANGLA	Second iteration
2.1	31-05-2019	Enric Torres	ANGLA	Revision and contributions for the second iteration
2.2	14-06-2019	Kimia Mirehbar & Enric Torres	ANGLA	Primary Final version
2.3	02-07-2019	Enric Torres & Kimia Mirehbar & Alex Soler	ANGLA	Modifications after partners reviews
2.4	09-07-2019	Kimia Mirehbar & Enric Torres	ANGLA	Modifications after partners second reviews

Disclaimer

The information, documentation and illustrations available in this deliverable, is written by the IMAC – project consortium under EC grant agreement H2020-ICT-2016-2 761974 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Statement of originality:

This document contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

EXECUTIVE SUMMARY

ImAc project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 761974 and is looking for development of solutions that facilitate the procedure of production, transmission and presentation of accessibility services on immersive media.

One of the main objectives – among many – of ImAc is a comprehensive environment for professional users with the aim to manage and control the workflow of accessibility services production and distribution.

This objective is achieved in the third work package of ImAc (second task of WP3). The goal of this task is to develop a central control of the system (ACM or Accessibility Content Manager). This control system is required to act as:

- A centralized management and processing system of access services such as subtitles, audio description and sign language
- A centralized system which handles and manages the necessary communication and notification among various levels of professional users of the ACM. The levels of users correspond to their role in the access service production and distribution workflow and typically are categorized as system administrators, managers, producers, providers and reviewers. This categorization is not unique and can get extended as required by the broadcaster.

Stakeholders of this environment are broadcasters and access service providers or professionals who wish to use this tool to manage and produce access services. This document is addressed to stakeholders who wish to use this tool or professionals who are interested in getting to know ACM.

The existence of a platform like ACM is essential for projects that wish to offer access services due to below facts:

- One of the stakeholders of the environment are professional broadcasters. When dealing with professional use, the number of programmes which require access services increases accordingly, this increase results in rising the volume of 360 access services to be produced for each one. Hence, a centralized management and processing system is required for the purpose of production and distribution.
- High volume of 360 access services for each programme and works to be produced. Hence an accessibility content manager is required for professional users.
- It is significant to note that access service production is apart from the post-production of the video itself. This means the procedure of access service production is operated in a parallel different circuit accompanied by Low Quality video, not the main one, and also supporting various number of service providers and maybe also broadcasters.

The ACM will be beneficial to broadcasters in two main aspects:

- Production:
 - Assign the production of access service files to producers (usually an external

service provider or professional).

- Send a Low Quality 360 degree video file to the producer so they can carry out the access service production.
- Cataloguing:
 - Receive the access service files from the producers and verifying that they are correct for broadcasting or publication.
 - Catalogue the verified files for when they are required by the 360 degree programme for broadcasting or publication.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Enric	Torres Feixas	ANGLA	enric@anglatecnic.com
Kimiasadat	Mirehbar	ANGLA	kimia@anglatecnic.com
Àlex	Soler de Ferrater	ANGLA	alex@anglatecnic.com

CONTENTS

Revision history	2
Executive summary	3
Contributors	5
Contents	6
Illustrations and tables	9
List of acronym	11
1. Introduction	12
1.1. Purpose of this document	12
1.2. Scope of this document	12
1.3. Status of this document and structure	12
1.4. Relation with other ImAc works	12
2. Component description and primary concepts	14
2.1. General Description	14
2.1.1. Stakeholders	14
2.1.2. Objectives	15
2.2. Architecture	16
2.2.1. SM (System Management interface)	16
2.2.2. CM (Content Management interface)	17
2.2.3. ED (Editor interface)	17
2.2.4. How components serve the complete workflow	18
3. How to	20
3.1. Introduction	20
3.2. How to install the ACM code	20
3.3. How to access the ACM interfaces	24
4. User manual	25

4.1. System Management interface	25
4.1.1. Users	25
4.1.2. Profiles	27
4.1.3. Languages	29
4.1.4. System	30
4.1.5. Logs	33
4.1.6. Scenario	34
4.2. Content Management interface	34
4.2.1. Assets	34
4.2.1.1. Asset folders	35
4.2.1.2. Assets main page	35
4.2.1.3. File view	37
4.2.1.4. Asset form	38
4.2.2. Bin	43
4.2.3. Reports	43
4.2.4. User scenario	44
4.3. Editor interface	45
4.3.1. Main page	45
4.3.2. User scenario	45
5. ACM access service file formats	47
5.1. Subtitle import and export	47
5.2. Audio description import and export	47
5.3. Sign language import and export	47
6. ACM background processes	48
6.1. Video transcoding to low quality process	48
6.1.1. XML Flag File	48
6.1.2. ImaCron script	50

6.1.3. InsertLow	50
6.2. Audio rendering process	51
6.2.1. Audio rendering request	51
6.2.2. Notification to the Cloud Renderer	52
6.2.3. Cloud Renderer report	53
6.3. Packaging process	58
6.3.1. Requirements for packaging	60
6.3.2. JSON structure sent to Packager	60
6.3.3. Packager response	65
Annex I: Technical terms	66
Annex II: ACM background processes	67
References	68

ILLUSTRATIONS AND TABLES

Illustration 1: Dependencies between tasks and deliverables of ImAc	13
Illustration 2: Parties relation & characteristics	15
Illustration 3: System Management interface Architecture	17
Illustration 4: Content Management interface architecture	17
Illustration 5: Steps for the production of content for 360 media access services	19
Illustration 6: ACM login page	24
Illustration 7: User edition/creation form	27
Illustration 8: Language creation/edition form	30
Illustration 9: Template edition/creation form	33
Illustration 10: Creating the “Subtitle manager” profile	34
Illustration 11: Folders hierarchical order	35
Illustration 12: Reports	44
Illustration 13: Import asset from ImAc FTP	49
Illustration 14: Request for publish	58
Illustration 15: Prepack	59
Table 1: Users list	26
Table 2: Profiles	28
Table 3: Profile creation form	29
Table 4: Languages list	30
Table 5: Variables of system	31
Table 6: Engines of the system	32
Table 7: Notification templates list	32
Table 8: Logs – SM	33
Table 9: Assets list	36

Table 10: File view	38
Table 11: Asset form	39
Table 12: Asset form – video properties	40
Table 13: Asset AD properties	43
Table 14: Task list via ED	45

LIST OF ACRONYM

Acronym	Description
ACM	Accessibility Content Manager
AD	Audio Description
AST	Audio Subtitles
ST	Subtitles
SL	Sign Language
SM	System Management interface
CM	Content Management interface
FTP	File Transfer Protocol
MAM	Media Asset Management
LQ	Low Quality
HQ	High Quality
TC	Time Code
VoD	Video on Demand
SSH	Secure Shell
JSON	JavaScript Object Notation
VoG	Voice of God
URL	Uniform Resource Locator
RAM	Random-access memory
FoA	First-order Ambisonics
OS	Operating System

1. INTRODUCTION

1.1. PURPOSE OF THIS DOCUMENT

One of the objectives – among many – of ImAc is to create a high level production and management system for immersive media access services with goal to control the whole production/management workflow.

This deliverable presents a comprehensive description of the Accessibility Content Manager (ACM) developed under the framework of T3.2 “Accessibility Content Manager” of the ImAc project. The goal of this task is to develop a central control of the system where ImAc contents such as subtitles, sign language and audio description files are available to users and external modules. In Annex I there is the description of some technical terms that will be used along this document.

This task is specifically aimed to “professional users” who ensure the standard process of access service files production and management, hence this deliverable addresses the functionality of the developed Accessibility Content Manager (ACM) tool in the project’s second iteration.

1.2. SCOPE OF THIS DOCUMENT

This deliverable is focused on giving a bright view to professional users of ACM. It is presented alongside the newest version of ACM software and has an informative role. Also the background processes in ACM are explained in detail.

Before using ACM for managing purposes, It is recommended to read this document completely (except chapter 6 where the background processes are explained).

1.3. STATUS OF THIS DOCUMENT AND STRUCTURE

This document is the second and last version of this deliverable which presents the second and last iteration developments on this issue and it is foreseen to be published in M19.

The first version of this document was submitted in July 2018 with the progress made in ACM in the first iteration.

Moreover in this report, during first chapter, an introduction is presented. The introduction is followed by a presentation of architecture and a brief description in chapter 2. The next chapter is a guide explaining the installation process and user access to ACM followed by chapter 4 with a detailed user manual for professional users. Chapter 5 explains the access service files that can be imported and exported to and from ACM followed by a detailed description of the background processes in chapter 6.

1.4. RELATION WITH OTHER ImAc WORKS

Illustration 1 demonstrates the interconnection between various tasks of ImAc. T3.2 is influenced from T3.1 (architecture design). The result of T3.2 is presented alongside the ACM software in D3.2 and would act as an input/result for other ImAc tasks specially WP4 tasks. In WP4 the access service production tools are developed and they get their resources from ACM and their structure is relevant and influential for each other.

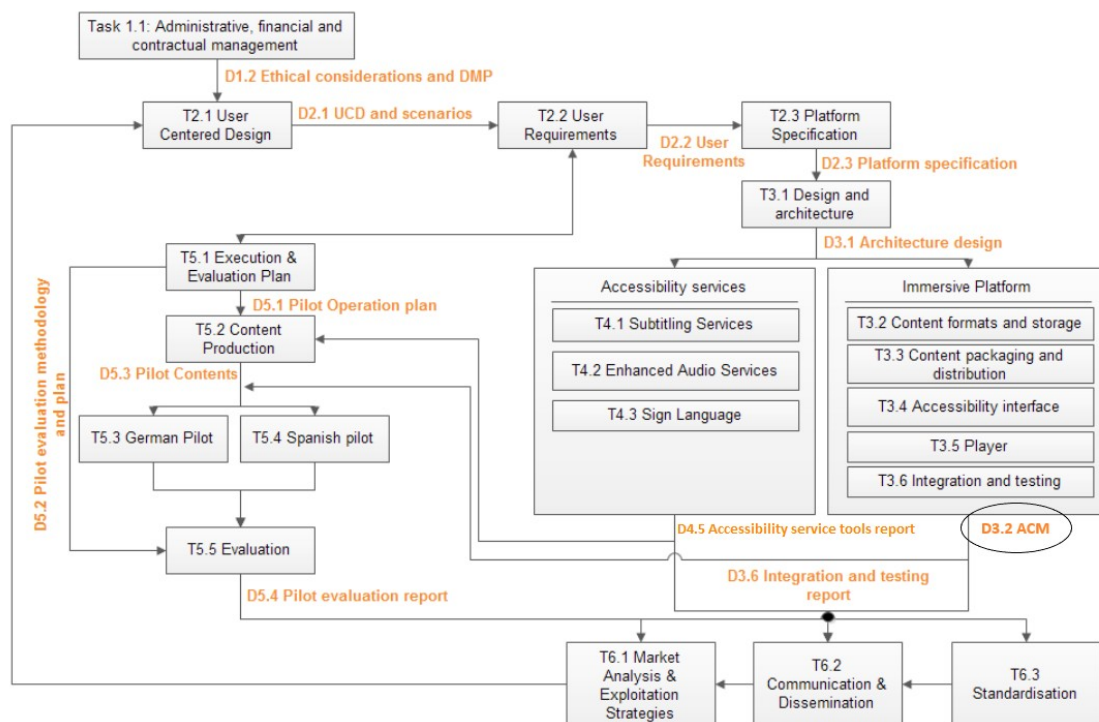


Illustration 1: Dependencies between tasks and deliverables of ImAc

2. COMPONENT DESCRIPTION AND PRIMARY CONCEPTS

ImAc evolves around offering 360 access services to interested second parties. The reasoning of why a central software is needed in order to produce and manage the access services was presented in Executive Summary. This necessity is true for every type of audiovisual media, however ImAc puts emphasis on 360 media due to the fact that the 360 media industry is rapidly growing and lack of a system to serve as a tool for management, production, distribution and delivery of access service files for this type of media (360).

In this chapter it is described how the ACM components are working with each other, to whom they are beneficial and how the architecture is compatible with user requirements deduced from previous tasks. The whole ImAc architecture is based on professional end users requirements gathered during the first and second phases of the ImAc project, as well as on the know-how and lessons learned from the tasks and tests performed during the project lifetime. Also there has been a special attention paid to the common practices and infrastructures available in today's broadcaster world while defining the components.

This section starts with a general description and primary concepts and finishes with an architecture exploration.

2.1. GENERAL DESCRIPTION

As stated before, T3.2 is looking to develop a complete and robust ACM, where all videos for production along with their access service contents are available, also it is a place where all automatic processes and notifications occur.

In Executive summary a brief light was shed on the stakeholders. Here, the stakeholders are defined with more details in 2.1.1 followed by previously scheduled objectives.

2.1.1. Stakeholders

It is necessary to recognise the professional users who are potential consumers of the ACM. Illustration 2, displays the stakeholders alongside the beneficial factors offered to them by ACM which are: security, API management, settings, stakeholders interconnection, user dashboard and notification and messaging. The cloud symbol represents the fact that ACM is a web-based interface (reasons explained next in this chapter).

