Deliverable

Project Acronym:	**ImmersiaTV**
Grant Agreement number:	688619
Project Title:	*Immersive Experiences around TV, an integrated toolset for the production and distribution of immersive and interactive content across devices.*

D2.3 Content ideation, production scenarios and requirement analysis

Revision: 0.9

Authors:

Luk Overmeire (VRT)

Joan Llobera (i2CAT)

Delivery date: M03

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement 688619

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

Abstract: This Deliverable addresses the content ideation and production scenarios for pilot 1 of the ImmersiaTV project, as well as a requirement analysis regarding software development needs that need to be considered to create and deliver synchronous experiences across devices (HMD, tablet, TV) based on omnidirectional video.

It specifies the software needs for pre-production, for content editing (both using a simple editor and a more advanced compositing tool), and to deliver the experience in different devices (HMD, tablet, TV).

REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	15/02/16	L.Overmeire	VRT	ToC
0.2	02/05/2015	J.Llobera	i2CAT	first version including all the contributions
0.3	04/05/2016	S.Fernandez	i2CAT	First review
0.4	09/05/2016	J.Llobera	i2CAT	Review integrating comments and suggestions
0.5	10/05/2016	L.Overmeire	VRT	Detailed review
0.6	12/05/2016	J.Llobera	i2CAT	Review integrating all new comments and suggestions
0.7	01/06/2016	P.Pamplona	i2CAT	Review of style and ambiguities in content
0.8	02/06/2016	J.Llobera	i2CAT	Addressing the new comments
0.9	06/06/2016	P.Pamplona	i2CAT	Template and format improvements

Disclaimer

The information, documentation and figures available in this deliverable, is written by the **ImmersiaTV** (*Immersive Experiences around TV, an integrated toolset for the production and distribution of immersive and interactive content across devices*) – project consortium under EC grant agreement H2020 - ICT15 688619 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This document contains original unpublished work except where clearly indicated otherwise.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

This Deliverable addresses the content ideation and production scenarios for pilot 1 of the ImmersiaTV project, as well as a requirement analysis regarding software development needs that need to be considered to create and deliver synchronous experiences across devices (HMD, tablet, TV) based on omnidirectional video.

First, it addresses the process and options available for content ideation. For this purpose, it describes the main considerations a content creator needs to address regarding content ideation, and the consequences that choosing one or another option has regarding content production, editing and post-production, for pilot 1, concerned with an offline documentary. The range of options goes from shooting with a Chroma key and assume a large amount of post-production work, to shoot everything with miniaturized cameras and allow for small post-production.

After introducing the range of options available to the content creator, the content specifically selected for Pilot 1 is introduced, reasoning the choice adopted, complemented with a detailed script. Given the constraints introduced, and the willingness to introduce multi-platform content throughout the whole experience, the most appropriate strategy seems to be to build a fictionalized documentary. This means, using actors, instead of general people, and use directive micro-cameras combined with omnidirectional shots. This production strategy allows keeping some of the spirit of the documentary, including little post-production, and a relatively small crew.

After the content ideation, this deliverable describes typical user scenarios of content creation, and the consequences these user scenarios have in terms of software specification in the context of content creation within the scope of ImmersiaTV (multi-platform content based on omnidirectional video). Given the currently available commercial and free tools available, contrasted to the requirements gathered in the different workshops, we have detected the need for developing three tools:

1) a tool for production preparation,
2) an edition tool allowing the creation of synchronous omnidirectional and directive content targeting different platforms, and
3) a multi-platform player allowing to visualize the content created across devices.

This last software tool, the multi-platform player, will be used both in a production environment to create content and by the end-user to experience it.

User Scenarios and specific requirements in the forms of user stories are also developed for these different scenarios. Finally, a third part lists the software requirements, and will be used in Work Package 3 to design the ImmersiaTV software architecture.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Maria	Pacheco	Lightbox	maria@lightbox.pt
David	Cassany	i2CAT	david.cassany@i2cat.net
Wim	Forceville	Fisheye	wim@fisheye.eu
Wout	Standaert	Fisheye	wout@fisheye.eu
Luis	Ferreira	Lightbox	ferreira@ligthbox.pt
Helder	Campos	Lightbox	helder@lightbox.pt
Rui	Sousa	Lightbox	rui@lightbox.pt
Rik	Bauwens	VRT	Rik.bauwens@vrt.be
Tom	Cornille	VRT	Tom.cornille@vrt.be

CONTENTS

TABLE OF FIGURES

LIST OF ACRONYMS

Acronym	Description
HMD	Head-mounted display
VR	Virtual reality
WP	Work Package
CGI	Computer Graphic Image

1. INTRODUCTION

1.1. Purpose of this document

This deliverable addresses the content ideation and production scenarios for pilot 1 of the ImmersiaTV project, as well as a requirement analysis and specification regarding software development needs to create for pre-production, capturing, post-production and to deliver synchronous experiences across devices (HMD, tablet, TV) based on omnidirectional video.

This document is organized in a set of structured insights and requirements that together with the outcomes of Tasks 2.1 and 2.2, as well as the content ideation process in Task 2.3, define the software requirements regarding content production and end-user experience.

1.2. Scope of this document

The ImmersiaTV DOW already outlines several details regarding the end-user experience and the production tools which are relevant for this document. At least the following sections of the DOW are relevant:

Page 8, section 1.3.1, Concept:

This project will use omnidirectional video enriched with novel techniques of audiovisual production to deliver a novel form of Broadcast content that matches the demands of immersive displays, and can be shared with tablet and traditional TV consumers.

Using a head mounted display it is possible to render several video streams, not necessarily omnidirectional, simultaneously, smartly inserted within its very large field of view. These inserts would be experienced as audiovisual portals, which would appear, grow, cover the whole field of view or disappear, depending both on the storyteller's choices and end-user behaviour. Using this technique, the solidly proven techniques used to build narratives within an audiovisual production – close shot to show the reaction of the main characters, slow motion to repeat a crucial moment, etc.- can still be used in the context of immersive displays, where cuts between omnidirectional shots would provoke discomfort.

It should be noticed that the previous choices in content format do not prevent these experiences to be broadcast live, and we will demonstrate the benefits of this approach both for offline and live production.

Page 129, section 1.3.2.1 Content Format:

The use of audiovisual portals will be complemented with the delivery of content in 2 temporal modes: broadcast and exploration.

A broadcast mode, where the timing and order of the scenes and events forming the broadcasted content will be fixed at the production stage. The broadcasted content will therefore be shared across devices, even if the end-user will still have some freedom to choose what portion of the scene he looks at.

Therefore the experience is not coherent across devices: each device user will choose freely the scenes he will receive. An exploration mode, where each end-user, through his head movements or by moving his tablet, will be able to navigate across scenes, affect their order as well as the timing of events are delivered. In exploration mode, there are different paths that the end-user can explore.

Page 134, section 1.3.2.6 Display and Interaction:

In addition, it will send to the server end-user anonymized information in order the codecs can define a region of interest. This information will also enable the end-user to send through social media links of pictures or videos of the user's explorations. In other terms: if the end-user want to create a video of a particular viewpoint, or sequence, he or she will only send high-level information back to the central server, where the appropriate video sequence will be generated, and made available through a unique link automatically generated in the end-users device, to allow him or her integrating this experience within his social media channels.

The end-user will:

- *In an immersive display, have access to several "portals". Some of them will appear as a result of the end-users' actions, some will have been determined beforehand by the storyteller.*
- *At certain moments be able to choose one of several video sources through head movements, or tablet movements*
- *In an immersive display, the user will always be able to move his head around to explore the main omnidirectional image*
- *In a tablet the user can pan around, and zoom. He can also share a video of his individual explorations through social media*

In addition, this software solution will be able to handle synchronization across devices within a completely distributed architecture.

Page 95, Section 1.3.3. Work package descriptions:

Traditionally, the creation process of a TV production is as follows: define the story, research the story, write the scenario, prepare the shooting logistics, write call sheet (people planning), capture the story, post-production of the captured material and distribute the finished program.

In the context of ImmersiaTV, we have to gradually develop and fine-tune this creation and post-production process, in order to include immersive content and to cater for different immersive devices. Production tools, format creation and scenario mechanisms, interaction design and the basics of the new cinematographic language have to be developed and adapted to maximize the engagement of the audience with the immersive content.

1.3. Relation with other ImmersiaTV activities

Based on the structured insights in end user and professional user requirements reported in deliverables D2.1 and D2.2, this document is the result of the analysis of ImmersiaTV content formats, production strategy and requirements, focused on the technical workflow to realize these scenarios. This document will be used within WP3 for the software architecture design (Task 3.1), as well as for the content creation, the pilots, and the technical evaluation. The relationship between this task and the other related ImmersiaTV tasks is shown below.

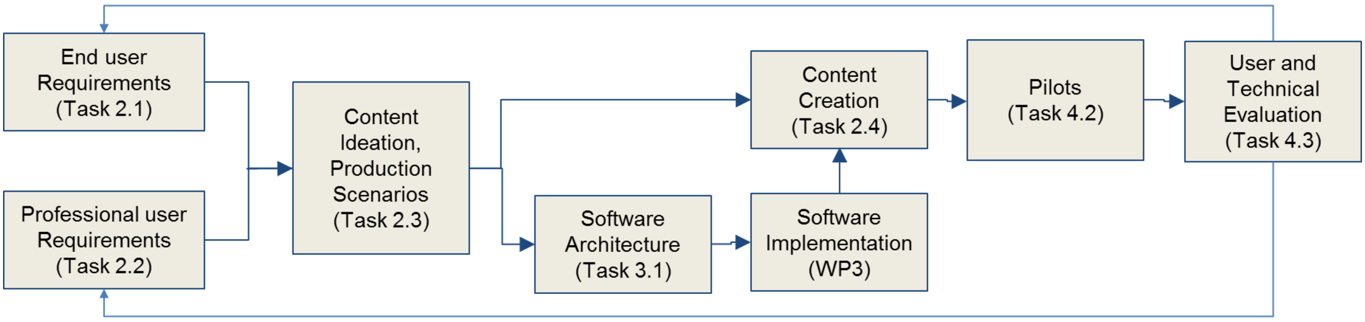
End user Requirements (Task 2.1)
Professional user Requirements (Task 2.2)
Content Ideation, Production Scenarios (Task 2.3)
Software Architecture (Task 3.1)
Content Creation (Task 2.4)
Software Implementation (WP3)
Pilots (Task 4.2)
User and Technical Evaluation (Task 4.3)

Figure 1: Relation between this deliverable and other WP2 tasks

2. CONTENT IDEATION

We now detail considerations that need to be taken into account to define a production strategy to create immersive content in a multi-device environment (HMD, TV, tablet).

2.1. Shooting options

Shooting content that works both for omnidirectional and directive formats (i.e. traditional 16:9 format) imposes several restrictions on how to shoot. In proof of concept shootings, we have learned that it is very hard to shoot them separately and then montage them to the corresponding (same) time window afterwards. The actors cannot perfectly synchronize their performance of one take to the performance of another take of the same shot. It also gives problems with audio. If the performance was not perfectly in sync, the viewer might miss certain parts of dialogue, or hear parts of dialogue two times when switching between media.

Therefore we suggest that directive shooting and 360° shooting for ImmersiaTV happens at the same time. However, since it is often not desirable that the camera crew needed for a directive camera shooting is present in the omnidirectional video, it is necessary to decide which shooting strategy will be adopted early on, since this will have a significant impact in the rest of the content production. Several workflows are possible.

2.1.1. The 360° camera doubles as a directional camera

The output product of a 360° camera does not necessarily need to be 360° video. Using the 360° video that it produces, we can place a virtual camera inside this 360° view that takes a 16:9 cut-out of any desired orientation and zoom level. The virtual camera can be fixed or animated, and traditional cuts can be made by switching to different camera rotations in time.

Advantages:

- There is no directive camera crew, so no additional postproduction is needed to remove them from the 360° view.
- Camera movement that is not possible with a traditional camera can be applied, e.g. very robotic camera movement or camera movement with a certain easing. These are possible with simple animations in After Effects.

Disadvantages:

- All camera standpoints of the directive edit originate from the same standpoint, so creative shots are limited.
- The maximum zoom level or close-up level is restricted by the resolution of the 360° camera.
- The performance of the actors has to be checked by the actors themselves, or a wireless live preview system needs to be used so the director can check if the performance was satisfactory.

2.1.2. 360° camera shoots a clean plate on scene to remove crew in post-production

On location, a clean plate can be shot by the omnidirectional camera. This is a certain duration of video where no crew or actors are present. This video can later be used to superimpose on the directional camera crew to hide them from view.

Advantages:

- The directional camera crew has much more freedom with their standpoints and framing
- Director can be present during the shoot to check the performance of the actors

Disadvantages:

- Using the clean plate to mask away the crew adds a very significant amount of work in post-production
- When the actors cross in front of crew members, the actors have to be cut out frame-by-frame in a process called rotoscoping1
- If the lighting conditions change during capturing, the clean plate will no longer match the lighting conditions of the actual shot, so it cannot be used anymore to mask away the crew. Therefore this method is only viable when there is a sufficient amount of control over lighting conditions.

2.1.3. The crew is visible in the 360º shot

For certain types of content, it is not needed to hide the crew. It is natural that they are there. For example in a news gathering situation, the 360 camera can give an extra perspective to an interview with a person. The interviewer, traditional camera man and boom operator are in interaction with the interviewee for a "behind the scenes" view.

This is also the expected workflow to capture a live event, like a soccer match. Other camera crews are in full view of the 360° camera but they are expected to be there.

Advantages:

- Gives a unique perspective on the scene
- Lends a lot of credibility to the report - nothing is hidden

Disadvantages:

- Only applicable to a very limited type of content

2.1.4. The crew hides behind strategically placed objects in the scene

Objects or walls that occlude the directional camera crew could be used in the scene.

Advantages:

- The crew is hidden with minimal post-production
- The directional camera crew has more freedom in making shots

Disadvantages:

- Objects might be taking up a very prominent space in the scene
- Crew has to be wary of shadows and reflections
- Requires more planning - might be more suited for a fiction narrative were the same set is reused

[1] https://en.wikipedia.org/wiki/Rotoscoping

2.1.5. Mini cameras are placed on the tripod

This is a variation on workflow 2.1.1, where all shots will essentially be from the same point of view. Here we use higher quality cameras to overcome the resolution limitations of workflow 1.

Advantages:

- Since the tripod has to be removed in post anyway, this option adds minimal extra post-production
- Offers better quality than a 360° camera cut-out

Disadvantages:

- Framing cannot be corrected by a camera operator
- No camera movement possible except for motorized rigs
- Everything is filmed from the same standpoint

2.1.6. Micro cameras are placed on the set

As a variation on the 2 previous workflows, this configuration allows hiding micro cameras around the set and putting them hidden behind props (ex. inside a fruit basket, on a balcony between plates). This approach does not need (or directly excludes) the presence of any crew member on set during the filming.

Since on the set there is only the tripod for the 360° video and the hidden cameras, minimal post-production is required. However, there must be a very clear storyboard and extensive planning on the placement of the micro cameras taking into account the movements of the actors.

Advantages:

- Minimal post-production
- The directional camera crew has more freedom in preparing shots, giving more options in the editing phase
- The director is free to add as many cameras he wants to cover all the actions
- Several standpoints are available for shooting

Disadvantages:

- Framing cannot be corrected by a camera operator
- No camera movements (panning, tilting, etc.) are possible
- Little or no improvisation by the actors is possible

2.1.7. The scene consists of CGI and/or composited video

Not all productions require a full 360° video setup. Digital environments can be used in conjunction with CGI characters or Chroma key video. A good example of this is the David Attenborough Giant Dinosaur video by BBC One: https://www.youtube.com/watch?v=rfh-64s5va4 (see also Figure 2).

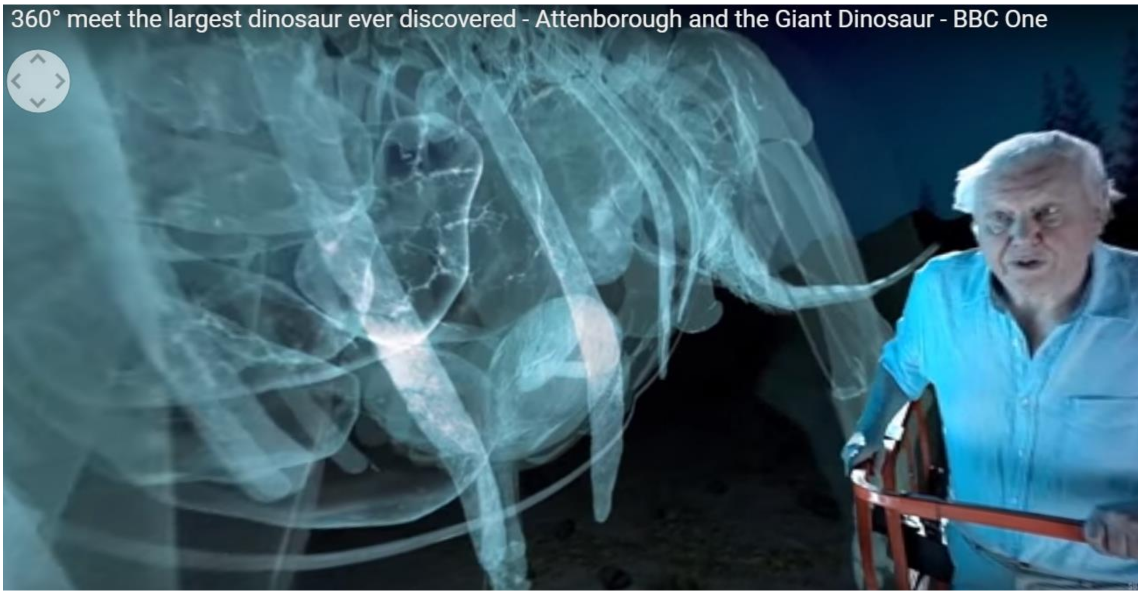
360° meet the largest dinosaur ever discovered - Attenborough and the Giant Dinosaur - BBC One

Figure 2: Sample of the video from BBC – Attenborough and the Giant Dinosaur

Attenborough is standing on a platform in front of a Chroma key background, filmed with a directive camera. This footage is then inserted into a fully CG environment. The environment doesn't have to be CGI, it can also be based on 360° photography or video.

Advantages:

- Allows for very complex shots
- Full control over the environment

Disadvantages:

- Allows less interaction with the environment by the actors
- Lots of post-production required
- Lots of planning required
- Only for high-budget productions

2.2. Interaction within and between devices

There are different platforms involved in ImmersiaTV content experiences:

1) television,
2) second screen and
3) head mounted display

The main milestone to achieve in pilot 1 is synchronized content, created offline, delivered across these three devices. From the outcome of the user workshops and internal brainstorming, the following interaction mechanisms, both between and within devices, can already be taken into account to specify the requirements in terms of interaction. We will generically refer to one of these interaction mechanisms using the term *ImmersiaTV Scene Typology.* These are the different possibilities considered:

1) A regular broadcast on TV, and a multi-camera selection menu on the second screen. Clicking on one of the menus in the second screen triggers a content on the main screen and/or on the second screen. Optionally, there is an option to send it to the TV (it does not need to happen always).

2) The viewer is presented a map view in an HMD: he can see all the video streams available and looking into one takes him in that stream. The map view is activated either by looking to a specific visual item, by using an overlay or with a button on the HMD
3) An experience in an immersive display, where the viewpoint selected by the user by moving his head is also shown in the television. As an alternative, this viewpoint could appear as a picture in picture.
4) Content created synchronously live (for example, covering a live event with traditional and directive cameras) can also be experienced synchronously live through different devices in the corresponding formats. The same concept can be applied for off-line content.
5) A tablet user can see the content just as if it is a traditional TV documentary, but when trimmed omnidirectional shots are shown on the tablet he is able to "look out of the frame" simply by moving his tablet (or dragging with the fingers on the tablet). For tablets, it would be enough to point this fact with a visual mark superimposed on the content.
6) An omnidirectional production in the HMD and an insert that is a window on the TV content. Optionally, looking at it switches to a TV view in the HMD, with an insert that allows switching back to omnidirectional view.
7) The TV shows a portion of the omnidirectional video, and the tablet shows the same. Dragging in the tablet changes the view in the TV.

It should be noticed that the previous list is not exhaustive: the content creator should be able to create its own *ImmersiaTV scene typology* within the content creation process.