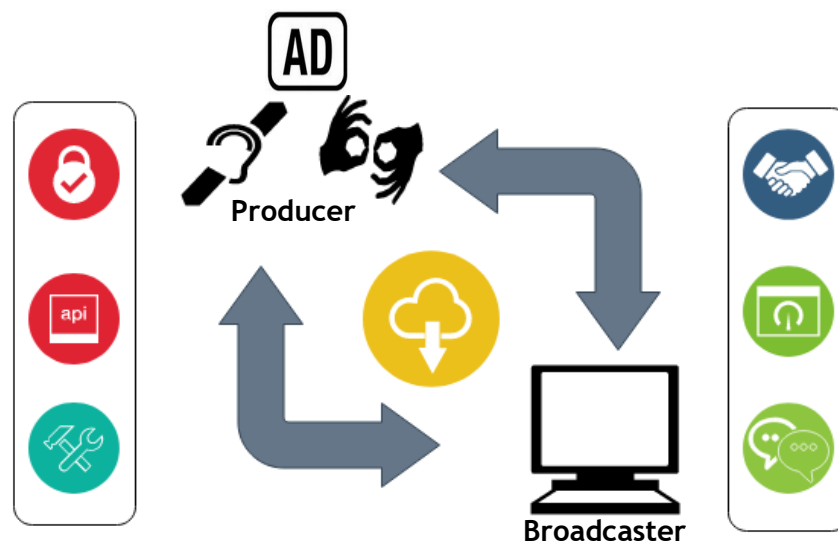


Illustration 2: Parties relation & characteristics

- **Broadcasters:** It refers to classic broadcasters who wish to offer 360° media to their audience and they need access services for the programmes. Typically broadcasting stations are comprised of numerous departments related to access services such as: accessibility department, quality assurance department, informatics etc.
- **Producers:** The term refers to educated professional people who are certified to produce access service files for programmes such as: subtitlers, audio describers and signers (sign interpreters). Producers can be part of the broadcasting station as a whole department or outsourced to an external entity.

2.1.2. Objectives

According to the roadmap, up to this date the latest version of the ACM web interfaces have been developed. The most important reasons of why ACM is presented as web interface software are that the web-based components saves the computational load and are reachable by users independent from their OS (Operating System) and version, device type and also they eliminate complicated hardware installations. The full reasoning of why web-based interface serves such tool better, can be found in ImAc D2.3ⁱ where the platform specifications are explained. Further details on the matter are skipped here in order to present the document as concise as possible.

This web-based interface achieves the below scheduled objectives:

- To access in a secured way with user login identification, assigning the correct profile and platform right access for every user.
- To allow uploading HQ immersive videos and assigning the metadata needed for the production of access service contents such as subtitling, audio description and sign language. Although the video uploading process occurs with HQ videos, LQ video is generated automatically by ACM from the HQ version and made available to producers

in order to reduce the volume and the fact that the existence of HQ video is absolutely not required while managing and producing access service files, and usage of an LQ video benefits the users in the sense of internet bandwidth and copyrights legalities. Additionally, it is necessary that the LQ video is compatible with the browsers where the web editors will be executed.

- To allow uploading and downloading access service files and communication with the corresponding web editors for its production.
- To catalogue all the contents in Assets.
- To access to settings and configurations in order to have a standard environment for system administrative purposes.
- To access to statistical data and logs about the use of the platform.
- To allow to standardize the interconnection between different interfaces alongside with appropriate user permissions and tools.

2.2. ARCHITECTURE

The ACM GUI structure is comprised of four sections or web interfaces. For the purpose of management and control, three of these four sections have been identified and developed in T3.2, hence only these three are explained in this document. The fourth one is addressed in ImAc D3.3 which addresses “Prepack” and it contains information about distribution rather than management that is concern of the present deliverable.

Every section of ACM is destined to a type of stakeholder who would benefit from using ACM in their profession. Next, these particular sections and their purpose will be described. After elaboration of sections separately, it is described how these sections serve the professional user in the complete workflow of access service file production, management, transmission and delivery.

1. SM (System Management interface)
2. CM (Content Management interface)
3. ED (Editor interface)

2.2.1. SM (System Management interface)

ACM requires an interface for system administrative tasks. This requirement has been taken into account by developing SM interface. SM is destined to platform administrators and broadcasters who wish to set and configure some specific settings related to access service and also IT management of the platform.

Illustration 3 shows the sections which SM is comprised of. In chapter “User Manual”, reader will go through every section with detailed explanation of the functionalities.

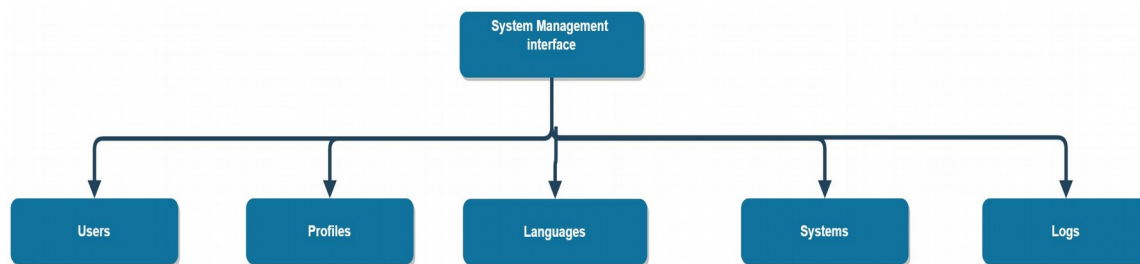


Illustration 3: System Management interface Architecture

2.2.2. CM (Content Management interface)

CM interface is directly destined to broadcasters and professional users who have duties such as:

- Creation/edition of assets
- Organization and distribution of production tasks among access service providers
- Management and control of access service workflow
- Reports on access services

CM is a place with a focus on the management of both the contents needed for the production of the access service files as well as the management of the results of the production.

Illustration 4, demonstrates the structure of CM.

The detailed functionality of CM will be presented in chapter “User manual”.

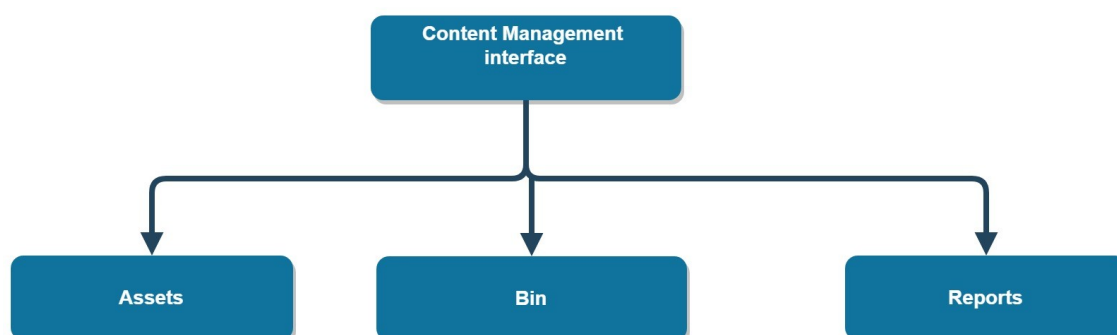


Illustration 4: Content Management interface architecture

2.2.3. ED (Editor interface)

ED is the interface which allows access service producers to manage their production tasks and interact with the ImAc editor tools. When access service production tasks are assigned to

producers, they can search, list, view and organize their tasks using the ED.

ED provides a direct access to ImAc editors. When the access service production task is finished, the producer is able to change the status of the task (only to “Completed”) which will need a final revision by a professional reviewer user (normally, broadcaster employee who works in the quality assurance department) who has the rights to consider the access service content ready for delivery. Chapter 4 will provide a more detailed explanation about the ED.

2.2.4. How components serve the complete workflow

After going through each cell separately, a complete use case where ACM is integrated in a real broadcasting environment is explained in this chapter in order to shed light on the role of ACM during the whole procedure of managing and producing 360 access services.

It is recommended to read the following points and see the graph of Illustration 5 simultaneously.

1. When the broadcaster indicates in their systems that one or more access services are required in a 360 programme, the MAM (Media Asset Management) system or video digitization system (where the master media for the 360 programmes reside) must, automatically if possible, create a copy of the master 360 media into a LQ 360 video file and an additional file with informative metadata of the 360 programme and requests. More complete information of the procedure of transcoding an HQ video into LQ is presented in D3.3. It is noteworthy to mention that the reason that LQ video is chosen to work with during the procedure is explained earlier in the document.
2. The ACM platform automatically receives the LQ video file with the corresponding metadata and will create a new Asset in the platform for the 360 programme. The Asset will be accessible from the Content Management (CM) interface and will contain all the files and metadata of the programme, received by both the broadcaster and professional access service files providers.
3. The broadcaster’s accessibility department team accesses the Content Management interface of the ImAc platform to manage production workflow. The term “management” refers to checking the general information, making sure the metadata is right by viewing them, and finally assigning the task to produce the contents of each access service file (ST, AD, SL) to a specific producer (employees or external service providers) who receives a notification on the matter. The production tasks are varied based on different factors such as access service file type (ST, AD, SL) and language. These tasks are assigned to different producers based on their abilities (for instance, whether the producer speaks a certain language or whether he/she is a subtitler or a signer).
4. On the other side at ED (Editor interface), the producer is able to see the production tasks which have been assigned to them previously from CM. The status of the task changes automatically from “Pending” to “In progress” when the producer starts the producing of the access service (possible status are: Pending, In progress, Completed, Verified, Rejected). The only status that the producer can change manually is it to “Completed” when the production work assigned in the task is finished. This step also makes come true another objective of ACM stated earlier in the document.

5. The work done by the producer is mainly executed on other ImAc tools (editors) which are not main concern of this deliverable and they are discussed in WP4. Hence they are skipped here. If the reader needs information on the ImAc editors, they can find it in D4.1ⁱⁱ, D4.2ⁱⁱⁱ and D4.3^{iv} which address the ST, AD and SL editors respectively.
6. The broadcaster's quality assurance team accesses the CM interface of the ImAc platform to verify and validate the productions. To do this, they use the CM view that shows a list of files with status “completed”, from here they can execute the preview and validate (change status to “Verified”) or reject (change status to “Rejected”) the access service file. If rejected the task for the producer is created again for them to work on the access service file in order to be favourable to quality assurance team.
7. Finally, when the access service files (ST, AD, SL) have the status “Verified”, they are available when required by the broadcaster’s playout during the broadcasting or publication on their websites for VoD of the immersive programme.

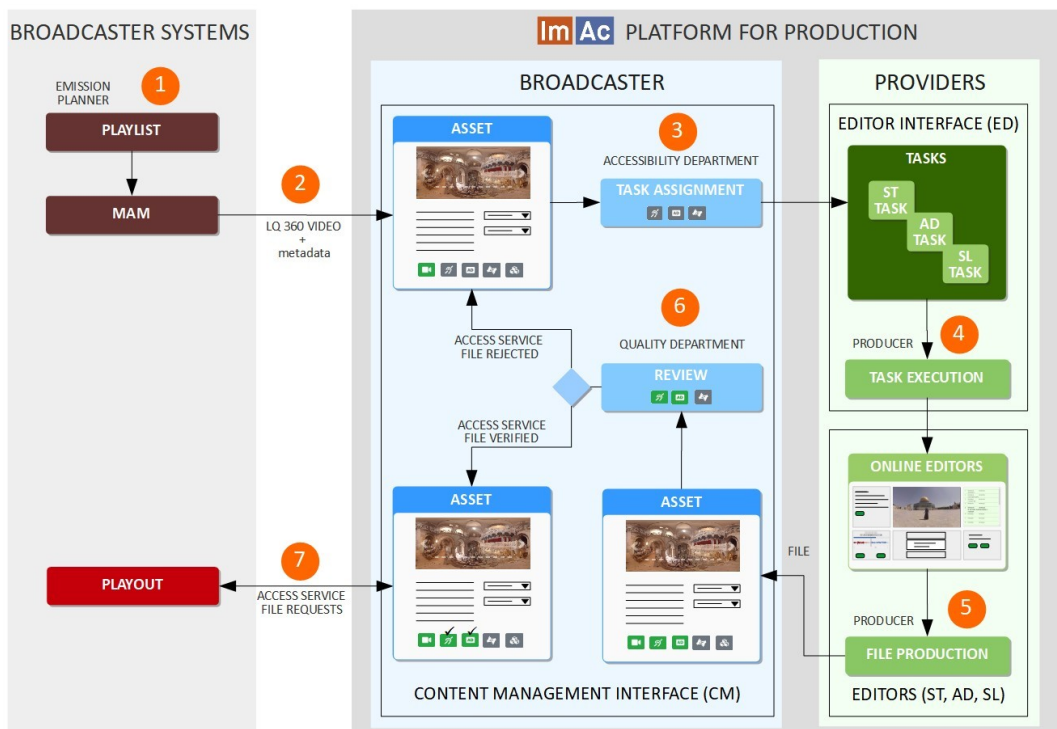


Illustration 5: Steps for the production of content for 360 media access services

3. How to

3.1. INTRODUCTION

This chapter describes the ImAc ACM software installation process and the user access to ACM. The source code is not publicly available and the server is protected via username and password, however the access to the code is open to EC reviewers.

3.2. HOW TO INSTALL THE ACM CODE

- **System requirements**

OS: Linux Debian 9

Processor: 2-core

RAM: 8 GB

Disk space: 2 TB (at the moment of document submission)

- **Access**

In order to install the ImAc ACM, the user needs to access the server via SSH (console)

- **Server:** ****
- **User & pass:** as configured by system administrator

- **Installation**

In order to start packages installation, it is required to install a Linux Debian 9 OS. A “Root” user with full access is essential for installing and setting procedure which comes next in this chapter.

- **Packages**

The system administrator has to install the following Packages in order to have all the tools and programs to run it:

```
apt-get update

apt-get install vim screen rsync ntp less man net-tools apache2 php php7.0-
mysql php7.0-curl php7.0-gettext php7.0-mbstring php7.0-xml openssl mysql-
server

php7.0-odbc curl apt-transport-https php7.0-sybase freetds-common libsybdb5
exim4
```

These commands will install the main tools for the system:

- Apache Web Server

- MySQL Database (MariaDB)
- PHP 7.0 engine
- System Tools

Each package has to be configured accordingly.

Note: the user can use “vim” editor to edit each file.