2.3. Pilot 1 Content production scenario

At the current stage of the production, the following points are clear:

- The project will be a fictionalized documentary. This is the only option possible given the production constraints (see section 2.1 Shooting options).
- Content that is synchronous across devices will be available all the time.
- Production wise, we will use the option described in section 2.1.6. Several directional micro cameras will be placed around the set given the (controlled) actors positions and movements. This avoids having any character (Crew, Director) on scene that are not part of the script. In post-production, we will only have to remove the tripod of the 360° camera and the micro cameras from the footage.
- For omnidirectional video recording we will use the 6xGOPRO RIG. The assembly will consist of 6 cameras put together in a RIG attached to a tripod or monopod to film the scenes.
- The filming crew and the set decoration crew will have to work together from pre-production to post-production not only on the desired footage and style the director wants for the project but also on the planning of where to put the micro cameras and how to hide them with props.

We will have the following crew on set: director, director of photography, camera operator, assistant camera operator, make-up assistant, digital image technician, art director, wardrobe director and producer.

We now introduce a detailed script for this fictionalized documentary, corresponding to the scenes planned and the content available in each device.

Detailed scenario description

Immersia TV
DRAGON FORCE: the Making of future heroes
Total time: 15 minutes

—

Storyline:
In this documentary, we follow the steps of David, a young Portuguese athlete who joined Dragon Force, FC Porto football school, to pursue his dream of becoming a successful football player. During his busy, hard-working days, we will meet his family, his friends and the dedication of this 14 year old dreamer. Shot in immersive technology, "Dragon Force: The Making of Future Heroes" gives you a literal inside view of what it takes to become one of the great.

—

SCENE 1
KID'S ROOM
Time: 1 minute
Camera: 360º
It's early in the morning, the room is still dark and the kid is sleeping in his bed. Viewers can hear footsteps coming from one side (the side of the door). Someone opens the door.
[**TV**: user will hear the steps and a transition will enable them to see the mother walking into the room and waking up her son. It is possible to include an icon alerting TV viewer that he can get extra info on other device.]
[**HMD**: The footsteps will indicate that the user can look the way where the sound (steps) comes from by moving his head. He will see the mother opening the door and entering the room. Spatialized sound.
[**Tablet**: user can see extra info about the location. For example: Portugal, Porto, 07:30, Silva's house.]
Mother enters the room, stays a little while watching her son sleeping and starts to wake him up.
Mother:
David, it's time for you to wake up.
The kid initially turns his face around but then he raises his head and looks for his football gear the other side of the room.
[**TV**: 4 shots: sleeping; door opening; mother talking; kid looking at the football gear; kid getting up.]
[**HMD:** Can watch what the kid is watching.]
[**Tablet**: Can watch what he is watching.]
The kid gets up and goes to the kitchen for breakfast.

—

SCENE 2
FAMILY IN THE KITCHEN
Time: 1 to 2 minutes
Camera: 360º (first person view, like if the user was sitting with the family at the table)
Conversation between the members of the family around the kitchen table while having breakfast. Mom (Teresa), dad (Miguel), the kid (David) and his sister (Maria).
Users can hear other noises such as a cat (Emilio) meowing.

The conversation is a normal one, from a family point of view. Father asks David how is it going in the football club. The sister says she wants to join too. The family smiles and say she can, when she's a little bit older.
Mother alerts kid for the fact that school is very important. That his grades should be good, not only good at football but also at school.
The camera goes from one person to another, as they are talking.
[**TV**: subjective camera, like if the user was sitting with the family. Only sees cat when it enters framing by climbing to the table. There are some details, complementary to the story but not relevant for main narrative, that will happen only on other devices. This way we prevent the TV of feeling like he is missing out something important.]
[**HMD**: subjective camera, like if the user was sitting with the family. The HMD user can look for the cat when he hears it]
[**Tablet**: icon indication of more info of each character. Each person at the kitchen has a graphic ID: name, age, job, etc. User can look for the cat and see graphic ID about it too. When he stops at Emilio the device vibrates, a call to action, so the user can click for example on another video of Emilio and find out what it has done to the ham that was at the kitchen table.

___
SCENE 3 AND 4
STREET AND METRO
Time: 2 to 3 minutes
Camera: 360º (first person view)
The kid is walking with a sports bag to the metro station. It is a short walk.
[**TV**: user sees the kid perspective (subjective camera). Icon alerting for more info on other devices.]
[**HMD**: subjective camera. User can see the street, other people passing by, neighbors saying hello as it was for him. A portal appears with a map (left side in the HMD) indicating his location. Portal already showing coach preparing the field for the practice.]
[**Tablet**: can see the street (arrows that indicate the ability to move around, other people passing by, see graphic info of the neighborhood, ex: the location of the kid and the location of the field, how far he is.]
The kid enters the subway and sits down with a ball in his hands. The ball falls.
[**TV**: user first hears someone talking and then sees the person who catches the ball and returns it to the kid.]
[**HMD**: user can see the subway and where does the ball go. Another kid (a team colleague: Fábio) catches the ball and starts talking to the user has if he was David.]
[**Tablet**: can see the subway and where does the ball go. Another kid (a team colleague: Fábio) catches the ball and starts talking to the user has if he was David.]
Fábio returns the ball to David and sits next to him.

___
SCENE 5
DRAGON FORCE LOCKER ROOM
Time: 20 to 30 seconds
Camera: 360º
At this scene all users will feel what it's like to be at a locker room. There will be no dialogs, just the kids getting ready for the practice. They talk, laugh, joke around. The purpose is for the users to feel the atmosphere, like a sneak peek.
Users hear a whistle and watch the kids running to the field, leaving the locker room empty.
Kids getting into the locker room. All the kids playing around and heading to the field when they hear a whistle.

___
SCENE 6

THE FIELD – INTERVIEW
Time: 3 to 4 minutes
Camera: 360º
At the field, the kid starts talking about himself.
Kid:
I'm David, I'm 14 years old and this is my left foot (laughs).
All users can hear the other kids laughing out loud and saying: *Next Messi*; *He's the best*!, …
[**TV**: user can hear the other kids but only when the plan changes they are able to see them.]
[**HMD**: when he hears the other kids, user understands that he can move his head to see the others kids, the field and the surrounding areas. Possibility to insert a portal with the option for closer view.]
[**Tablet**: user can see the others kids, the field, the surrounding areas. Dragon Force logo appears. If he taps on the logo, it will show extra info about the football school.]
The interview is conducted between coach and player. They are changing the ball while they are talking. They run, laugh, make and answer questions.
When the coach asks who is his favorite player, David answers Messi. As he talks about him, images of the international football player start to appear in the screens in different ways.
[**TV**: user can see the change of focus between each one. Icon appears when Messi's footage appears on other devices, alerting for him to connect to the tablet to watch it.]
[**HMD**: user can see an icon indicating that he can chose the point of view (from the kid, from the coach); or switch from the kid to the coach by turning his head. A portal appears with footage from Messi (archive item).]
[**Tablet**: user can see both of them or just one of them. When the kid starts talking about Messi, the user can see an icon alerting for other video of Messi, with all the player outstanding statistics (goals, awards, etc.); icon to go back at any time to the main narrative.]

___

SCENE 7
THE SCHOOL GAME
Time: 2 minutes
Camera: 360º
Shots of the game between David's team and another one. When he strikes the ball, the shot changes to the goalkeeper. He jumps to stop the ball but it just passes right trough. GOAL!
[**TV**: user can see the change of focus between each one; follows the director's choice. Watches the normal replay.]
[**HMD**: user can choose the camera's point of view (from the team bench, from the audience or in the field).]
[**Tablet:** At the replay, user can chose from two points of view: from the team bench and after the goal. A portal appears on the left, indicating heart rate and kilometers run by kid. iteration 1: footage of similar goals of top players.]

___

SCENE 8
FAMILY INTERVIEW
Time: 30 seconds
Camera: 360º
The family is in the car, driving home. During the trip, they talk about the game, how exciting it was to see David leading his team to victory.
[**TV**: user can see each person when they are talking. follows the director's choice.]

[**HMD**: user is "seated" in the backseat, like if he was on the car too. From this point of view, he can rotate the head and see the kid and his sister, one in each side. In front, he can see the mother and father. Can look away through the window.]
[**Tablet**: user can look away through the window, can focus on one person.]

___

SCENE 9
THE SCHOOL
Time: 2 to 3 minutes
Camera: 180º+180º
Another day. David is at school. Classroom shot. Omni camera shoots the class and the recess (playing football, of course).
[**TV**: user can see a master shot of the classroom with transitions to close-ups of the students and teacher.]
[**HMD**: user can see the students side of the classroom (180º) and the teacher side (180º). Teacher is talking about a mathematical concept. When the user focus on the teacher the board becomes a portal with graphic footage of a calculation: using extrapolation, the teacher will calculate the physical and stamina conditions of the kid in one month from here, if he continues his practices and training.]
[**Tablet**: user can see the students side of the classroom (180º) and the teacher side (180º). Teacher is talking about a mathematical concept, the Pythagorean theorem. User can go out or back to the classroom whenever he wants, fully aware of what he missed.
The bell rings and all the kids run to the yard. The teacher tells David to wait a little longer and starts talking with him.
[**TV**: user can see the students leaving the classroom. And then a transition to the teacher/student conversation.]
[**HMD**: user can see the students leaving the classroom and can see them outside playing but now at a certain distance. If he looks the other way, to the classroom, he can see the teacher/student conversation. The user hears in surround both the kids outside and the conversation inside. Depending on where he is focused, he hears the sound louder. This will indicate the user that he can see both scenes by moving his head in one direction or the other. Also, it can be added here the option to choose a point of view in the conversation: from the teacher or from the kid.]
[**Tablet:** user can see the students leaving the classroom and can see them outside playing but now at a certain distance. He can go outsider but he must be fully aware that he will lose the conversation.]
The teacher tells David that although he is a great promise in football, he must not forget to complete his studies. A football career is short and he must have the knowledge to continue studying so he has a brilliant future even after his career in sports.
David thanks his teacher's advice and says he knows that. Other football players are also university graduates and he wants to do the same.
[**TV**: user can watch the conversation.]
[**HMD**: user follows the conversation. When David talks about other football players who graduated, a portal appears with footage of famous footballers who finished college.]
[**Tablet:** can choose between watching the conversation of watching the kids playing outside.]
David leaves the classroom to meet with his colleagues. He comes back and says to the teacher:
Teacher, I would like you to come one day to watch a game.
Even if you don't like football.
He then runs to the field.

__

SCENE 10
THE PROFESSIONAL GAME
Time: 1 minute
Camera: 360º
The kid goes to a big game (FC Porto) as the ball boy. He is next to the pitch. We can see his viewpoint, we feel his enthusiasm and admiration for the players and for the game.
[**TV**: user watches the game in the kid's point of view.]
[**HMD**: user watches the game in the kid's point of view. Portal with optional info on the players that pass by. Can pause this information.]
[**Tablet:** user watches the game in the kid's point of view. Can access statistics from the previous games between those two teams.]

Suggestion for additional scene: user can see the match at different places: from balcony, next to the pitch, behind the goal, at the bench. he can see/choose replays, the list of the team players that are on the game, etc.

END CREDITS
End credits appear graphically with extra footage of the making of.
[**TV**: user watches end credits and windows with footage of the making-of.]
[**HMD**: Watches the kids playing outside with end credits above and a portal showing the making-of.]
[**Tablet:** Watches the kids playing outside with end credits above. Links for the companies who created the documentary and video of the making-of.]

3. PRODUCTION SCENARIOS

3.1. Production preparation

The specific scenario defined in 2.3 requires a pre-production workflow that loosely follows the following user scenario:

User scenario: The media department of a broadcast company has commissioned a VR documentary on the dream of a young football kid David, who joined the Dragon Force, FC Porto football school. A producer (Luis) has received the order to make a documentary as engaging as possible that can be experienced simultaneously and synchronously on multiple devices in a living room, including tv, tablets and head mounted displays, leaving the end user free to choose the viewing experience he or she wants. In order to fulfil the request of his employer, Luis faces multiple challenges of both creative, technical (workflow) and financial nature:

- Optimal storytelling on each of the targeted devices, while at the same time provide a coherent experience across these devices
- Prepare, capture and produce for TV and HMD simultaneously: make it possible and do it efficiently
- Make the VR experience as immersive, attractive and interactive as possible, in order it adds to the TV experience instead of breaking it
- Make a high quality production within the tight budget constraints

Current VR production tools fall short to cope with the challenges above. Luis needs an innovative platform that address both creative and technical needs. We now outline in further detail how this process is done by further expanding the user story.

3.1.1. Prepare the story

User scenario1: In order to prepare his program, Luis starts the production preparation application in order to prepare the new documentary. After entering the usual technical and management details and metadata about the program and the production crew, the real work can start. First, he **defines the main storyline** and the main events, actions and story elements (story beats) of the story. Luis wants to follow the kid at home, on the road, on the field, in school, … Next, he **defines the main and side characters** to help guide the interactive aspects of the experience: David's family, friends, trainer and teacher. He then **works out the story structure in more detail**, by elaborating the main storyline into well-described scenes, including the **definition of sub-storylines for TV, tablet and HMD**, as well as alternative plots based on the action of the end-user. Now, it's time to also take care of the envisioned end user scenarios, more specifically to **define the interaction design** and determine the intended user journeys through the story. At this point, everything is in place to dive into the details. The outcome of this process is a blueprint of the story format, which Luis can **save as a master template representing the format script containing all relevant metadata** to be used, changed or extended in subsequent steps in the production flow.

The preparation of an audiovisual production typically requires writing a script and use it as a basis to refine it iteratively, adding details such as shooting locations, description of characters, etc. This kind of preparation is generally done in documents such as the script, the shooting breakdown and other production documents commonly used in normal omnidirectional and traditional shootings. In a normal production, this can be done with generic office software (word, excel…) with a list of footage for locations, actors, etc.

The production scenarios of pilot 1 do not require additional tools, and we therefore do not foresee specific software development to realize the concept of ImmersiaTV. It might be necessary though, to reconsider this assertion depending on the choice of production strategy (see section **¡Error! No se encuentra el origen de la referencia.** in Section 2.1), particularly when different actions are possible by the end-user and we always want to preserve a consistent universe, despite the different narrative paths adopted by end-users.

Keeping in mind this last scenario, we can outline a list of software requirements (SR) for such a content ideation and pre-production software.

Preparing the story

SR.1.1 The content creator can create the main storyline

SR.1.2 The content creator can define the main and side characters

SR.1.3 The content creator can define the detailed story structure

SR.1.4 The content creator can define the sub-storylines for TV, tablet and HMD

SR.1.5 The content creator can define the user interaction design

(For each scene:)

SR.1.6 The content creator can define the multi-platform logic

SR.1.7 The content creator can define the viewer perspective(s)

SR.1.8 The content creator can define the detailed script

SR.1.9 The content creator can define if it uses Omnidirectional and/or directive content

SR.1.10 The content creator can define the viewing angle

SR.1.11 The content creator can define the interaction points: portal, AR object, caption, graphics, …

SR.1.12 The content creator can define transition between scenes

SR.1.13 The content creator can specify use of audio for guidance and transitions

SR.1.14 The content creator can indicate use of camera movements

SR.1.15 The content creator can indicate forced exploration mode where applicable

SR.1.16 The content creator can save resulting format script as master template

3.1.2. Prepare the production

User scenario2: While preparing the story, Luis asks Eva to do some research on location and to investigate potential scenes and actors. Eva takes a low end VR capturing device with her, and captures some raw 360° photo and video material to visualize potential shots and scenes. A smartphone app enables her to record some directive footage as well, and to **add all materials as placeholders in the format script**. The materials are instantly verified by Luis, who can **perform a first VR preview** based on the raw material if he wants to. This **preview is able to automatically take the story script format into account if the content is available.** Luis can ask Eva for additional try-outs with occasional actors via **in-app direct messaging**.

Luis is now ready to **define the shooting plan**: he selects the used equipment in function of the scenes and determines the setup of the different cameras and microphones. The relevant data of the shooting plan is also saved as production metadata that remains available throughout the production workflow.

A particular bottleneck he will have to contemplate and solve is **how he can combine omnidirectional and directive capturing whenever it is needed for the story**.

This user scenario can be translated in the following requirements:

Preparing the production

SR.2.1 The content creator can add on-location research material as placeholder in format script

SR.2.2 The content creator can perform first VR preview of relevant scenes

SR.2.3 The content creator can define the shooting plan

SR.2.4 The content creator can define the VR/directive capturing strategy

At this stage, given the fact that all these tasks seem achievable with standard software in the context of pilot 1, we do not proceed to further analysis of these requirements, and this module will not be integrated in the first version of the architecture.

We will, however, revisit such an assumption for pilots 2 and 3, in case it reveals itself necessary.

3.2. Editing and Post-production

By post-production we define any process that has to be done after the shooting. This includes stitching, synchronisation, editing, compositing and publication of multi-platform content.

The ImmersiaTV project addresses omnidirectional content production in a multiplatform environment, and as such it covers several kinds of user experience involving different kinds of hardware. This section outlines in further detail an optimal content production process that delivers such experiences in a smooth and intuitive way. At this stage, we only describe the case of pilot 1 in which the broadcast mode applies: this means that the three devices show synchronized content. In further iterations (pilot 3), the production workflow will be enriched with the possibility of creating an exploration mode, where different end-users can trigger different paths in the experience.

In ImmersiaTV, interaction is defined inside a scene typology. A scene typology is a combination of distribution and interaction mechanisms that allow synchronizing different audio and video streams, both within and between devices, and enable interaction between them, both within a certain device and between devices. The scene typology is therefore implemented across several players running simultaneously in different devices and communicating between them.

One of the challenges of creating multi-platform content is to define a content format that can, simultaneously, be adapted to the specificities of each platform and, at the same time, work across devices, allowing the user to do seamless transitions between devices if they want to (see section 2.2 for a comprehensive list of options). For multiplatform omnidirectional video experiences, in the worst case scenario, the content creator will have to edit 3 different video sequences, and do so very carefully in order the timing of the three videos is consistent across platforms. This can be a tedious, cumbersome and unintuitive process. In ImmersiaTV, we aim to find ways to efficiently address combined TV and VR content production scenarios by optimizing synergies and cross-connections between these two typically hardly separated flows in today's practices.

The requirement analysis of the post-production process has been done thoroughly in tasks T2.2 and T2.3 to make sure that, on the one hand, no creative limitations are introduced by the developed tooling and that, on the other side, the content of omnidirectional and synchronized directive content can be produced without major technical burden, as long as the content creator makes use of a series of plugins, libraries and materials provided by the ImmersiaTV consortium (See also Figure 3).

Consistently with previous sections, we outline a user scenario to specify the requirements of the ImmersiaTV post-production process.

User scenario3: After the shooting is completed, Luis **visualizes the raw material across the different end-user devices**. He checks some **tutorials** to find out how to **stitch and synchronize** the videos recorded by the different cameras in the omnidirectional camera rig, as well as the directive cameras used to have directive shots that are synchronized with the omnidirectional ones.

Once this step is completed, he **imports everything in a standard edition software**. To reconcile the needs of having scenes with an output in the form of 360° videos (aspect ratio 2:1) and scenes using traditional shots (aspect ratio 16:9), he creates **two sequences (one 16:9 and one 2:1) embedded in a container scene**. This allows him to edit for TV and for omnidirectional video keeping a close eye on the timings of both projects.

He is able to **edit the whole omnidirectional sequence directly in his editing software thanks to a set of predefined video transitions**, which allow transitioning between omnidirectional video scenes easily, without needing advanced compositing techniques. However, at some point he wants to customize one of these transitions in a more **advanced mode: in a compositing software**, he opens the video transition, edits the omnidirectional video projected into a sphere, re-exports it in a format understood by his editing software, and completes the edition of the scene within his video editing software.

After this, he wants to introduce interactivity: he introduces **conditional transitions between shots and scenes**, defines the condition on which the transition is triggered and the impact it has: as a result, when the HMD user looks into a certain 16:9 video portal for more than 2 seconds, the video becomes bigger. Using the same method, **he defines a different scene typology,** this time **involving different devices**: in the final part of the audiovisual experience, the end-user will see 12 videos organised in a matrix in his tablet. When he will click on one, the TV will show that content. To create this scene typology, he organises the layout of the different videos, indicates that the scene is for the tablet, and introduces, for each of the different videos, a different conditional transition, each of which is pointing to a different scene within the 16:9 content being edited. He also tests whether clicking on each of the videos with a conditional transition triggers the appropriate TV content.

Finally, **he selects what scenes are for the TV, the tablet and the HMD**, clicks **a special export button, and obtains a set of videos and metadata that is ready for broadcast**. He checks that the content is accessible with his different devices and run synchronously, and goes home happily having completed a multi-platform content production much more easily than what he would have imagined.