- **Apache**

Configure the project, its locations and parameters:

```
vim /etc/apache2/sites-available/acm_deliverable.conf
```

```
Alias /acm_deliverable/ /var/www/content_manager_deliverable/html/

<Location /acm_deliverable/>
    order deny,allow
    deny from all
    allow from 172.19.192.0/255.255.255.0
    allow from 127.0.0.1/255.255.255.0
    allow from all
    Options Indexes FollowSymLinks MultiViews
    php_flag magic_quotes_gpc Off
    php_flag short_open_tag Off
    php_flag register_globals Off
    php_value upload_max_filesize 5G
    php_value post_max_size 5G
    php_value memory_limit 2G
    php_value max_execution_time 600
    php_value max_input_time 60
    php_value max_file_uploads 50
    AddDefaultCharset UTF-8
    php_value session.gc_maxlifetime 14400
    ErrorDocument 404 /acm_deliverable/404_not_found.php
</Location>
```

```
vim /etc/apache2/sites-available/editor_deliverable.conf
```

```
Alias /editor_deliverable/ /var/www/content_manager_deliverable/html/ed/

<Location /editor_deliverable/>
    order deny,allow
    deny from all
```

```

allow from 172.19.192.0/255.255.255.0
allow from 127.0.0.1/255.255.255.0
allow from all
Options Indexes FollowSymLinks MultiViews
php_flag magic_quotes_gpc Off
php_flag short_open_tag Off
php_flag register_globals Off
php_value upload_max_filesize 5G
php_value post_max_size 5G
php_value memory_limit 2G
php_value max_execution_time 600
php_value max_input_time 60
php_value max_file_uploads 50
AddDefaultCharset UTF-8
php_value session.gc_maxlifetime 14400
ErrorDocument 404 /editor_deliverable/404_not_found.php
</Location>

```

At this point Apache Web Server needs to be reloaded in order to apply this configuration.

```

a2ensite acm
a2ensite editor
service apache2 reload

```

- **PHP**

The following modification disables some logs from the Apache server log trace:

```
vim /etc/php/7.0/apache2/php.ini
```

```
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT & ~E_NOTICE
```

- **CODE**

A zipped file will be provided for the installation:

```
content_manager_r20.tgz
```

Navigate to the web server code folder:

```

cd /var/www/
sudo mkdir content_manager_deliverable
sudo chown www-data:www-data content_manager_deliverable
sudo chmod 775 content_manager_deliverable
cd content_manager_deliverable

```

Unzip the code file:

```
tar xvfz content_manager_rXX.tgz
```

Configure paths and database parameters in the following file:

```
vim html/includes.inc.php
```

```
<?php
require_once("/var/www/content_manager_deliverable/includes/connection.inc.
php");?>
```

Configure paths and database parameters in the following file:

```
vim includes/config-local.inc.php
```

```
//PATHS
define("PATH_ROOT", "/var/www/content_manager_deliverable");
//ROOT
define("ROOT_PAGES", "/acm_deliverable");
define("ROOT_PAGES_ED", "/editor_deliverable");
//BBDD
$bbdd_usuario='imac';
$bbdd_pwd='****';
$bbdd_servidor='localhost';
$bbdd_bbdd='content_manager_deliverable';
$bbdd_driver='mysqli';
```

- **MYSQL**

A MariaDB 10.1.26 Mysql distribution is used, and the user “imac” is required to import the database.

```
mysql -u root -p
```

```
GRANT ALL PRIVILEGES ON *.* To 'imac'@'%' IDENTIFIED BY 'PASSWORD';
FLUSH PRIVILEGES;
exit;
```

Note: 'PASSWORD' to be changed to the correct value. Ask the administrator.

```
cd /var/www/content_manager/bbdd
mysql -u imac -p content_manager < content_manager_r20.sql
```

Type the password in order to import.

- **Crontab**

This file configures the Linux Task Manager to execute periodically some scripts:

```
vim /etc/crontab
```

```
#CONTENT_MANAGER
* * * * * root /var/www/content_manager/scripts/generate_transcoding.php
0 0 * * * root /var/www/content_manager/scripts/clean_transcodings.php
```

3.3. How to access the ACM interfaces

Once everything is installed and set, the user should be able to access the web interfaces normally. To access the current installed version of the ACM:

1. Open your preferred browser (Chrome and Firefox latest versions).
2. Introduce the URL provided by ACM administrator:
<https://imac.gpac-licensing.com/acm/>
3. A login page will appear as shown in illustration 6:
4. Enter username and password provided by ACM administrators:

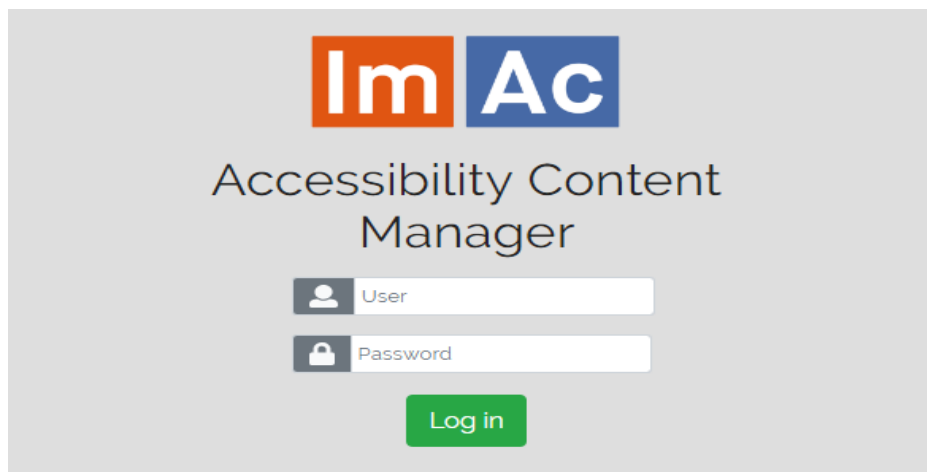


Illustration 6: ACM login page

4. USER MANUAL

In this section, a detailed user manual of the ACM (SM+CM+ED) is introduced.

It is also required to mention that a notable volume of documentation has been created aimed to different applications:

1. D3.2 as a project deliverable and primary resource for ImAc ACM.
2. Demo video demonstrating ACM functionality.
3. ACM short user guide aimed to first-time users (to be used in pilots, test, demos, etc.)
4. ACM factsheet aimed to websites, symposiums, workshops, conferences, advertisements, etc.

There are three sections to be explained in the upcoming three subchapters (4.1 to 4.3). Chapter starts with the SM (System Management interface), continues with the CM (Content Management interface) and finally ED (Editor interface) is the last one to be presented.

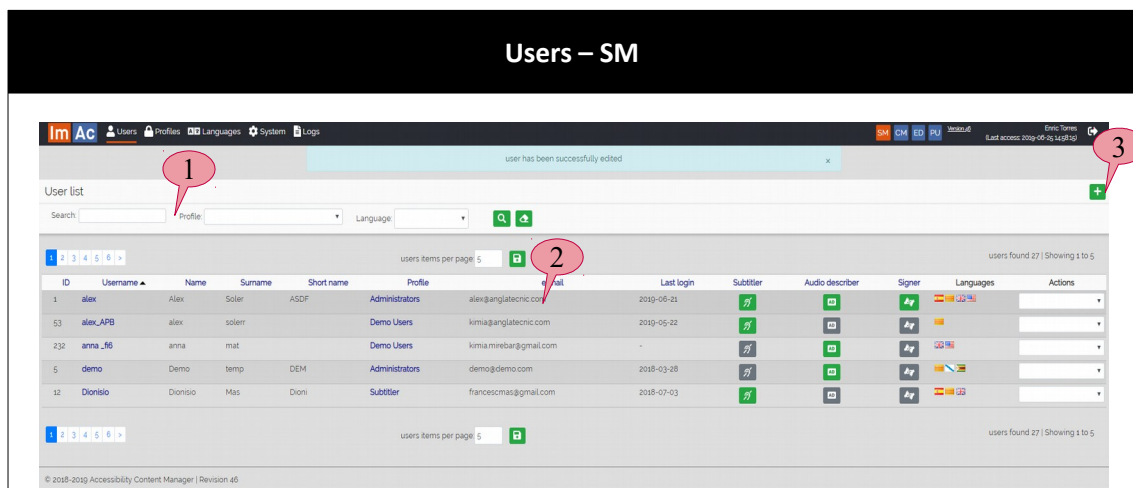
4.1. SYSTEM MANAGEMENT INTERFACE

System Management is defined for platform administration, and allows administrators and broadcaster managers with system administrative duties to control and manage the settings and permissions of professional users working in the ACM.

Chapter 4.1 is divided in 6 sub-chapters, the first five sub-chapters go through the different functionalities of the SM, and the final part gives a scenario (use case) as an example for means of clarification.

4.1.1. Users

The functionality of this part is focused on users administration.



1	<p>With this tool, user can search through the existing users based on three different criteria:</p> <ul style="list-style-type: none"> • Free search • Profile type • Language
2	<p>The list of users is available in this part. All of the registered users can be found here with their following information:</p> <ul style="list-style-type: none"> • ID number (unique) • Username (by clicking on it – in the case of having permissions of users edition – the logged in user is redirected to a form page for editing the settings belonging to clicked-on user – see illustration 6) • Full name and short name • Profile type – email address • Last login date and time • The access service production tasks they are able to cover (ST, AD, SL) • The languages they can support • Actions available (edition, deletion, block)
3	<p>By clicking on this positive sign button, a new user can be added.</p>

Table 1: Users list

At the moment of creating/editing a user, the form shown in illustration 7 appears. This form needs to be completed in order to create/edit users. The required information in the form is typical and basic (personal information, type of access service provision, languages).

It is noteworthy that all of the users including broadcasters and access service producers are created here. It is significant that at the moment of creating a producer such as subtitler, audio describer or sign interpreter, the corresponding options and also the languages they aim to work with are appropriately marked in the form in illustration 7.

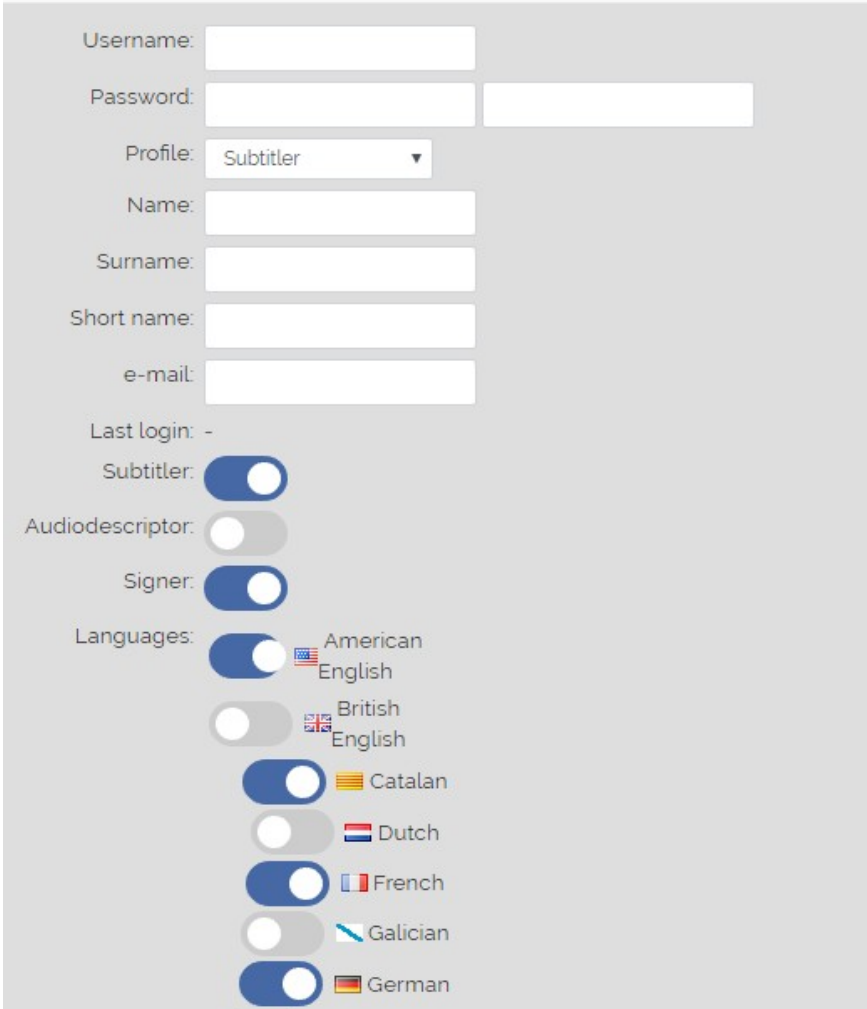
The form is a user creation or editing interface. It features several input fields: 'Username', 'Password' (with a confirmation field), 'Profile' (a dropdown menu currently showing 'Subtitler'), 'Name', 'Surname', 'Short name', and 'e-mail'. Below these fields, there is a 'Last login' status showing '-'. A series of toggle switches are provided for selecting roles: 'Subtitler' (checked), 'Audiodescriptor' (unchecked), and 'Signer' (checked). Under the 'Languages' section, there are more toggle switches for various languages, each accompanied by a small flag icon: 'American English' (checked), 'British English' (unchecked), 'Catalan' (checked), 'Dutch' (unchecked), 'French' (checked), 'Galician' (unchecked), and 'German' (checked).

Illustration 7: User edition/creation form

4.1.2. Profiles

It is important for any system of this scale to define different profile types, because the level of permissions is different for each stakeholder and user defined earlier in this document, so there is a need to define different levels of access and permissions for each one. Profile types refer to different types of permission given to a user based on their access criteria.