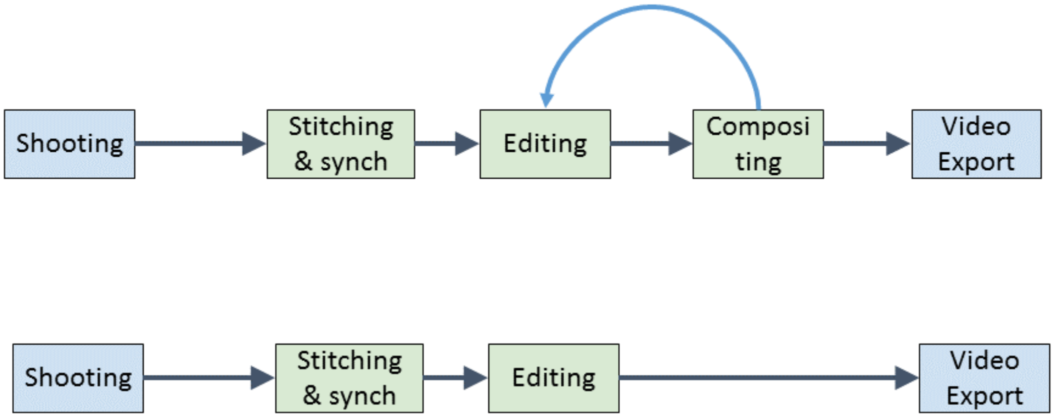

Figure 3: ImmersiaTV post-production workflow. On top, the default process of omnidirectional video post-production. This involves a significant amount of compositing, i.e., advanced post-production At the bottom, a workflow closer to a traditional content editing process allowing for much faster content creation, but with more limited creative options. Ideally, the post-production pipeline of ImmersiaTV should allow for both.

3.2.1. Editing and Compositing requirements

We can define an editing tool that allows editing a multi-platform omnidirectional video project in one go for the different platforms. This editing tool would define multi-platform content as a series of editing constraints, in a way similar to how in a modern video editor audio and video tracks are by default coupled (in Adobe Premiere's terminology, these are called synchronized tracks). The constraints imposed by such editing tool would allow automating a process similar to what computer scientists call *multi-platform compilation*: generate all the video formats needed for the different platforms automatically from the same project. The challenge of designing this plugin is to specify the extra information needed for this multi-platform content to be delivered synchronously, and allowing for the definition of interaction mechanisms, without imposing any constraint in terms of storytelling for the different targeted platforms. Following the user scenario previously introduced, we can identify the following requirements for the post-production tools:

SR.3.1 The content creator can visualize the raw material across the different end-user devices.

SR.3.2 The content creator can use a standard edition software (Adobe Premiere, Final Cut, or other), e.g. by using predefined transitions, and avoid, for simple projects, using advanced compositing software.

SR.3.3 In the editor software, the content creator can edit content for TV and for omnidirectional video in such a way that the timings of the content for the different targeted devices is continuously visible.

SR.3.4 The content creator can use Windows and OS X.

SR.3.5 The content creator can make use of an advanced mode in a compositing software (Nuke, Adobe After Effects)

> SR.3.6 The content creator can introduce interactivity within the editor timeline through *conditional transitions* between shots and scenes
>
> SR.3.7 The content creator can select, within the editor timeline, which video assets are visible within the TV, the tablet and the HMD
>
> SR.3.8 The content creator can also create *ImmersiaTV scene typologies*, i.e., interaction between devices, through *conditional transitions* within the editor timeline

An ImmersiaTV scene typology (see Section 2.2) is a set of interaction mechanisms, both within and between devices. Such scene typologies will have to be defined in a metadata scheme whose syntax allows connecting input from the content edition process and dynamic mechanisms within the multi-platform player used by end-users across devices. Specific widgets in the editing software will allow defining how the different clips should look in the final production (a rectangular portal, an omnidirectional video projected on a sphere, integrated in a more complex mesh, etc.) and behave (where the portals should appear, whether they should be placed relative to the user's head orientation or relative to the scene, whether they should trigger additional video materials or other kinds of assets, etc.).

Since the main requirement of pilot 1 is to deliver a synchronous experience across devices, we can foresee a general requirement that has an impact across the whole post-production process

> SR.3.9 In pilot 1, the end user will experience the content with a common timing across devices (HMD, TV, tablet), it will be continuous and have no time jumps

A further analysis of requirement 3.2.3, i.e. to allow using both standard editing and advanced user software, has prompted a further refinement of the requirements, taking into account the following considerations:

- Regarding the specifications of transitions and portals based on omnidirectional content, the simplest approach seems to be to do it with black and white video MATTE transitions. A transition in a black and white video MATTE is a way to define transitions between different content material. The software reads the black video as the part of the video that is invisible and white for the part that is visible (see Figure 5).
- In order for the transition to work, one needs to prepare the first shot, or background shot, and the shot that follows it, and to which it transitions to. The transition effect is also needed. The shots and the transition need to be in different video tracks, as the transition effect uses the track to determine when the transition takes place in the timeline.

In Figure 4, in the VIDEO 03 track (henceforth referred to as V3 for brevity purposes), the black and white transition is placed. Next, in the V2 track, one places the shot that it transitions to, that is, the resulting shot. Finally, in the V1 track, our background shot. It is at the point that the three shots intersect in the timeline that the transition will occur; if there is nothing underneath the transition, for example, in V1, the transition will not work, just show part of it or nothing at all. This model can be applied to any type of transition footage, as long as it is presented in black and white. Therefore, from a content edition perspective, it seems to facilitate greatly the task to specify that:

> SR.3.10 The content editor, using either a classic video editor or an advanced one, will easily define transitions and interactive transitions between omnidirectional videos using black and white video matte.

Finally, workshops with content editors to clarify this option have shown that it is necessary to:

> SR.3.11 The content editor will be able to add a beauty layer to the interactive transition. This beauty layer will unfold synchronously with the video MATTE. It will be used to add borders and eventually other visual content needed for the transition. (see also Figure 4)

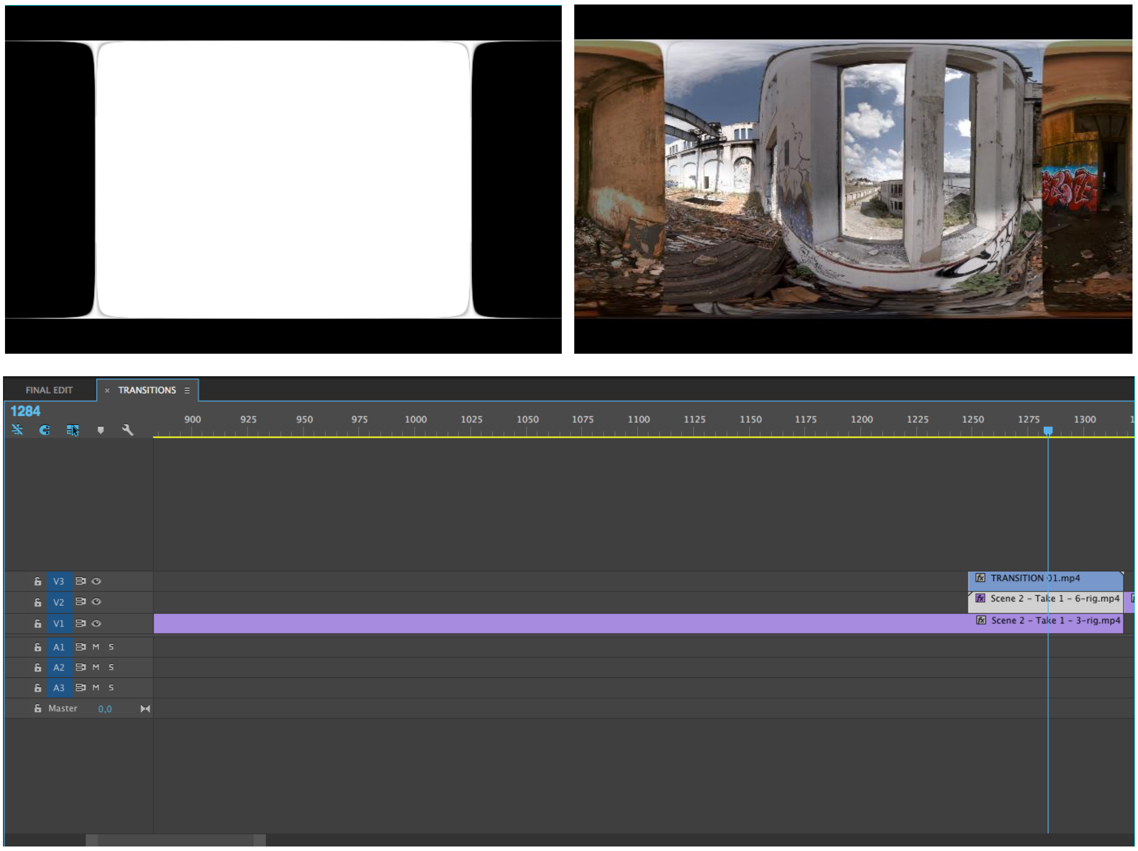

Figure 4: Interactive transitions between omnidirectional videos using matte. On top left, a frame of a black and white video matte video. On top right, the result of applying such mask on two omnidirectional videos. On bottom, the edition timeline corresponding to the output shown on top.

Interactive transitions are particularly interesting for the first pilot of ImmersiaTV because, despite all content will be synchronized in time, it is possible for transitions and portals to show some level of interaction. For example, by expanding a text when clicking on a particular item on a tablet, or by looking steadily at a certain portal in an HMD. Another advantage of using black and white video matte is that it is also possible to visualize different interaction scenarios within the editor. Concretely, part of the transition can be frozen to show only a portion of the following shot. For example, in Adobe Premiere, using the playhead on the timeline, right clicking the shot and choosing the ADD FRAME HOLD option, one can freeze part of the transition that, otherwise, would have a set time of unfolding. That portion can be of any duration, followed by the remaining transition footage, so that it can complete. Similarly, a custom filter for ImmersiaTV could create content that freezes or makes the black and white video matte complete based on the interactive input of the user. It also has the advantage that such transitions can be created beforehand, giving a set of pre-defined options to the editor, and avoiding the need, for simple projects, to use advance compositing software.