To the moment, there are 8 different types of profile on the platform. Table 2, sheds a light on this issue better:

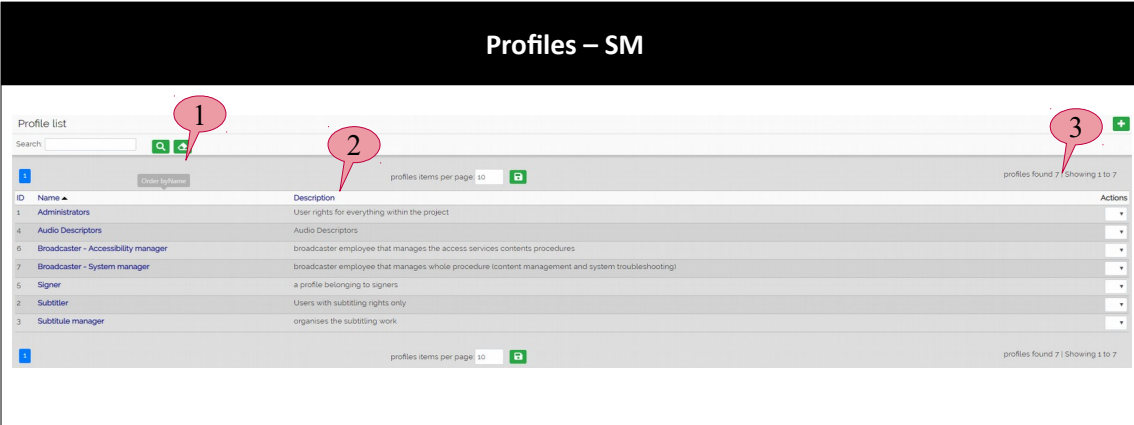
Profiles – SM		
		
1	With this tool, user can search through the existing profiles.	
2	A list of profiles with their description.	
	Administrator	User rights for everything within the project
	Broadcaster – Accessibility manager	broadcaster employee that manages the access service procedures
	Broadcaster – System manager	broadcaster employee that manages system administration and troubleshooting
	Audio descriptor	Audio description producers
	Subtitler	Subtitle producers
	Signer	Sign language producers
	Subtitle manager	organises the subtitling work
	Demo users	Profiles aimed to new users for testing purposes
3	By clicking on this sign button, the user can add a new profile	

Table 2: Profiles

By clicking on the positive sign button on the top right of the page, the user is redirected to a separate page consisting of a form for profile creation. In this environment, the technician can choose different capabilities and access permissions for the profile (reading, editing, deleting, downloading)

Table 3 shows such form:

Profile form– SM

New profile

1

Name:

2

Description:

Superuser:

Folders:

Read

Create

Edit

Remove

Choose a folder

System Manager user rights

Read

Create

Edit

Remove

Users

Profiles

Languages

System

Logs

Content Manager user rights

Read

Create

Edit

Remove

Download

Assets

Trash

Reports

Editor Tools user rights

Read

Edit

Subtitle

Audio Description

Signed Language

Videos

PrePack user rights

Read

Edit

PrePacks

1	General information of the profile (name and description).
2	Access permission to folders of CM, with possibility to choose among existing folders on the ACM.
3	Various access permission to various parts of the ACM.

Table 3: Profile creation form

4.1.3. Languages

In this sub-chapter, new languages for the access service contents can be created or existing ones can be edited.

D3.2 – Accessibility Content Manager

29

Version 2.4 09.07.2019

Languages – SM	
1	With this tool, the user can search through the existing languages
2	List of existing languages with their locales, by clicking on the language name, the user is redirected to the edition page
3	By clicking on the positive sign, a language creation form is opened.

Table 4: Languages list

By clicking on the positive sign, a creation form appears. Illustration 8 shows the case where the user wishes to edit an existing language. In the case of creation the inputs are empty and no ID is specified yet.

Illustration 8: Language creation/edition form

4.1.4. System

“System” section corresponds to administrative actions done by system administrators or broadcaster system managers. It is divided into three parts which will be discussed respectively in tables 5, 6, 7:

Variables – System – SM					
System options Variables Engines Notifications	Variable list				
	Search <input type="text"/>				
	GENERAL				
	Title	Name	Value	Description	Editable Actions
	Id of template for notify editors	ID_NOTIFICATION_EDITOR	4	Notifications Template for available file to subtitle, audiodescriptor or signer	1
	Interval to generate keyframes	KEY_FRAME_INTERVAL	0.1	This variable is used to generate keyframes taken from uploaded file of the asset	1
	PUBLISH				
	Title	Name	Value	Description	Editable Actions
	Audio file name convention template	AUDIO_FILE_TEMPLATE	outputFormat/audiosFormat/audi-model-iph	Name template for all the audio files that will be requested to Cloud Renderer	1
	Default video resolutions to transcode in Packager	DEFAULT_VIDEO_RESOLUTIONS	1080 720 540 480	Include, separated by commas, the different resolutions that will be de default parameter to transcode in Packager	1
	SYSTEM				
	Title	Name	Value	Description	Editable Actions
	Days to clean finished transcodifications	DAYS_CLEAN_TRANSCODINGS	3	Set the number of days to clean transcodifications older than this value in days	1
	Audio Renderer Receiver URL	URL_WS_CLOUD_RENDERER	https://trngkylt7l.execute-api.eu-central-1.amazonaws.com/DEV	url to audio renderer receiver starts rendering process	1

In this section some system variables can be set or modified, for instance during the installation and customization of ACM.

Variables are divided into three categories:

- **GENERAL:** corresponds to general interface variables.
- **PUBLISH:** corresponds to variables related to web service in general.
- **SYSTEMS:** corresponds to system variables.

1	With this tool, the user can search through the existing variables.
2	List of existing variables with their characteristics, by clicking on the variable's name, the user is redirected to an edition page
3	By clicking on the positive sign, the user is able to create a new variable

Table 5: Variables of system

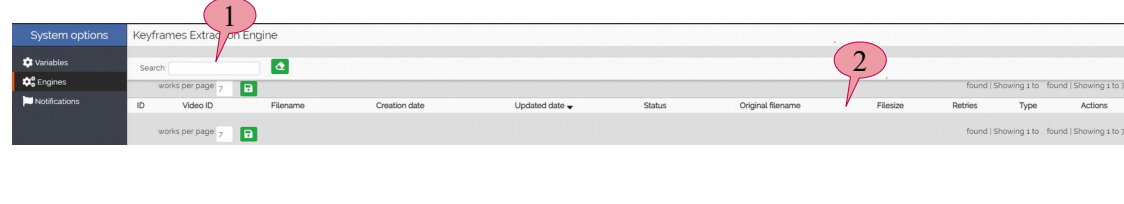
Engines – System – SM	
 <p>The screenshot shows the 'Engines' management interface. On the left is a sidebar with 'System options', 'Variables', 'Engines', and 'Notifications'. The main area is titled 'Keyframes Extraction Engine'. It features a search bar (callout 1) and a table of existing engines (callout 2). The table has columns for ID, Video ID, Filename, Creation date, Updated date, Status, Original filename, Filesize, Retries, Type, and Actions. The table is currently empty, showing only pagination controls.</p>	
<p>This section shows the working queues of the ACM engines (software without user interaction). Only one engine is developed for the ImAc project that extracts the keyframes once a video has been uploaded.</p>	
1	With this tool, the user can search through the existing engines.
2	List of existing engines with their characteristics.

Table 6: Engines of the system

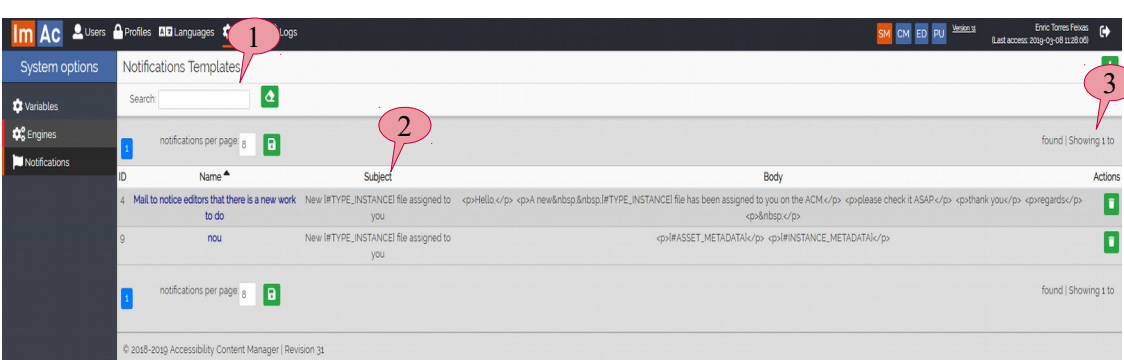
Notifications – System – SM	
 <p>The screenshot shows the 'Notifications Templates' management interface. On the left is a sidebar with 'System options', 'Variables', 'Engines', and 'Notifications'. The main area is titled 'Notifications Templates'. It features a search bar (callout 1) and a table of existing templates (callout 2). The table has columns for ID, Name, Subject, Body, and Actions. The table contains two rows of templates. A red callout circle 3 points to a green plus icon in the Actions column of the first row.</p>	
<p>This section lets the user configure notification and mailing templates using a special form. The form is shown in illustration 9.</p>	
1	With this tool, the user can search through the existing notification templates.
2	List of existing templates with their name, subject and body, by clicking on the name, the user is redirected to an edition page.
3	By clicking on the positive sign, a creation form is opened.

Table 7: Notification templates list

4.1.6. Scenario

In this sub-chapter, the following scenario is elaborated to clarify SM functionality. This scenario is an example on how to create a new profile for subtitle managers, setting the access permissions of this profile conveniently and then creating a new user based on this profile.

1. The user enters the form (table 3) and creates a new profile named “subtitle manager”.
2. This profile aims to people who want to manage the subtitling workflow.
3. Illustration 10 shows the specifications of this profile, note the permissions this profile has.
4. Next move, the user goes to User creation form to create a new user, now it is possible to assign the profile that has just been created to the new user, illustration 7 shows how it is done.

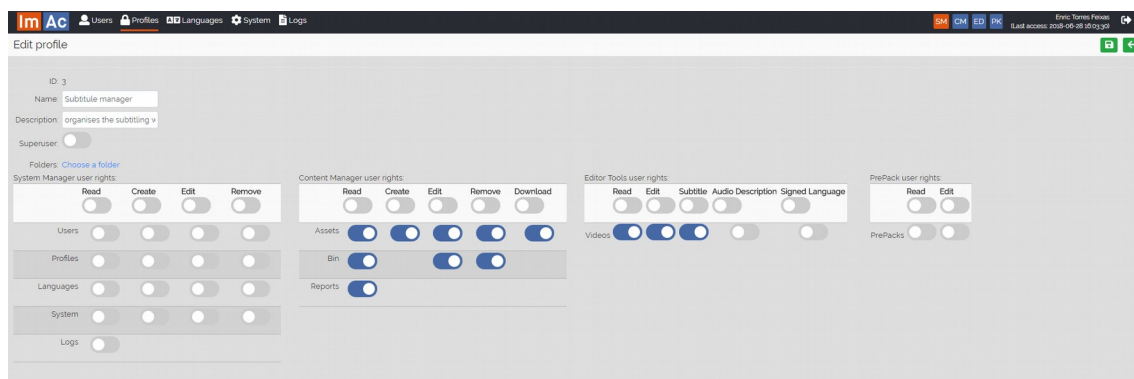


Illustration 10: Creating the “Subtitle manager” profile

4.2. CONTENT MANAGEMENT INTERFACE

Content Management interface is the place where all the access service contents can be found, managed and controlled. It is divided into the following subsections that are explained in the next three sub-chapter:

- Assets – management of assets and their access service contents and production tasks.
- Bin – a temporary bin to keep deleted assets
- Reports – stores statistical data and information about the CM

4.2.1. Assets

Due to the importance of CM and assets, this interface is going to be explained section by section.

The contents are organised in Assets. Each asset has all the contents that are related to a 360 programme and required for both producing the access service files and managing the results of the production. So the asset includes metadata and file instances such as the LQ video file and all the access service files for the programme.

This section allows the user to manage and catalogue the assets and also their properties individually.

There are two ways to catalogue assets:

- By metadata, so to find an asset the user needs to use the search bar.
- By folder, so assets are catalogued in a hierarchical order.

4.2.1.1. Asset folders

Folders of CM have the objective of creating a more flexible and organised interface. They can contain assets in themselves and are editable. By default, the user enters the page while being in folder “Root” and in the hierarchical view. There is a small folder icon on top left which lets the user switch between metadata view and hierarchical view (table 9 shows a full page screen while being in hierarchical view and in one of sub-folders).

To change between the two mentioned ways, the user needs to click on the folder icon on the top left of the page (see folder icon in Illustration 11). When the icon shows a closed folder the metadata way is selected, when it is an open folder the hierarchical way is selected.

It is possible to create/delete/rename folders and navigate through them. Navigating by folder comes in handy when the work volume of broadcaster is high and the videos are numerous, so the user can organise the work based on criteria they look for (based on schedule of emission, importance, channels, etc.).

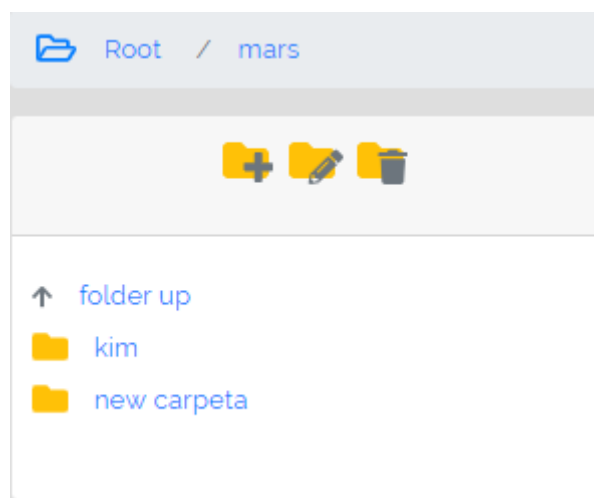
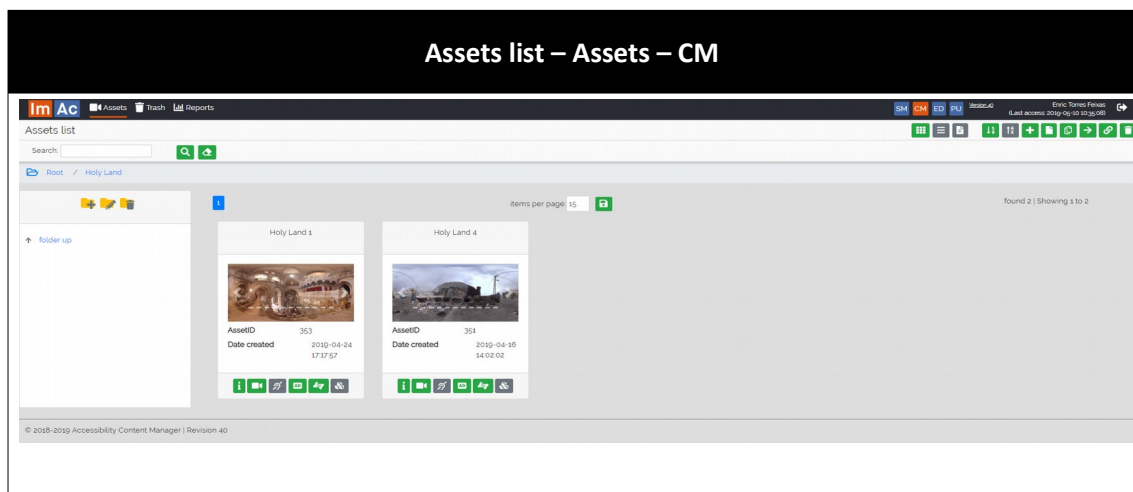


Illustration 11: Folders hierarchical order

4.2.1.2. Assets main page

The assets main page is the one that matters the most when dealing with broadcasters responsibilities on contents management.



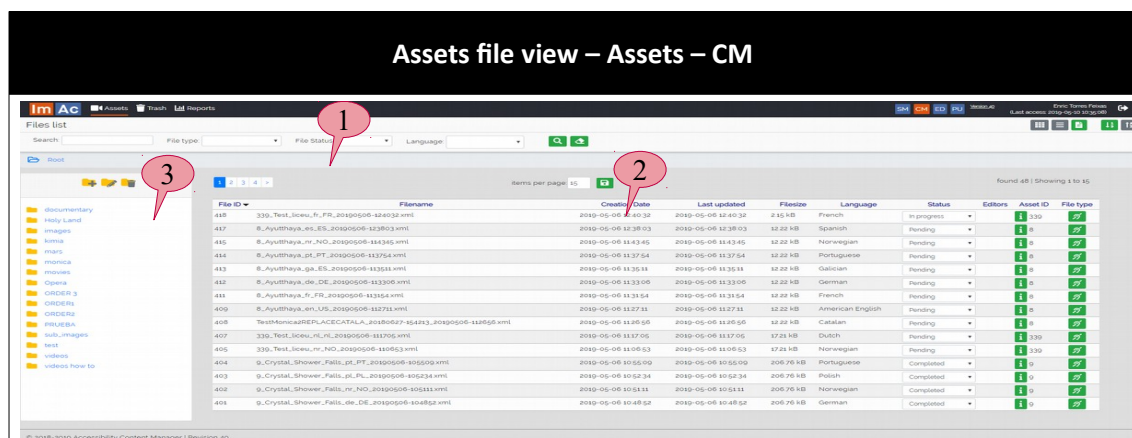
The page contains a list of available assets with the possibility to search through them. Following buttons are to be interacted with by the user on the top right of page:

	Sorts assets by numerical order
	Sorts assets by alphabetical order
	Changes the view of assets by grid view or list view
	To be used when the user wants to add a new asset
	When the user wants to import assets from FTP where the high resolution video resides (6.1 explains about the procedure of web services and video import)
	The user selects one or various asset (for multiple selection use the Ctrl key) and with this button chooses the new name and destination among folders for it then copies it to a new path
	The user selects one or various assets (for multiple selection use the Ctrl key) and with this button chooses the new destination among folders for it then moves it to a new path
	The user selects one or various assets (for multiple selection use the Ctrl key) and with this button create links to them
	The user selects one or various assets (for multiple selection use the Ctrl key) and with this button erases them

Table 9: Assets list

4.2.1.3. File view

In addition to the points mentioned in 4.2.1.2, CM gives the user the possibility to navigate through assets and their instances by three different modes. As seen in table 9, the grid and list view are clear. However in addition to these, it is possible to switch to a view that shows not assets, but all the access service files available on ACM (ST, AD, SL). Table 10 displays more information in this sense:



This view gives the user all the available access service files on ACM.



This is the button that gives the user this view on the top right.

1

On the top of the page, there are search boxes with following criteria:

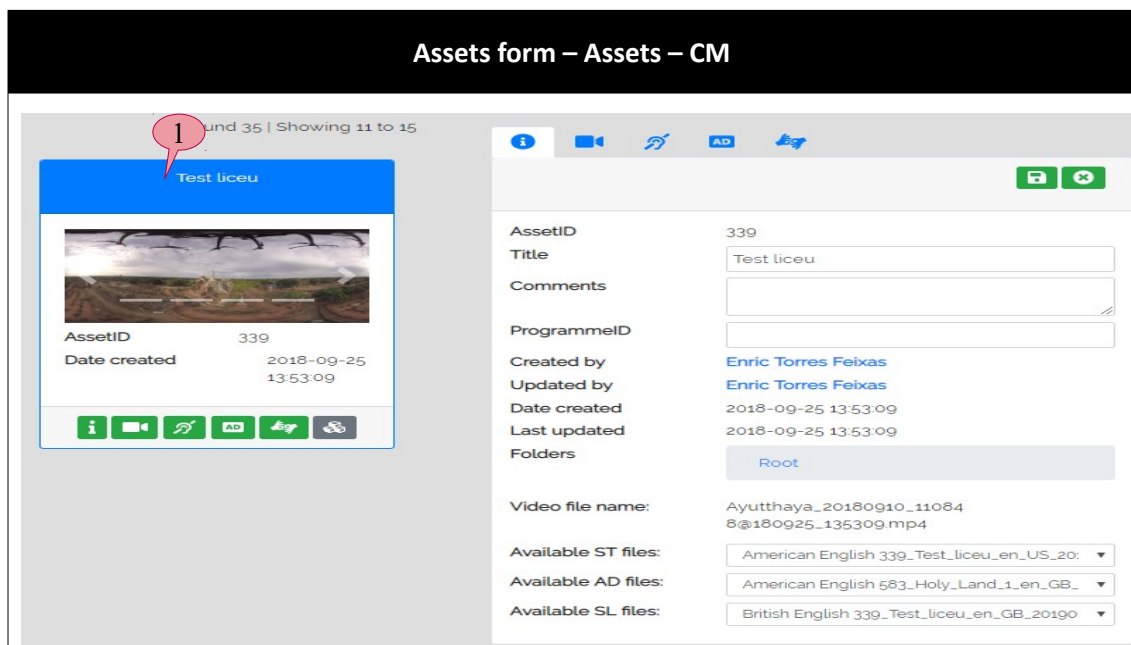
1. Free search
2. File type which can be either:
 - I. Subtitles
 - II. Audio description
 - III. Sign language
3. File status which can be either:
 - I. Pending: at first all instances are at this status
 - II. In progress: The broadcaster accessibility department assigns the work to a producer and once the producer has started the work the status changes to "In progress"
 - III. Completed: when the producer finishes, they can change the status to completed via other ImAc interface (ED)

Assets file view – Assets – CM	
	<p>IV. Rejected: after being completed broadcaster's quality assurance department needs to either reject or verify the final result</p> <p>V. Verified: see above</p> <p>4. Language</p>
2	<p>List of files with following information and the possibility to order by them:</p> <ol style="list-style-type: none"> 1. ID 2. File name 3. Creation date 4. Last updated 5. File size 6. Language 7. Status 8. Editors 9. Asset ID 10. Type
3	It is possible to navigate through files by folder hierarchy as well

Table 10: File view

4.2.1.4. Asset form

As demonstrated in 4.2.1.2 the assets list contains all of the available assets with their properties. By double clicking on an asset or clicking one of the buttons, the asset form opens. Table 11 shows what happens when an asset form opens:



The page contains a list of available assets with the possibility to search through them. When double clicking on an asset or by clicking on any bottom green buttons which reflect the type of access service file, the right form opens and the first view (general) or other type (depending on the green button you have pressed) is seen.






 1	The asset card is shown containing: a video preview, basic information and the availability of each access service (when the icon is green it means the access service content is available right now and when it is grey it shows that it does not exist and it needs to be created if needed)
  	General information of the asset in a single view: <ol style="list-style-type: none"> 1. Titles and comments 2. User who has created with related dates 3. the directory of the asset in the folders 4. Available video and access service files
	Video properties (see table 12)
	Subtitles properties (see table 13)
	Audio description properties (see table 13)
	Sign language properties (see table 13)

Table 11: Asset form

Table 12 demonstrates what video properties of an asset contains and how the user can interact with it:

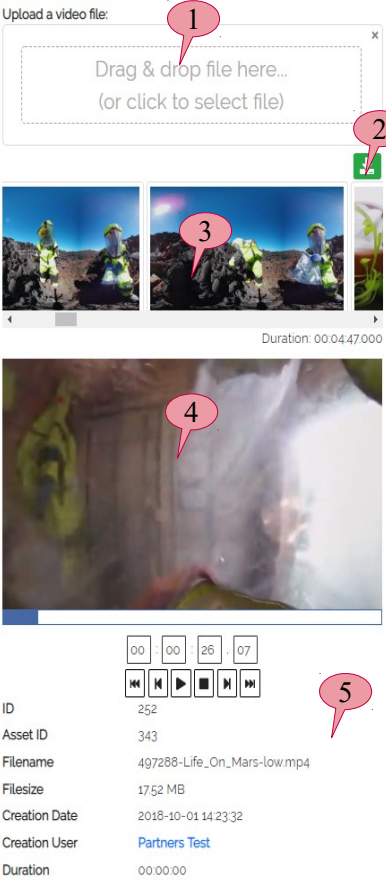
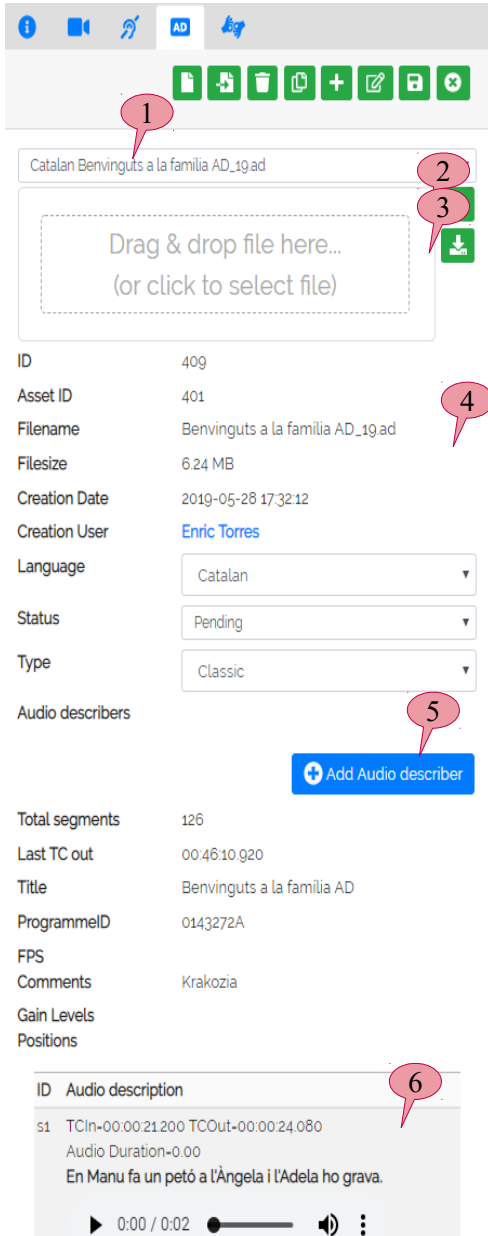






Assets form (video) – Assets – CM	
 <p>The screenshot shows the 'Assets form (video) – Assets – CM' interface. It includes an upload area (1), a download/upload button (2), a video preview in frames (3), a video player (4), and a metadata section (5). The metadata section lists: ID (252), Asset ID (343), Filename (497288-Life_On_Mars-low.mp4), Filesize (17.52 MB), Creation Date (2018-10-01 14:23:32), Creation User (Partners Test), and Duration (00:00:00).</p>	The left image shows the asset form – video properties:
	1 A place to upload a new video file from PC (the LQ video)
	2 A button to download and upload the existing/new video (in the case of existence or a new one all over again)
	3 A preview of the video shown in frames from it
	4 Video player but in a small scale along side with video control buttons (play, pause, stop, etc.)
	5 General information of the video: <ul style="list-style-type: none">• ID (unique)• Asset ID (unique)• File name• File size• Creation date• Creator• Video duration

Table 12: Asset form – video properties

The remaining properties (ST, AD and SL) are quite similar. In table 13 the properties are shown. AD is chosen in the table and ST and SL has been eliminated for convenience, as AD properties contain all of the information and it is the one with more buttons and options:

Asset form (AD, SL, ST) – Assets – CM

		The left image shows the asset form – AD properties:	
1	A place to choose the preferred AD file (instance), because an asset can contain multiple AD files		
2	A button to upload an AD file with various formats (more information presented in previous chapter)		
3	A button to download the existing file (the one displayed in the above select), with option to be exported in various formats (more information presented in previous chapter)		
4	General information of the file: <ul style="list-style-type: none">• ID (unique)• Asset ID (unique)• File name• File size• Creation date and user• Language• Status of work• Type of AD (static, dynamic, classic)		
5	A button to a dialogue to assign the production task to an audio descriptor		
6	Preview of the AD file		
	Deletes the file		

Asset form (AD, SL, ST) – Assets – CM		
		Copies the file with existing TC and segments with possibility to choose AD type and language
		Adds an empty AD file to be produced (possibility to choose AD type and language)
		Opens the corresponding editor with the LQ video and file.
		Saves the changes
		This button which only exists in AD form, lets the user import mp3, wav and m4a audio tracks. These tracks are converted to AD files automatically to create segments (at the moment of importing, choosing type and language is possible). What the action does is having an audio track consisting of segments with silences between each two. The silences are detected to determine time codes for conversion of this track to an .ad file



Asset form (AD, SL, ST) – Assets – CM	
	 This button only exists in AD and SL form and it mainly serves in following conditions: <ul style="list-style-type: none"> When in AD, it lets the user choose from an existing subtitle files in the same asset to be imported as an AD file with the same timings and script, with the audio segments to be recorded by the responsible person when creating an AD file for voice over. When in SL, it lets the user choose from existing subtitle files in the same asset to be imported as an SL file. This import helps the user to have the same timing and script from ST in order to transform the format to a sing language file and it facilitates the procedure.
	 Closes the form

Table 13: Asset AD properties

4.2.2. Bin

Bin acts like a recycle bin for the system. Contents which have been moved to bin cannot be edited or changed but it can be restored back into Assets or can be deleted forever.