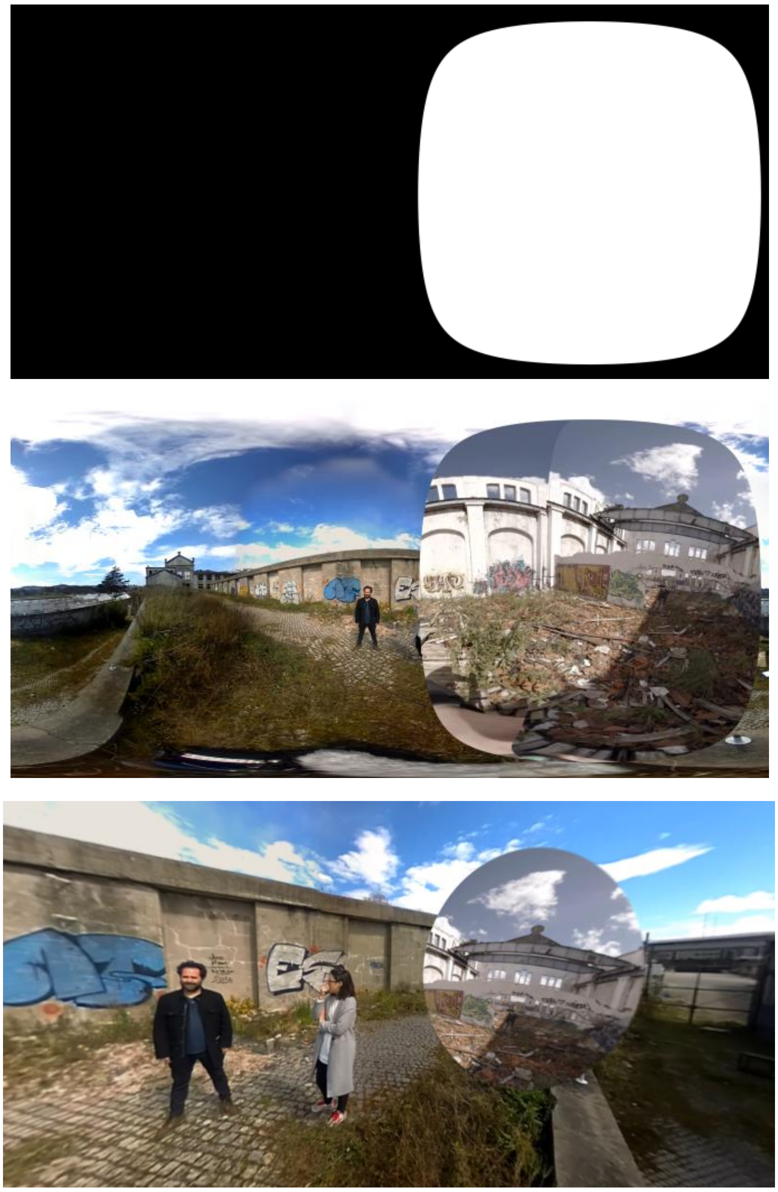

Figure 5: Inserting portals by applying video masks in video. On Top, Black-and-White Image Generated in a compositing software (Adobe After Effects) with a third party plugin (Mettle's Skybox), in equirectangular projection. In the middle, the black and white video matte applied to two omnidirectional videos in a standard edition software (Adobe Premiere). At the bottom, a portion of a video frame with the same black and white video matte applied, without the equirectangular projection.

3.2.2. Pre-visualisation requirements

Adapting the traditional workflow of a video editor to omnidirectional and multiplatform content creation imposes specific visualisation requirements. Concretely:

SR.3.12 The content creator will be able to see omnidirectional content both in projected and non-projected views by using Previsualisation tools integrated in the content editor (see Figure 5)

Within the paradigm of ImmersiaTV, it is also necessary to ensure that:

SR.3.13 The content creator will be able to pre-visualize transitions and interactive transitions within the editing software

SR.3.14 The content creator will be able to pre-visualize synchronized playout between different devices, for example, to see how TV and HMD content fit in timing.

3.2.3. Content publication

This section only applies to the offline scenario. To prevent human errors, the videos and metadata resulting from the edition process should be generated automatically by a plugin within the post-production software.

The exporting of files needs to support several cases. If we consider a transition from one video to another, the following scenarios are possible:

- Both video are omnidirectional and referenced to the world (i.e. a portal containing the new video does not "travel" when the user looks around). In such case, the new omnidirectional video can be exported already blended with the video output.
- New video is referenced to the end-user perspective. This implies a separate black and white video matte and content video will have to be generated and sent to the broadcast server
- The new video is interactively rendered depending on the behaviour of the user. In this case, even if it is world-referenced, a separate black and white video matte and content video have to be generated and sent to the broadcast server

Therefore, regarding the generation of all the files to allow the delivery of the content, the content creator will have:

SR.3.15 An *export button* will generate a set of videos and metadata that is ready to distribute content across devices. The video exporter needs to have the following specificities:

SR.3.16 The export functionality will accept sequences involving different aspect ratios, due to differences in omnidirectional and traditional video formats (most likely solved through *nested sequences*).

SR.3.17 The common cutting points between devices will be visualized putting the content for the different devices in two sequences, one on top of the other.

SR.3.18 It will be possible to define a label specifying the destination for each sequence

SR.3.19 The outcome should be:

1) A set of videos in the highest resolution possible. The videos should have a shared timestamp. This means that the timestamp introduced at the frame level is common to all the different video streams. For example, the

first frame of a video introduced exactly at second 12 of the broadcast should have its first frame with a timestamp set at 00:00:12:00.

2) A metadata file detailing how the different videos have to be organised to compose an omnidirectional scene. This file should be compatible with broadband distribution standards.

3.3. Content playback

The user scenario introduced in section 3.2 assumes that it is possible to visualize the raw and the edited content within the different devices (HMD, tablet, TV). This requires a specific application, made available both to the content creator and to the content consumer.

User scenario 4: To get a better acquaintance of the materials captured, Luis visualizes seamlessly **the raw material across the different end-user devices**. Later on, once the edition is completed, he can also visualize the final edited content in the end-user devices, and test whether the interaction mechanisms –within and between devices- work as expected. Satisfied, he makes the content available online.

Finally, a mass of people download the ImmersiaTV app to their home devices, and are capable of consuming the content created by Luis, as well as interact within the content created.

We now define the major requirements of the application that will display and provide the immersive experience to the final client. For now, the requirements are oriented to implement the immersive experiences focusing on pilot 1 objectives.

We therefore do not introduce detailed specifications such as the look and feel of the application, which specific menus and options will be available, etc. These requirements will arise during the preparation of pilot 1 and they will be used to improve pilot 1 and the subsequent pilots 2 and 3.

At this stage all the requirements described below are the critical items required to enable the immersive experience expected for pilot 1 in this project, as it is described in the Description of Work (DoW) document, relevant parts of which are included in section 1.2. In the following subsections the major functional requirements are listed and elaborated.

3.3.1. Elemental multimedia player features

The elemental and traditional controls for a multimedia player are the classic buttons of **Fast Forward**, **Rewind**, **Play** & **Pause** and **Stop**. More over an API, menu or dialog has to be provided to set the multimedia source (a URL pointing to specific OTT content, a file in the local filesystem, etc.). The player of this project does not have a specific need of Fast Forward, Rewind and Pause, at least at this stage of the project.

Note that these controls present a different interface or behaviour depending on the device that is playing the content. There are three different kinds of devices or screens in the project scope: TV sets, tablets or smartphones and HMDs. The mechanism of interaction implemented may be different in all targeted devices (touch screen, head movement, TV remote).

In addition, smartphones can act or not as an HMD. Therefore, another requirement is to have the possibility to use the mobile device as a HMD or smartphone/tablet in the smartphone or tablet app.

Summarizing, we can list the following requirements:

SR.4.1 Basic controls. The basic controls of the player will be:

- **Select media source:** which is likely to be a list of available content, located in public servers.
- **Play**: Starts to process the selected source.
- **Stop**: Stops the current reproduction and allows you to select a content once again.
- **Select tablet or HMD mode**: switch from tablet to HMD behaviour and rendering.

Finally another feature which is not mandatory but interesting to consider for the software design is to allow the possibility of playing standard content (without metadata provided by custom ImmersiaTV descriptors). This will help to design more generic and reusable software out of the project.

3.3.2. ImmersiaTV Scene and Interaction management

According to the DoW the immersive content will have to be rendered in a virtual environment where it must be possible to include objects (i.e. portals) at any place of the virtual world. Practically this means that the player will have to be capable to recreate a 3D spherical virtual environment to display 360° videos, but also place objects at certain arbitrary positions. In addition, part of these elements will have to react interactively to the behaviour of the user.

The implications of such scene management requirements are large: the player will have to deal with 3D environments which are not part of the omnidirectional recording, and will allow having multiple omnidirectional or directive videos that are composed and rendered simultaneously in a single scene.

Therefore, the player has the following requirements:

SR.4.2 Metadata to describe and define the scene: The scene composition information has to be distributed to the player. This includes information like which videos are visible and where they placed or how video scenes are composed. This metadata may be transmitted in a multiplex or signalled within the stream itself, or it might be transmitted using a parallel communication channel.

Given the fact that the project will cover live use cases in future pilot iterations, we foresee a preference to transmit metadata within the stream, this would facilitate dynamic scene changes and the corresponding synchronization.

SR.4.3 The scene is device dependent. Each type of device will have to render a different scene, as the interaction with the user will be different. This implies **there is a scene description for each type of device.**

SR.4.4 Render multimedia content over textures and 3D objects**.** One or several videos will be displayed in different positions over the 3D scene (over a spherical surface, as a regular 360° video, or over plain surface in a portal-alike effect).

SR.4.5 Apply video masks in videos. A mask is needed to overlay more than one video over the same texture forming an overlay of an arbitrary shape (i.e. to render a portal as a circle over the 360° sphere). (see Figure 5)

SR.4.6 Interaction management. The player needs to be able to process a systematic way to define interaction mechanisms in the end-user devices, and the methods implementing such interaction mechanisms need to be made available to the content creator.

At this stage of the project the user interface to interact cannot be completely defined how t but the architecture design needs to allow for several end-user interfaces adapted to each device (smartTV, tablet and HMD).

In tablets the user interface may include active regions of the scene, so the user can touch and activate actions by interacting only by using the touch screen. In HMD the end-user input will be pretty different; for instance, a change in the scene may be activated by looking to a specific region during a certain duration of time. Finally the smartTV will require using the remote control (which is difficult as the industry is fragmented and not properly standardized) or using a remote API (like in second screen applications).