4.2.3. Reports

In this subsection, the user can checkout the statistical data and information showed in an interactive manner that can be understood by all of the workers. To this date, 3 charts are displayed:

- Downloaded files
- New files
- Finished files

We can search through this data by date. Illustration 12 shows the page clearly:

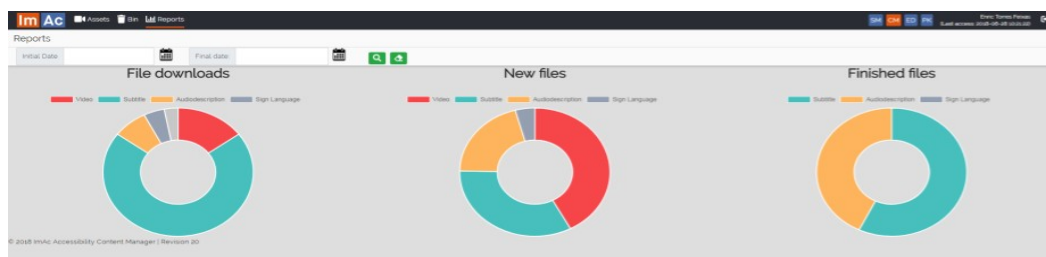


Illustration 12: Reports

4.2.4. User scenario

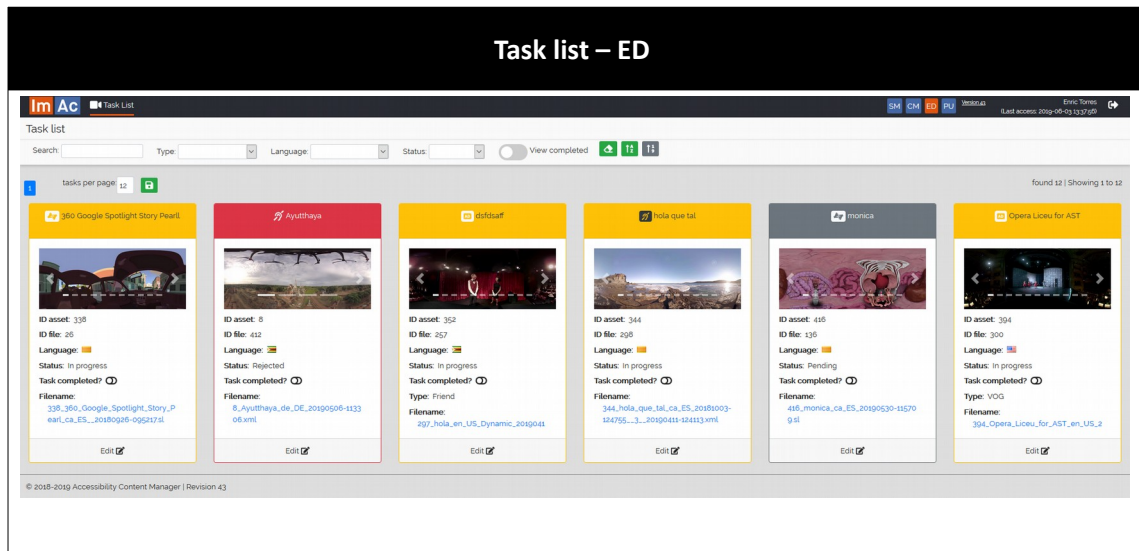
Now that the reader knows every section of ACM, a scenario is described. Imagine creating an asset from scratch:

1. There are two ways to create assets:
 - a) Manually as shown in table 10 with the positive sign. User can create an asset, upload the LQ video and modify its characteristics.
 - b) Automatically by receiving a LQ video by the broadcaster (see 2.2).
2. After being created, on the top right there are options for assets like moving, copying, linking and erasing (see table 11).
3. By entering in the asset form, the user goes to the desired access service section when a new access service content is required, creates a new instance (empty access service file), and edits its characteristics like language and type (see table 13).
4. From the new instance, user assigns the task of producing the access service content (editing the access service file) to one or various producers such as subtitlers, audio describers or signers (depending on the access service type).
5. The specific access service button turns green when new instances have been created (see table 11).
6. This step is executed outside CM. From ED a producer (professional or external service provider) do the task of access service production that has been assigned to them and when the access service production is finished change the task status to “Completed”.
7. Once the production task is set to completed, the access service content is ready to be reviewed and validated by the broadcaster’s quality assurance department. After review they either confirm it and change the status to “Verified” or they send it for a re-do and change the status to “Rejected” and with regards to this, the producer would know they need to work on it again. One good recommendation for broadcasters is to change to “File list view”, order the files by their status to find out which tasks are completed in order to verify them.
8. Once the access service files are validated by the broadcaster, the broadcaster’s play-out system can access them when the 360 programme is published or broadcasted. For this it is necessary that the platform is fully integrated with the broadcaster’s systems.

4.3. EDITOR INTERFACE

4.3.1. Main page

This is the last section to be demonstrated in this deliverable. This is the interface specially destined to access service producers and lets them see their previously assigned tasks, search through them and more importantly gives them direct access to ImAc editors. Table 14 shows more details.



The page contains a list of previously assigned tasks to the producer who is viewing the page.

1	<p>The possibility to search through tasks based on:</p> <ul style="list-style-type: none">• Type (ST, AD, SL)• Language• Status
2	<p>Existing tasks in shape of cards. The above colour of each card reflects the current status of it and also type of it. While below of it characteristics are seen. When the production task is completed, the producer should mark the corresponding option in order to change the status.</p> <p>The most below button (“Edit”) takes the user to the corresponding editor with the video and the file to be produced.</p>

Table 14: Task list via ED

4.3.2. User scenario

Imagine a producer is working on ED, these are typical steps they would take:

1. User signs in the platform.

2. User sees all their production tasks in ED main page and the file status (Pending, In progress or Rejected).
3. User chooses a task to do and is redirected to the corresponding editor.
4. Once the production is finished, user changes the status to “Completed”.
5. User keeps coming back to the platform to see if their work has been “Verified” or “Rejected”. If rejected, they need to work on it again.

5. ACM ACCESS SERVICE FILE FORMATS

ACM grants the possibility for broadcasters to work with the formats defined in the ImAc project for the 360 access service files. However, they have the rights to export the result of their work on ACM to different broadcasting formats or import from other broadcasting formats. Below it is discussed the formats that are supported in this project.

5.1. SUBTITLE IMPORT AND EXPORT

By default, the ACM imports and exports subtitle files using the ImAc format which has been defined in T4.1 and will be described in D4.4. The ImAc format for subtitles is an XML file based on the IMSC1 profile of TTML. However the ACM also allows the user to export the subtitle file to EBU-TT(D) format when clicking the download button available on the user interface. The difference between the two is the number of elements in each one. ImAc format supports and demonstrates more elements such as longitude and latitude.

By clicking on the upload button provided on the CM interface, the user can also import a subtitle file with **EBU-TT(D)** format and upload it to the ACM to work with it accordingly. As mentioned above, some elements in this format are not existent and are left empty for the producer to be filled while editing.

5.2. AUDIO DESCRIPTION IMPORT AND EXPORT

By default, the ACM imports and exports audio description files using ImAc format which has been defined in T4.2 and described in D4.4. But it also allows to export audio description files into other formats and also import files with other formats from disk to the ACM.

For the purpose of importing the ESEF format (.esf) and the Fingertext format (.nar) are supported. To import an ESEF audio description the user chooses a zip file from disk which must include the .esf file and all the audio files, and to import a Fingertext audio description the user chooses a .nar file from disk (the Fingertext .nar file has all the audio files included in it). These file formats do not contain all information and elements required on ImAc access service files, so when chosen, a dialogue opens requiring the user to choose a language and type of AD for the file.

The ACM also allows to export ESEF format (.esf) and the Fingertext format (.nar), so when pressing the download button provided in the CM interface, a dialogue opens to choose either ImAc audio description (.ad), ESEF audio description (.esf) or Fingertext audio description (.nar).

5.3. SIGN LANGUAGE IMPORT AND EXPORT

The ACM imports and exports sign language files in ImAc format which has been defined in T4.3 and described in D4.4.

6. ACM BACKGROUND PROCESSES

As mentioned in chapter 1 and 2, the aim of ACM is management of the access service production and distribution workflow. For the purpose above, the existence of background process and web services is essential in order to communicate between different elements end points.

A full list of existing background processes and a brief definition is presented in Annex II. However, a description of the background processes with its web services is made in this chapter.

6.1. VIDEO TRANSCODING TO LOW QUALITY PROCESS

As mentioned in previous chapter, the ACM works with LQ videos. However, it does accept HQ videos for uploading and via the web service “Video transcoding to low quality” transcodes them into LQ videos. In this sub-chapter the three stages of this background process are explained.

6.1.1. XML Flag File

The ImAc platform requires that the 360° high resolution video is uploaded to the ImAc server. When this high resolution video has been successfully uploaded, the user must upload an XML flag file to trigger the video transcoding process by the ImaCron script (see next chapter) to the LQ video, this procedure is fully described in ImAc D3.3 deliverable^v. When the LQ video (automatically produced by the ACM, user only uploads the HQ version) is available, the ACM automatically imports it and creates a new asset that contains the LQ video. The XML flag file has to be named the same as the video file, ending with “.xml” extension instead of the video extension and can be created automatically from ACM. Illustration 13 demonstrates the window for this purpose. The factors to be chosen are explained in chapter 6.1.1. Due to the fact that creating XML file might be difficult and time consuming, an interface as illustration 13 is created in ACM so the user would enter the metadata via this.

Among all the inputs of illustration 13, the variables “Video size” and “Audio format” (Stereo, binaural or FoA) are important for the right transcoding of the HQ video into the LQ one. Other variables depend on the user choice purely (ACM folder, title, etc.).

Import asset from FTP

Title	Asset title (mandatory)		
Language	American English		▼
Programme ID	Internal Programme ID (optional)		
ACM folder	Choose the ACM folder (optional)		
Video file name			
	160901_RBB_Abendschau_4K-25mbs_injected_20180618-092647.mp4		▼
Video file size	In bytes (mandatory)		
Audio file name			
	160901_RBB_Abendschau_4K-25mbs_injected_20180618-092647.mp4		▼
Audio channels			
Audio language	American English		▼
Audio format	Stereo		▼

[Import](#) [Close](#)

Illustration 13: Import asset from ImAc FTP

The XML flag file format is as follows. It is noteworthy that this is only an example to show how the XML file is produced and what parameters are crucial. As mentioned above these parameters come directly from the metadata provided by the user in illustration 13 form.

```
<?xml version="1.0" encoding="utf-8"?>
<content acm_virtual_folder="IRT_Test" lang="de_DE" programmeID="497313"
title="Foxfinder Alt">
  <inputs>
    <video>
      <file>videoname.mp4</file>
      <size>4830548798</size>
      <!-- Size of file in Bytes (Mandatory) -->
    </video>
    <audio>
```

```

<file>videoname.mp4</file>
<size>4830548798</size>
<channels>4</channels>
<lang>de_DE</lang>
<format>ambisonics</format>

</audio>
</inputs>
<output>
  <targetFolder>./outputs/</targetFolder>
</output>
</content>

```

6.1.2. ImaCron script

A Linux cron job will trigger the ImaCron script every five minutes. A cron is a time-based job scheduler that allows the system to perform some user scripts. ImaCron checks if there is a HQ video with an associated XML flag in the FTP server “input” folder (/home/imaac/ftp/input), checks its variables, starts the transcoding of the HQ video into a capable HTML5 LQ video and notifies the ACM when done. The capable HTML5 LQ video is needed so that the Web Editors are able to playback the video in a smoothly way.

The below command line corresponds to the “ffmpeg” application that does the transcoding from the HQ video into the LQ video. It also states the video and audio codecs, video resolution as well as other parameters:

- `ffmpeg -i % $file -c:a aac -vf scale=-2:720 -c:v libx264 -bf 0 -crf 22 low_quality.mp4`

The imaCron writes all the performed tasks as well as possible errors produced when parsing the XML file or the transcoding of the video into the imaCron logs. These logs are text files located in the “imacron_logs” folder (/home/imaac/ftp/imacron_logs).

After the transcoding is finished, imaCron will move the video files (HQ and LQ) and the XML flag file into the “output” folder (/home/imaac/ftp/output) if the process has been successful, or into the “error” folder (/home/imaac/ftp/error) otherwise. I will then call the ACM web service “insertLow” to notify that the video has been transcoded with the necessary metadata to automatically create the asset in ACM.

6.1.3. InsertLow

InsertLow is an ACM web service that imports the LQ video file into ACM and maps the high resolution video file location into the database for further uses such as with Cloud Renderer or Packager (ImAc tools described in detail in D4.2 and D3.3 respectively) as it will be explained later in this document.

The following JSON (an open-standard file format for data transmission) is an example of the request sent to the InsertLow:

```
{
  "pathXML": "/home/imac/ftp_test/output/assetName.xml",
  "pathVideoLow": "/home/imac/ftp_test/output/assetName-low.mp4",
  "pathVideoHigh": "/home/imac/ftp_test/output/assetName.mp4"
}
```

And the following is an example of the JSON response from InsertLow to confirm that the task has been executed correctly (with no errors):

```
{
  "result": "1",
  "msg": "Asset 350 with programme ID 497313 successfully created for low
resolution video /home/imac/ftp_test/output/assetName-low.mp4"
}
```

The possible values for “result” are:

- “1”: Asset successfully created
- “0”: Error while creating the asset

6.2. AUDIO RENDERING PROCESS

Before sending the AD file to the player, the file requires to be mixed with the main programme audio. This method has been chosen as the player is not designed to do the mixing of the AD during the playback in order to minimize processing load at the player side. It is then required to have an engine which prepares this mixing in advance. This engine is called the Cloud Renderer (another ImAc tool) which is explained in detail in D4.2.

The Cloud Renderer has a web service in order to receive notifications from ACM for incoming audio rendering jobs. It receives the location of the HQ video file to extract the main audio, the location of the AD file and the different parameters of the required job (AD type, output format, language, audio file format, asset ID).

For this web service to be executed, first a set of requirements need to be fulfilled:

- AD file with the “Verified” status for at least one language and one presentation mode (the list of presentation modes and a brief description can be found in Annex I).
- High Resolution video mapped into the database.
- If done manually the user needs to have “publish” permission.

In this sub-chapter the three stages of this process are explained.

6.2.1. Audio rendering request

When the requirements are met ACM will launch a notification to the Cloud Renderer to start the audio rendering (see chapter 6.2.2). This process can start automatically when an AD file changes its status to “verified”. The different possible status of access service tasks are

discussed in previous chapter.

The user can also start the Cloud Renderer notification manually by opening the “Asset publication” window from the Asset in CM and clicking the “Prepare” button if the requirements for audio rendering are fulfilled.

ACM waits for the Cloud Renderer report (see chapter 6.2.3).

6.2.2. Notification to the Cloud Renderer

The following call to the Cloud Renderer web service with some specific parameters is made to start the audio rendering process:

- Method: “PUT”
- Data type: JSON
- Parameters:
 - **assetId**: ID of the ACM Asset
 - **callbackUrl**: URL for the Cloud Renderer to call back the audioRenderingReport.
 - **audioFileTemplate**: text format options for the filename.
 - **audioDescriptionType**: “VOG”, “Friend” or “Dynamic”
 - **outputFormats**: “stereo”, “binaural” or “FOA”. Multiple choice.
 - **audioFileFormats**: “aac”

The JSON structure of the call is as follows, it is noteworthy to mention that below code is purely an example extracted from ACM with aims to demonstrate the parameters and their possible value:

```
{
  "assetId": "357",
  "callbackUrl": "https://imacplatformserver.com/acm_test/ws/insertOBA.php",
  "uploadUrl": "ftp://imacplatformserver.com/ftp_test/output/cloud_renderer/",
  "admFile": "https://imacplatformserver.com/acm_test/instances/audio/",
  "videoURL":
    "https://imacplatformserver.com/acm_test/instances/video_hq/foxfinder2.mp4",
  "audioFileTemplate": "{outputFormat}/{audioFileFormat}/{uid}-{mode}-{phi}.
    {extension}",
  "audioDescriptionFile":
    "https://imacplatformserver.com/acm_test/instances/audiodescription/
    357_Foxfinder_2_de_VOG_20181211-172302.ad",
  "audioDescriptionType": [
    "Dynamic"
  ],
  "audioDescriptionLanguage": "ca_ES",
  "outputFormats": [
    "stereo",
    "binaural",
    "FOA"
  ]
}
```

```

],
"audioFileFormats": [
  "aac"
]
}

```

6.2.3. Cloud Renderer report

Once the Cloud Renderer has finished the audio rendering process, it calls the “audioRenderingReport” ACM web service with a JSON structure that includes information of the different audio rendered files and their location. The JSON structure is then parsed by this ACM web service and the audio files are mapped into the database:

- Method: “POST”
- Data type: JSON
- Parameters:
 - **assetId**: ID of the ACM Asset.
 - **status**: “processing”, “finished” or “error”.
 - **outputFormat**: “0+2+0” (stereo), “FOA” (First Order Ambisonics) or “binaural”.
 - **output**: location of the rendered audio
 - parameters
 - **Pos**: “Dynamic”, “VOG” (Classic), “Friend” (Static)
 - **Gain**: “low”, “medium”, “high”

The JSON structure of the received call is as follows, it is noteworthy to mention that below code is purely an example extracted from ACM with aims to demonstrate the parameters and their possible value:

```

{
  "assetId": 404,
  "Message": [
    {
      "status": "processing",
      "phi": 0,
      "uid": "943194cfbc708a96d9a335f9202eb86c",
      "parameters": {

      },
      "title": "357VIDEO-static-0",
      "outputFormat": "0+2+0",
      "sceneFile":
"943194cfbc708a96d9a335f9202eb86c/working_dir/12584537636305156995.xml",
      "output": [
        "https://imacplatformserver.com/stereo/aac/357VIDEO-static-0.aac"
      ],
    }
  ]
}

```

```

    "rotationMode": "NONE"
  },
  {
    "status": "closed",
    "phi": 0,
    "uid": "943194cfbc708a96d9a335f9202eb86c",
    "parameters": {

    },
    "title": "357VIDEO-static-0",
    "outputFormat": "binaural",
    "sceneFile":
"943194cfbc708a96d9a335f9202eb86c/working_dir/12584537636305156995.xml",
    "output": [
      "https://imacplatformserver.com/binaural/aac/357VIDEO-static-0.aac"
    ],
    "rotationMode": "NONE"
  },
  {
    "status": "processing",
    "phi": 0,
    "uid": "943194cfbc708a96d9a335f9202eb86c",
    "parameters": {

    },
    "title": "357VIDEO-static-0",
    "outputFormat": "foa",
    "sceneFile":
"943194cfbc708a96d9a335f9202eb86c/working_dir/12584537636305156995.xml",
    "output": [
      "https://imacplatformserver.com/foa/357VIDEO-static-0.mp4"
    ],
    "rotationMode": "NONE"
  },
  {
    "status": "closed",
    "audioDescriptionLanguage": "ca_ES",
    "phi": 0,
    "uid": "943194cfbc708a96d9a335f9202eb86c",
    "parameters": {
      "s3": {
        "Pos": "Dynamic",
        "Gain": "medium"
      },
      "s2": {
        "Pos": "Dynamic",
        "Gain": "medium"
      },
      "s1": {
        "Pos": "Dynamic",
        "Gain": "medium"
      },
      "s7": {
        "Pos": "Dynamic",
        "Gain": "medium"
      }
    }
  },

```

```

        "s6": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s5": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s4": {
            "Pos": "Dynamic",
            "Gain": "medium"
        }
    },
    "title":
    "357_s1PosDynamic_s1Gainmedium_s2PosDynamic_s2Gainmedium_s3PosDynamic_s3Gainme
    dium_s4PosDynamic_s4Gainmedium_s5PosDynamic_s5Gainmedium_s6PosDynamic_s6Gainme
    dium_s7PosDynamic_s7Gainmedium-static-0",
    "outputFormat": "0+2+0",
    "sceneFile":
    "943194cfbc708a96d9a335f9202eb86c/working_dir/14396656611306295977.xml",
    "output": [

    "https://imacplatformserver.com/stereo/aac/357_s1PosDynamic_s1Gainmedium_s2Pos
    Dynamic_s2Gainmedium_s3PosDynamic_s3Gainmedium_s4PosDynamic_s4Gainmedium_s5Pos
    Dynamic_s5Gainmedium_s6PosDynamic_s6Gainmedium_s7PosDynamic_s7Gainmedium-
    static-0.aac"
    ],
    "rotationMode": "NONE"
},
{
    "status": "processing",
    "audioDescriptionLanguage": "ca_ES",
    "phi": 0,
    "uid": "943194cfbc708a96d9a335f9202eb86c",
    "parameters": {
        "s3": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s2": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s1": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s7": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s6": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s5": {

```

```

        "Pos": "Dynamic",
        "Gain": "medium"
    },
    "s4": {
        "Pos": "Dynamic",
        "Gain": "medium"
    }
},
"title":
"357_s1PosDynamic_s1Gainmedium_s2PosDynamic_s2Gainmedium_s3PosDynamic_s3Gainme
dium_s4PosDynamic_s4Gainmedium_s5PosDynamic_s5Gainmedium_s6PosDynamic_s6Gainme
dium_s7PosDynamic_s7Gainmedium-static-0",
    "outputFormat": "binaural",
    "sceneFile":
"943194cfbc708a96d9a335f9202eb86c/working_dir/14396656611306295977.xml",
    "output": [

"https://imacplatformserver.com/binaural/aac/357_s1PosDynamic_s1Gainmedium_s2P
osDynamic_s2Gainmedium_s3PosDynamic_s3Gainmedium_s4PosDynamic_s4Gainmedium_s5P
osDynamic_s5Gainmedium_s6PosDynamic_s6Gainmedium_s7PosDynamic_s7Gainmedium-
static-0.aac"
    ],
    "rotationMode": "NONE"
},
{
    "status": "processing",
    "audioDescriptionLanguage": "ca_ES",
    "phi": 0,
    "uid": "943194cfbc708a96d9a335f9202eb86c",
    "parameters": {
        "s3": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s2": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s1": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s7": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s6": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s5": {
            "Pos": "Dynamic",
            "Gain": "medium"
        },
        "s4": {
            "Pos": "Dynamic",

```



```

        "Gain": "medium"
    }
},
    "title":
"357_s1PosDynamic_s1Gainmedium_s2PosDynamic_s2Gainmedium_s3PosDynamic_s3Gainme
dium_s4PosDynamic_s4Gainmedium_s5PosDynamic_s5Gainmedium_s6PosDynamic_s6Gainme
dium_s7PosDynamic_s7Gainmedium-static-0",
    "outputFormat": "foa",
    "sceneFile":
"943194cfbc708a96d9a335f9202eb86c/working_dir/14396656611306295977.xml",
    "output": [

"https://imacplatformserver.com/foa/357_s1PosDynamic_s1Gainmedium_s2PosDynamic
_s2Gainmedium_s3PosDynamic_s3Gainmedium_s4PosDynamic_s4Gainmedium_s5PosDynamic
_s5Gainmedium_s6PosDynamic_s6Gainmedium_s7PosDynamic_s7Gainmedium-static-
0.mp4"
    ],
    "rotationMode": "NONE"
},
{
    "status": "processing",
    "audioDescriptionLanguage": "ca_ES",
    "phi": 0,
    "uid": "943194cfbc708a96d9a335f9202eb86c",
    "parameters": {
        "s3": {
            "Pos": "Dynamic",
            "Gain": "high"
        },
        "s2": {
            "Pos": "Dynamic",
            "Gain": "high"
        },
        "s1": {
            "Pos": "Dynamic",
            "Gain": "high"
        },
        "s7": {
            "Pos": "Dynamic",
            "Gain": "high"
        },
        "s6": {
            "Pos": "Dynamic",
            "Gain": "high"
        },
        "s5": {
            "Pos": "Dynamic",
            "Gain": "high"
        },
        "s4": {
            "Pos": "Dynamic",
            "Gain": "high"
        }
    },
    "title":
"357VIDEO_s1PosDynamic_s1Gainhigh_s2PosDynamic_s2Gainhigh_s3PosDynamic_s3Gainh

```

```

igh_s4PosDynamic_s4Gainhigh_s5PosDynamic_s5Gainhigh_s6PosDynamic_s6Gainhigh_s7
PosDynamic_s7Gainhigh-static-0",
    "outputFormat": "0+2+0",
    "sceneFile":
"943194cfbc708a96d9a335f9202eb86c/working_dir/6136371155426622975.xml",
    "output": [

"https://imacplatformserver.com/stereo/aac/357VIDEO_s1PosDynamic_s1Gainhigh_s2
PosDynamic_s2Gainhigh_s3PosDynamic_s3Gainhigh_s4PosDynamic_s4Gainhigh_s5PosDyn
amic_s5Gainhigh_s6PosDynamic_s6Gainhigh_s7PosDynamic_s7Gainhigh-static-0.aac"
    ],
    "rotationMode": "NONE"
  }
]
}

```

The location of the rendered audio files and its parameters are stored in the ACM database, so they are available for packaging (chapter 6.3).