3.3.3. Multitrack support

The player will have to be capable to support multi-tracks. As already noted in previous chapter it is necessary to be capable of decoding and rendering more than one video stream simultaneously (portals, map scene, second screen cases, etc.). This applies to all devices as second screen scenes, map scenes or any other type of point and click scene is feasible in HMD, tablets and in TV sets (even within a more limited interactivity mode).

Multi-track in such immersive and cross device content experiences implies having to decode in sync several streams, which may or may not refer to all of the available tracks. In fact, the player must provide the possibility to decode only the specific streams which are relevant for the current scene, so according to the scene there will be a logic that specifies or selects the appropriate tracks.

3.3.4. Synchronization across devices

As it has been stated from the project conception, the project aims to provide a coherent experience across devices. As it is known in second screen applications or in content provided in multiple platforms, a common issue is how to synchronize the delivery and playback of the same or strictly related content in multiple devices and platforms. ImmersiaTV aims to achieve a very precise level of synchronization, up to a frame level. Moreover as the set of available devices is not static, it cannot be assumed that the devices are all going to always the same and always present: there can be a single HMD, none or even several HMDs in a single session; and the same accounts for TVs or tablets. Also devices can be turned on and off during the content play-out.

Synchronization requirements are:

> SR.4.7 Achieve a frame level precise synchronisation: This is relevant as devices can display different omnidirectional and directional contents that were shot together, so any sort of desynchronization is going to be noticeable by the user.
>
> SR.4.8 The devices may need to synchronize to any base media time at start up: A device can be turned on when there is already the reproduction going on in another device, so the one joining the group must get synchronized without affecting the other ongoing playbacks.

3.3.5. Audio configuration

There are no specific requirement about the audio playback, however, given the fact that several devices can reproduce the content simultaneously, there must be some sort of audio control. It doesn't make much sense that the tablet and the TV play audio simultaneously without using headphones, and depending on the HMD device, this is also applicable, as not all HMD have incorporated headphones.

At this stage of the project, and awaiting the feedback of future proofs of concept and further pilot iterations, there is only a single requirement: enable or disable audio playback. It can be achieved by some on/off feature that is developed specifically or it can be done by using the platform volume control. The fact is that it must be taken into account during the player design, to ensure it is going to be possible and easy to turn on and off audio playback.

SR.4.9 Basic audio control in the end-user devices

3.3.6. Second screen functionality

Beyond the scope of pilot 1, one of the possibilities in the area of adding interactivity in the user experience can be related to adding multi-camera selection for the TV set. Actually this is like adding a multi-camera second screen application between the tablet and the TV. The idea is to display all possible streams in a mosaic in the tablet, so the user can switch the camera that is shown on TV by selecting it on the tablet. In second screen technologies this is a known feature with clearly identified issues or challenges, which basically rely on synchronizing the tablet and the TV, establishing a communication channel in real-time between the TV and tablet; and, finally, switching from one stream to another fast and smoothly enough to not disturb the user.

In the context of ImmersiaTV this functionally needs to be further defined and conceptualized in the following stages of the project. However as it is understood up to now this implies the following requirements:

SR.4.10 Real time communication channel between devices: sending messages from one device to another

SR.4.11 Second screen scene definition: The definition of the second screen view (mosaic layout) in the tablet must be defined within the content production process.

There seem to be two approaches:

- Include in the player a mosaic mode, which means that there is a relevant part of the code coupled to that feature. On the other side this approach does not have any impact on post-production flow.
- Include in the post-production metadata the description of the second screen view for the tablet, as any other VR scene for the HMD or tablet. This approach implies that the metadata include all the needed information to perform such a functionality. The metadata is produced during the post-production, so it has impact on the post-production work flow.

3.3.7. Screen cast

Beyond the scope of pilot 1, another feature that can be considered for the player, especially regarding the usage of the HMD, is to cast the stream rendered by the HMD to another screen (tablet, TV or even HMD).

Being able to share screen casts, both for head-mounted displays and for tablets, is a desirable functionality not only in a home environment, but also for social media integration: the end-user would be able to capture attractive parts of his experience and share it through social media. Summarizing, the following requirement can be introduced:

SR.4.12 The end-user can capture screen casts and share them with other devices

SR.4.13 The end-user can capture screen casts and share them through social media

4. USE CASE AND REQUIREMENT ANALYSIS

From the analysis of the user scenarios in previous sections, as well as the general requirements of D2.1 and D2.2, we have identified 4 general user scenarios (US), each with its specific software requirements (SR).

US1. Story preparation

SR.1.1 The content creator can create the main storyline

SR.1.2 The content creator can define the main and side characters

SR.1.3 The content creator can define the detailed story structure

SR.1.4 The content creator can define the sub-storylines for TV, tablet and HMD

SR.1.5 The content creator can define the user interaction design

(For each scene:)

SR.1.6 The content creator can define the multi-platform logic

SR.1.7 The content creator can define the viewer perspective(s)

SR.1.8 The content creator can define the detailed script

SR.1.9 The content creator can define if it uses Omnidirectional and/or directive content

SR.1.10 The content creator can define the viewing angle

SR.1.11 The content creator can define the interaction points: portal, AR object, caption, graphics, …

SR.1.12 The content creator can define transition between scenes

SR.1.13 The content creator can specify use of audio for guidance and transitions

SR.1.14 The content creator can indicate use of camera movements

SR.1.15 The content creator can indicate forced exploration mode where applicable

SR.1.16 The content creator can save resulting format script as master template

US2. Production preparation

SR.2.1 The content creator can add on-location research material as placeholder in format script

SR.2.2 The content creator can perform first VR preview of relevant scenes

SR.2.3 The content creator can define the shooting plan

SR.2.4 The content creator can define the VR/directive capturing strategy

US3. Edition and Compositing

SR.3.1 The content creator can visualize the raw material across the different end-user devices

SR.3.2 The content creator can use a standard edition software (Adobe Premiere, Final Cut, or other), and avoid, for simple projects, using advance compositing software

SR.3.3 In the editor software, the content creator can edit content for TV and for omnidirectional video in such a way that the timings of the content for the 2 targeted devices is visible constantly

SR.3.4 The content creator can use Windows and OS X

SR.3.5 The content creator can use of an advanced mode in a compositing software (Nuke, Adobe After Effects)

SR.3.6 The content creator can introduce interactivity within the editor timeline through *conditional transitions* between shots and scenes

SR.3.7 The content creator can select, within the editor timeline, which video assets are visible within the TV, the tablet and the HMD

SR.3.8 The content creator can also create *ImmersiaTV scene typologies*, i.e., interaction between devices, through *conditional transitions* within the editor timeline

SR.3.9 In pilot 1, the end user will experience the content with a common timing between devices (HMD, TV, tablet), it will be continuous and have no jumps

SR.3.10 The content editor, using either a classic video editor or an advanced one, will easily define transitions between omnidirectional videos using black and white video MATTE.

SR.3.11 The content editor will be able to add a *beauty layer* to the interactive transition which, unfolding synchronously with the black and white video matte, will add borders and eventually other visual content needed for the transition

SR.3.12 The content creator will allow seeing omnidirectional content both in projected and non-projected views by using Previsualisation tools integrated in the content editor.

SR.3.13 The content creator will be able to visualize transitions and interactive transitions will be visible within the editing software

SR.3.14 The content creator will be able to visualize synchronized playout between 2 devices, for example, to see how TV and HMD content fit in timing.

SR.3.15 An *export button* will generate a set of videos and metadata that is ready to distribute content across devices. The video exporter will have several specificities:

> SR.3.16 The export functionality will accept sequences involving different aspect ratios, due to differences in omnidirectional and traditional video formats (most likely solved through nested sequences).
>
> SR.3.17 The common cutting points between devices will be visualized putting the content for the different devices in 2 sequences, one on top of one another.

SR.3.18 It will be possible to define a label specifying the destination for each sequence

SR.3.19 The outcome should be:

1) A set of videos in the highest resolution possible. The videos should have a shared timestamp. This means that the timestamp introduced at the frame level is common to all the different fluxes. For example, the first frame of a video introduced exactly at second 12 of the broadcast should have its first frame with a timestamp set at 12.

2) A metadata file detailing how the different videos have to be organised to compose an omnidirectional scene. This file should be compatible with broadband distribution standards.

The previous use case also assumes there is a multi-platform player available, the main requirements of which we list below:

US4. Content Playback

SR4.1 Basic controls. The basic controls of the player will be:

- Select media source: which is likely to be a list of available content, located in public servers.
- Play: Starts to process the selected source.
- Stop: Stops the current reproduction and allows you to select a content once again.
- Select tablet or HMD mode: switch from tablet to HMD behaviour and rendering.

SR.4.2 The player will process metadata to describe and define the scene: The information regarding how the scene is composed must be distributed to the player. It must include information like which videos are visible and where are they placed or how are they composed. This data may be transmitted muxed or signalized within the stream itself, or it might be transmitted using a parallel communication channel.

SR.4.3 The scene is device dependent. Each type of device will have to render a different scene, as the interaction with the user will be different. This implies there is a scene description for each device.

SR.4.4 Render multimedia content over textures and 3D objects. One or several videos will be displayed in different positions over the 3D scene (over a spherical surface, as a regular 360° video, or over plain surface in a mirror or portal like effect).

SR.4.5 Apply video masks in videos. A mask is needed to overlay more than on video over the same texture forming an overlay of an arbitrary shape (i.e. to render a portal as a circle over the 360° sphere).

SR.4.6 Interaction management. There needs to be a systematic way to define interaction mechanisms in the end-user devices, and the methods implementing such interaction mechanisms need to be made available to the content creator.

SR.4.7 Achieve a frame level precision: This is relevant as devices can display different omnidirectional and directional contents that were shot together, so any sort of desynchronization is going to be noticeable by the user.

SR.4.8 The devices may need to synchronize to any base media time at start up: A device can be turned on when there is already the reproduction going on in another device, so the one joining the group must get synchronized without affecting the other ongoing reproductions.

SR.4.9 Basic audio control in the end-user devices

SR.4.10 Real time communication channel between devices: It will be needed to send messages from one device to another

SR.4.11 Second screen scene definition: The definition of the second screen view (mosaic layout) in the tablet must be defined within the content production process.