6.3. PACKAGING PROCESS

Once all the audiovisual and the access service files (ST, AD, SL) are ready for publication, the user will see that the desired Asset in CM has the “Publish” button enabled in a blue background colour (or green if it has been used before). Illustration 14 demonstrates the corresponding button:

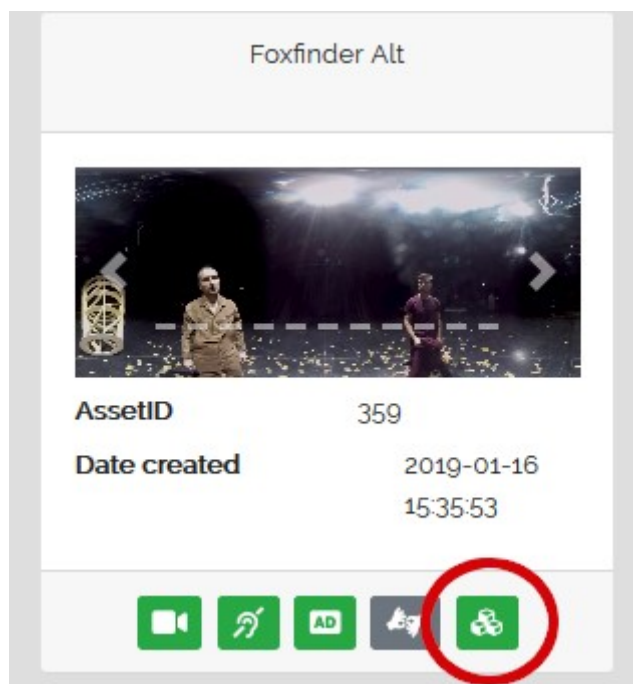


Illustration 14: Request for publish

When clicked, a window (illustration 15) will appear with the publication details, that is a list of the files available for different languages. If the requirements are met, the user can start the

process by clicking the “Publish” button.

Illustration 15 upper section corresponds to the audio files in different formats (stereo, FOA and binaural) and without audio description.

The bottom section of the illustration displays the audio files mixed with the AD and with different gains (low, medium and high) and positions (static -friend-, dynamic and classic -VoG-). Also the subtitle and sign language files are displayed in this section. All of them are available for each language.

Asset publication: **359 Foxfinder Alt**

Status

In progress

Packager message

Packager last modified

2019-02-20 15:07:12

Video Resolutions

1080.720.540.480

Save

	Main 360 Video low		
	Main 360 Video medium		
	Main 360 Video high	Finished	2019-02-12 10:05:53
	Stereo Main Mix	In progress	2019-02-20 15:07:12
	FOA Main Mix	In progress	2019-02-20 15:07:12
	Binaural Main Mix	In progress	2019-02-20 15:07:12
	ADM File		

Spanish
Catalan
Galician
British English
French
German
Polish
American English
Dutch

Norwegian
Portuguese

	Stereo Main Mix + AD Pos: center, volume: low	In progress	2019-02-20 15:07:12
	Stereo Main Mix + AD Pos: center, volume: medium	In progress	2019-02-20 15:07:12
	Stereo Main Mix + AD Pos: center, volume: high	In progress	2019-02-20 15:07:12
	FOA Main Mix + AD Pos: Friend, volume: low	In progress	2019-02-20 15:07:12
	FOA Main Mix + AD Pos: Friend, volume: medium	In progress	2019-02-20 15:07:12
	FOA Main Mix + AD Pos: Friend, volume: high	In progress	2019-02-20 15:07:12
	Binaural Main Mix + AD Pos: Friend, volume: low	In progress	2019-02-20 15:07:12
	Binaural Main Mix + AD Pos: Friend, volume: medium	In progress	2019-02-20 15:07:12
	Binaural Main Mix + AD Pos: Friend, volume: high	In progress	2019-02-20 15:07:12
	Audio Description German Friend	Finished	2019-02-14 12:30:16

publish

Illustration 15: Prepack

6.3.1. Requirements for packaging

Packaging is a procedure in which the preparation for later content streaming. This process is needed because the player uses DASH packaging and streaming.

For the procedure to start, a set of requirements needs to be fulfilled:

- At least one ST or SL file with “Verified” status.
- Audio rendered files by Cloud Renderer in case of AD (see chapter 6.2).
- High Resolution video mapped into the database (see chapter 6.1). If HQ does not exist user will receive an error and it is not possible to start the procedure of distribution.
- The user needs “publish” permission.

6.3.2. JSON structure sent to Packager

The JSON structure is the same for AD, ST and SL, but depending on the files that ACM is requesting to package, one or more accessibility files can be present.

The ACM sends to the Packager in the JSON structure all the information that it needs in order to prepare the Asset for its distribution, as well as some metadata that the ImAc Player (the 360° ImAc Player is described in detail in the D3.3 deliverable^{vi}) will need in its user interface:

- Method: “POST”
- Data type: JSON
- Parameters:
 - **assetId**: The ID of the ACM Asset
 - **callbackUrl**: The URL that the Packager needs to call back when the process has been finished, so that ACM can change the status of the Packaging process.
 - **programmeName**: The name of the Asset that will be shown in the Player.
 - **language**: main language for the asset.
 - **keyframe**: A video capture image for the Player.
 - **poster**: Image to present at the Player. Defaults to “./img/u25_c.png”
 - **descriptionArray**: Array with the different descriptions of the asset in the different languages. (Needed by the player)
 - **acces**: Array that states the available languages for each access service (needed by the player):
 - “AD” : [“ca_ES”, “en_GB”]
 - “SL”: [“ca_ES”]

- **"ST":** ["ca_ES", "es_ES", "en_GB"]
- **files:** The array of files that the asset has available for the Packager.
 - **mainVideo:** the HQ video
 - **url:** the HQ video file URL
 - **description:** the description of the type of file.
 - **urn:mpeg:dash:role:2011 :** the role of the file: "main"
 - **transcode:** transcoding resolutions: "2160", "1080", "720", "540", "480"
 - **subtitle:** the subtitle files. More than one can be present.
 - **url :** the subtitle file URL
 - **description:** the description of the type of file: "IMSC 1.0 English Hard of Hearing"
 - **urn:mpeg:dash:role:2011:** the role of the file: "caption"
 - **language:** the language of the subtitling: "en_GB", "es_ES", "ca_ES"...
 - **audio:** the audio files. More than one can be present.
 - **url :** the audio file URL
 - **description:** the description of the type of file: "FOA Main Mix + AD Pos: Friend, volume: medium", etc.
 - **urn:mpeg:dash:role:2011:** the role of the file: "alternate"
 - **language:** the language of the audio: "en_GB", "es_ES", "ca_ES"...
 - **audioFormat:** the format of the audio file: "stereo", "binaural", "FOA"
 - **ADposition:** The AD presentation mode: "VOG" (Classic), "Friend" (Static), "Dynamic"
 - **ADgain:** Audio Description gain: "low", "medium", "high"
 - **containsAD:** Wether this is an AD file or not: "0" or "1"
 - **signer:** the sign video files. More than one can be present.
 - **url :** the sign video file URL
 - **description:** the description of the type of file: "Sign Language Catalan"
 - **urn:mpeg:dash:role:2011:** the role of the file: "signer"
 - **language:** the language of the signer: "en_GB", "es_ES", "ca_ES"...

Examples of JSON structures:

In most cases, the notification to the packager will include more than one access service file, as the asset might be available with one or several ST, AD and SL files simultaneously.

In the following example, an AD and a ST files are provided to the Packager:

```
{
  "result": 1,
  "msg": "Asset 312 Pre-Pack files successfully requested",
  "assetId": 312,
  "programmeName": "Life on Mars",
  "keyframe": "https://imacplatformserver.com/acm/instances/keyframes/497288-
videoname-low.mp4_10.jpg",
  "poster": "./img/u25_c.png",
  "descriptionArray": {
    "ca_ES": ""
  },
  "acces": {
    "AD": [
      "ca_ES"
    ],
    "ST": [
      "ca_ES"
    ]
  },
  "language": "ca_ES",
  "files": {
    "mainVideo": [
      {
        "url": "https://imacplatformserver.com/acm/instances/video_hq/497288-
videoname.mp4",
        "description": "Main 360 Video high",
        "urn:mpeg:dash:role:2011": "main",
        "transcode": [
          "2160",
          "1080",
          "720"
        ]
      }
    ],
    "audio": [
      {
        "url": "https://imacplatformserver.com/acm/instances/audiodescription/
302_videoname_ca_ES_VOG_20180711-155753_20180906-162554.ad",
        "description": "Audio Description Catalan VOG",
        "urn:mpeg:dash:role:2011": "alternate",
        "audioFormat": "",
        "ADposition": "",

```

```

        "ADgain": "",
        "containsAD": "1",
        "language": "ca_ES"
    }
],
"subtitle": [
    {
        "url":
"https://imacplatformserver.com/acm/instances/subtitle/281_Participant_37_en_2
0180720-104420_20180905-103231.xml",
        "description": "IMSC 1.0 Catalan Hard of Hearing",
        "urn:mpeg:dash:role:2011": "caption",
        "language": "ca_ES"
    }
]
},
"callbackUrl": "https://imacplatformserver.com/acm/ws/callbackPackager.php"
}

```

In the following example, an AD file is provided to the Packager:

```

{
    "result": 1,
    "msg": "Asset 621 Pre-Pack files successfully requested",
    "assetId": 621,
    "programmeName": "Holy Land 4 AD Catalan",
    "keyframe":
"https://imacplatformserver.com/acm/instances/keyframes/videoname-AD-
Catalan_injected-low_20190523-181630.mp4_10.jpg",
    "poster": "./img/u25_c.png",
    "descriptionArray": {
        "ca_ES": ""
    },
    "acces": {
        "AD": [
            "ca_ES"
        ]
    },
    "language": "ca_ES",
    "files": {
        "mainVideo": [
            {
                "url":
"https://imacplatformserver.com/acm/instances/video_hq/videoname-E04-CAT-mm-
5_1.mp4",
                "description": "Main 360 Video high",
                "urn:mpeg:dash:role:2011": "main",
                "transcode": [
                    "2160",

```

```

        "1080",
        "720"
    ]
}
],
"audio": [
{
    "url": "https://imacplatformserver.com/acm/instances/audiodescription/582_video_name_4_en_US_VOG_20190425-103843_1__20190531-102235.ad",
    "description": "Audio Description Catalan Dynamic",
    "urn:mpeg:dash:role:2011": "alternate",
    "audioFormat": "",
    "ADposition": "",
    "ADgain": "",
    "containsAD": "1",
    "language": "ca_ES"
}
]
},
"callbackUrl": "https://imacplatformserver.com/acm/ws/callbackPackager.php"
}

```

In the following example, a SL file is provided to the Packager:

```

{
    "assetId": 351,
    "callbackUrl":
"https://imacplatformserver.com/acm_test/ws/callbackPackager.php",
    "programmeName": "Life on Mars",
    "language": "ca_ES",
    "keyframe":
"https://imacplatformserver.com/acm_test/instances/keyframes/497288-videoname-low_20181025-154126.mp4_10.jpg",
    "poster": "./img/u25_c.png",
    "descriptionArray": {
        "ca_ES": ""
    },
    "access": {
        "SL": [
            "ca_ES"
        ]
    },
    "files": {
        "mainVideo": [
            {
                "url":
"https://imacplatformserver.com/acm_test/instances/video_hq/497288-videoname-low_20181025-154126.mp4",

```



```

        "description": "Main 360 Video high",
        "urn:mpeg:dash:role:2011": "main",
        "transcode": [
            "2160",
            "1080",
            "720"
        ]
    },
    ],
    "signer": [
        {
            "url":
"https://imacplatformserver.com/acm_test/instances/signedlanguage/31/",
            "description": "Sign Language Catalan",
            "urn:mpeg:dash:role:2011": "signer",
            "language": "ca_ES"
        }
    ]
}
}

```

6.3.3. Packager response

Once the packaging process has been finished, the files are made available for the ImAc Player. The Packager notifies to the ACM the result, so that the user can be informed that the process has been finished and if there has been any errors.

The following is an example of the JSON structure received:

```

{
    "result": "1",
    "assetId": "357",
    "msg": "Packaging process finished",
    "language": "de"
}

```

ANNEX I: TECHNICAL TERMS

Term	Description
Variables	Variables to be registered in the database destined to a specific functionality
Engines	Engines are tools that run in the background without any user interface.
Static	Type of AD – AD anchored in the scene (also known as “Friend”)
Dynamic	Type of AD – AD comes from where the action is
Classic	Type of AD – AD centred in the scene (also known as “VoG”)
Time code	The producer decides the exact point of the video where the playback of access service items will start
Language locale	The abbreviation in which the language is defined in the code (i.e. ES for Spanish)
Keyframes	Images of video frames used by the user for a quick preview of the content.
Transcode	The process in which an uploaded high resolution video is transcoded into a capable HTML5 LQ video for being used in ACM and the Web Editors.

ANNEX II: ACM BACKGROUND PROCESSES

Background processes	Description
Video transcoding to low quality	This background process transcodes the HQ video that is uploaded to the ImAc server into the LQ video that ACM will automatically import. The process is triggered by the XML file that contains information for the transcodification and also the destination of the LQ video in ACM.
Cloud Renderer for audio rendering	For Audio Description files, once the describer has finished producing the AD file for at least one of the types (Classic, Static or Dynamic), ACM requests the audio rendering to the Cloud Renderer (to mix the AD audio with the 360° programme audio), so that it will be available later on for the Packager.
Packager for content distribution	The Packaging is the final step for content distribution, which gathers all the necessary files and metadata from ACM and prepares and processes them for the distribution to the ImAc Player.

REFERENCES

- i ImAc Deliverable 2.3. Platform Specification, Francesc Mas, 2018.
- ii ImAc Deliverable 4.1. Subtitle Production Tools, Peter tho Pesch, 2018.
- iii ImAc Deliverable 4.2. Audio Production Tools, Peter tho Pesch & Theresa Liebl, 2018.
- iv ImAc Deliverable 4.3. Sign Language Editor, Kimiasadat Mirehbar, 2018.
- v ImAc Deliverable 3.3 Content Packaging and Distribution, Marc Brelot, Romain Bouqueau, Rodolphe Fouquet, 2018.
- vi ImAc Deliverable 3.5 Player, Mario Montagut 2018.