SR.4.12 The end-user can capture screen casts and share them with other devices

SR.4.13 The end-user can capture screen casts and share them through social media

5. ANNEX I - PRELIMINARY REQUIREMENTS FOR LIVE PRODUCTION

The customization of the production process, as defined particularly in section **¡Error! No se encuentra el origen de la referencia.**, has also to be translated into a live scenario. This section reports on this design translated to live, but does not cover several aspects of Live production (for example, synchronisation between audiovisual streams within the production process).

Therefore, under the following assumptions:

- 1 standard TV production + 1 extra omnidirectional content production
- the omnidirectional content production has access also to ISOs (isolate feed) from the cameras
- the input from all the cameras has synchronized timestamps
- there is some tool to route the different video streams towards the server in charge of content delivery. Optionally, if we do not want to send all fluxes to the delivery server, a specific multiplexer or router to sub-select part of the fluxes that are routed to the delivery server can be controlled remotely by the content production tool

We can already introduce the following requirements to customize the Cinegy Live software to the requirements of Pilot 2 in ImmersiaTV.

General Requirements

The live production of ImmersiaTV will be done through a customization of Cinegy Live. Several customization requirements apply:

- **Editing.** Cinegy Live allows to customize the layout of different previews.
- **Scene composition.** Additionally, the layout of the scene, i.e., a composition of one or several omnidirectional videos and traditional shots needs to be editable in real time. This implies, at least:
  - Cropping directive videos
  - Selecting a portion of an omnidirectional video
  - Assigning the position of the insert within the omnidirectional video
  - Assigning the reference frame of the position (relative to the end-user, or relative to the world displayed)
  - Assigning the size and shape of the insertions

  The ideal and swift way to edit this scene composition features would be to have multi-touch capabilities in Cinegy Live A candidate device would be a multi-touch screen large enough to edit these videos, and precise enough for the multi-touch support.

- **ImmersiaTV metadata**. The composition of the scenes at the reception side will be done based on transmitted metadata specifying the different videos in a given scene, their position, reference frame, etc. Cinegy Live should be able to generate this in real time from the input provided through the GUI.
- **Output streams**. The output of the live production involves several audio and video streams, and metadata. It is important for broadcast purposes that:
  - the metadata, video and audio fluxes are provided separately.
  - the timestamp introduced at the frame level is common to all the different fluxes. This means that the first frame of a video introduced exactly at second 12 of the broadcast should have a timestamp of 12s.

  - The video and audio outputs should be appropriate for a live ingest by a DASH server. The concrete video encoding will be determined by the VRT distribution workflow.

    Todo: clarify the needs to integrate in VRT's distribution workflow

  - The metadata will be integrated dynamically in the DASH manifest by custom software developed in ImmersiaTV

    Todo: clarify the output format of the metadata for the live production

- **Preview in HMD.** It would be desirable to have live preview of the outcome of Cinegy Live for the different devices. The simplest way to do this would be to provide an rtsp stream, or an rtp stream with some file specifying the codec (like a .sdp in rtsp). This would allow a customized version of the ImmersiaTV player to read directly the output of Cinegy Live, and have device-specific preview with minimal delay (not possible through DASH due to the delays involved).
- **Scene typology management.** Special care should be taken to specific scene typologies involved in a live scenario. This means particular patterns of interaction within one device (Tablet, HMD), or between devices (for example, between TV and tablet, or between tablet and HMD, see section 2.2). Since defining these typologies in real-time seems out of the scope of what is feasible within the "mixing table" metaphor, the simplest approach in this scenario seems to define "Macros" or automated configurations that would generate the appropriate metadata for such scene typologies. This would have the advantage of being fast, adapted for real-time usage, at the price of having to define specific automated configurations for specific scene typologies. A way to define, edit and delete such pre-configured scene typologies within Cinegy Live, previous to the Live production, would be desirable.

GUI

The Live producer will have 4 kinds of panels (see also Figure 6):

1. Sources
2. Shapes Preview. This details the sources composing the omnidirectional scene, and in which shapes they fit.
3. World Reference Composition (Preview + Output). This involves a composition of omnidirectional video + portals that are FIXED in the reference frame of the world. These portals appear as overlays windows (picture in picture).
4. Consumer Reference Composition (Preview + Output). This involves a composition of an end-user viewpoint + portals that are FIXED in the reference frame of the End-user. In other terms they follow the head movements of the end-user, always appearing in the field of view.

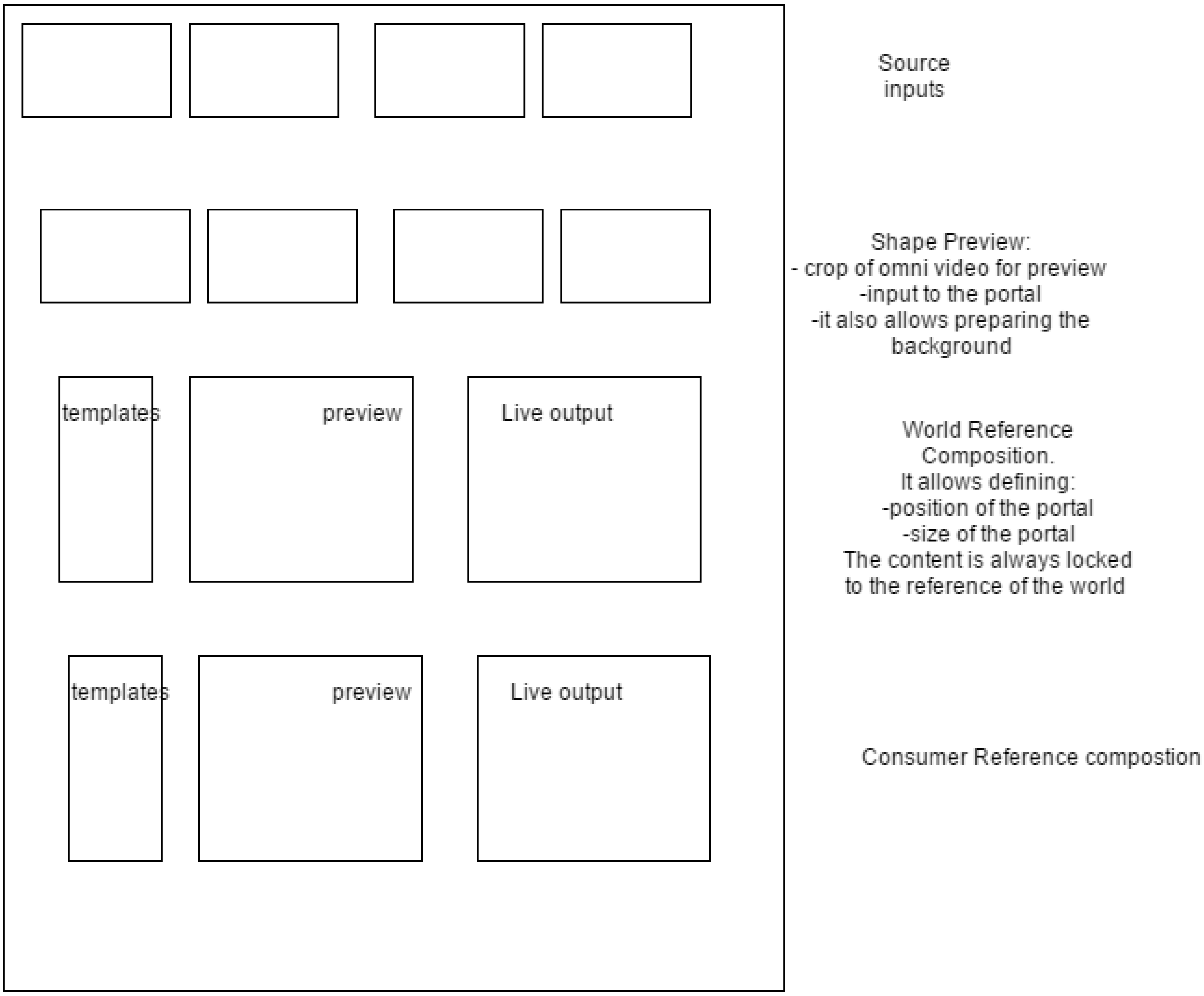
Source inputs
Shape Preview:
- crop of omni video for preview
-input to the portal
-it also allows preparing the background
templates
preview
Live output
World Reference Composition.
It allows defining:
-position of the portal
-size of the portal
The content is always locked to the reference of the world
templates
preview
Live output
Consumer Reference compostion

Figure 6: A mockup of the different panels in the live production tool

1 Sources Panels

These act essentially as a preview of the input.

The omnidirectional channel is visualized in the same form that the live video-stitcher (Vahanna) sends it

It is possible to select the target shape (see panels 2).

2 Shapes panels

In the shapes panels we define what goes into the portals, either directive or omnidirectional videos.

- The source that goes in the shapes is defined by drag and drop
- It is possible to define a cropped section of the omnidirectional video that specifies which part of the omnidirectional video should be visualized in the portal. The shape of the crop can be defined directly on the touch screen. The shape of the crops will be selected from luma masks.
- It will be possible to define luma masks videos which will act as transitions (appearance of the portal).

- Luma masks are predefined in a library. It is possible to add additional luma masks (generated separately).
- It will be possible to adjust the orientation of the luma mask. For example: given a rectangular mask, it is possible to pan and tilt the whole mask in order that the rectangle covers one or another section of the omnidirectional video.
- It is also possible to define an input as the background video. This background video will also have a possible luma mask to define the transition from one background to another background. It will only be possible to add an omnidirectional video to the background
- Directive and omnidirectional video will always maintain their original aspect ratio. Specific luma masks for different aspect ratios will be generated. Consistently, directive inputs will fit with rectangular shapes and, at first stage, no deformations to fit rectangles within spherical projections are contemplated.

<u>3. World reference composition panel</u>

This will consist in defining a set of Spheres and Rectangular Shapes, each with its own video input and luma mask. This composition panel will have 3 sub-panels:

- a list of template compositions.
- creation and preview of the composition.
    - define the position of the different inserts (portals, defined in the shapes panel) through drag and drop. The inserts will ALWAYS be facing the end-user.
    - define the size of the inserts by gesture
    - store a template composition
- the actual output composition. The only thing that can be done in the actual output is add additional graphical material to the composed scene

<u>4. User reference composition panel</u>

This panel also has 3 sub-panels: templates, preview and live output

The functionality is similar to panel 3, the only differences being:

- The overlays follow the end-user movements, the position is not fixed with the background.

Since there is no way of knowing how this will overlay with the videos in panel 3, the background shown is actually a black shape mimicking the viewpoint of the user.