

Deliverable

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

D2.3 Content ideation, production scenarios and requirement analysis

Revision: 1.5

Authors:

Luk Overmeire (VRT)

Delivery date: M24

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 2 of the ImmersiaTV project which aims to create and deliver synchronous experiences across devices (HMD, tablet, TV) for live events based on omnidirectional video. It also comprises a requirement analysis regarding the corresponding technical scenarios and software development for the different functional blocks of the ImmersiaTV end-to-end workflow: live capture & stitching, live production, live encoding and content distribution, content reception & display, and quality of experience. These technical specifications and software requirements form the basis for the architecture, core technical concepts and module implementations, and functional interfaces related to pilot 2, which are defined in Deliverable 3.1.

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
1.0	30/11/2016	Luk Overmeire	VRT	ToC
1.1	19/12/2016	Luk Overmeire	VRT	First version (pilot 2)
1.2	23/12/2016	Luk Overmeire	VRT	Reviewed version
1.3	30/05/2017	Luk Overmeire Gregg Young	VRT	Revisions after mid-term review
1.4	18/01/2018	Luk Overmeire	VRT	First version (pilot 3)
1.5	30/01/2018	Luk Overmeire	VRT	Reviewed version (pilot 3)

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 and 2 of the ImmersiaTV project, as well as a requirement analysis regarding technical specifications and software development needs that have to be considered to create and deliver synchronous experiences based on omnidirectional and directive video for live events across devices (HMD, tablet, TV).

In Section 2, the off-line and live pilot use cases are suggested, based on insights gathered from interviews and brainstorming with professional content creators and end users (D2.1 and D2.2). Key pilot concepts are introduced, followed by a detailed reference pilot content scenario. For pilot 1 in particular, we describe the different shooting options for simultaneous recording of directive and omnidirectional content. For pilot 2, the foreseen ImmersiaTV extension of traditional TV capturing of cyclocross in the pilot set-up is described, and the targeted experience is illustrated and explained based on concrete user interface examples. For off-line pilot 3, we elaborate the new concepts introduced that come with the newly introduced exploration mode. For the live pilot 3, we describe the extensions with regard to live pilot 2.

Based on these pilot content scenarios, we describe in Section 3 the corresponding production and consumption scenarios and user stories for off-line and live immersive content creation in the three ImmersiaTV pilots, approached from both professional and end user perspective. Then we discuss the consequences these user scenarios have in terms of technical specification and software requirements in the context of content creation within the scope of ImmersiaTV (multi-platform content based on omnidirectional video). For off-line documentary production we have detected the need for developing three tools:

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

For pilot 2, a central and essential role is played by the Live Production tool, which is the live equivalent of the off-line edition tool of pilot 1. It takes in incoming (stitched) streams from omnidirectional and directive capturing devices, provides the operator with live editing capabilities, produces metadata streams and synced video streams and sends it to distribution encoder for delivery to consumers. In contrast with pilot 1, Encoding is a real-time inline production processing step, requiring a series of trade-offs and encoding design decisions that affect the quality of experience, which will be assessed by the Quality of Experience module both on server and client side. Finally the ImmersiaTV Player (Reception, Interaction & Display) will be further accommodated to deal with the live scenarios of pilot 2.

For pilot 3, which stages advanced versions of off-line and live ImmersiaTV scenarios, the user scenarios further build upon pilot 1 and 2 adding new functionalities for professional users and more advanced experiences for end users.

Finally, in Section 4, an overview of the technical specifications and software requirements for both off-line and live scenarios are listed. This will be used in Work Package 3 to design the ImmersiaTV software architecture.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Stéphane	Valente	VideoStitch	stephane@video-stitch.com
Philippe	Bekaert	Imec EDM	philippe.bekaert@uhasselt.be
Alexander	Kelembet	Cinegy	kelembet@cinergy.com
David	McNally	EPFL	david.mcnally@epfl.ch
Juan	Nunez	i2CAT	Juan.antonio.nunez@i2cat.net
Maciej	Glowiak	PSNC	mac@man.poznan.pl
Saeed	Mahmoudpour	Imec VUB ETRO	smahmoud@etrovub.be
Szymon	Malewski	PSNC	szymonm@man.poznan.pl
Joao	Lourenço	Lightbox	joao@lightbox.pt

CONTENTS

Revision History	1
Executive summary	3
Contributors	4
List of acronyms	8
1. Introduction	9
1.1. Purpose and scope of this document	9
1.2. Scope of this document	10
1.3. Relation with other ImmersiaTV activities	11
2. Content ideation	13
2.1. Off-line documentary	13
2.1.1. Shooting options	13
2.1.2. Interaction within and between devices	16
2.1.3. Pilot 1 Content production scenario	17
2.1.4. Application of content format requirements in pilot 1	22
2.2. Live event	24
2.2.1. Candidate Pilot Use Cases	24
2.2.2. Key pilot concepts	25
2.2.3. Live pilot content scenario	28
2.2.4. Application of content format requirements in live pilots	33
2.2.4.1 Live pilot 2	33
2.3. Pilot 3 off-line: exploration mode (whodunnit)	34
2.3.1. Candidate Pilot Use Cases	34
2.3.2. Key pilot concepts	34
2.3.3. Pilot 3 content scenario	36
2.3.4. Application of content format requirements in pilot 3 (Exploration Mode)	39
3. User scenarios	40
3.1. Off-line documentary	40
3.1.1. Pre-production	40
3.1.2. Editing and Post-production	42
3.1.3. Content playback	51
3.1.4. Encoding	54
3.2. Live event	55
3.2.1. Production scenario	55
3.2.2. Consumption scenario	56

3.2.3.	Technical specification and requirement analysis	57
3.3.	Refined off-line.....	66
3.3.1.	Production scenario	66
3.3.2.	Consumption scenario	67
3.3.3.	Technical specification and requirement analysis	68
4.	Overview of requirements and specifications	71
4.1.	Off-line documentary	71
4.2.	Live event.....	74
4.3.	Exploration mode	78
5.	Mapping of WP2 requirements	79
5.1.	Off-line documentary (pilot 1)	80
5.2.	Live event (pilot 2 and 3)	81

TABLE OF FIGURES

Figure 1 Illustration requirement creation process in WP2	9
Figure 2: Relation between this deliverable and other WP2 tasks.....	12
Figure 3: Sample of the video from BBC – Attenborough and the Giant Dinosaur	16
Figure 4 - Assuming cameras are placed in a 360° scene (various colours represent different physical background). If the view from camera A is selected, portals to Cam B and Cam C can be placed on top of their real background (specific angle).	27
Figure 5: Camera plan for cyclocross capturing (Hamme, Belgium).....	29
Figure 6: ImmersiaTV HMD experience for live cyclocross	30
Figure 7: Portal showing head of the race (tv) content	31
Figure 8 - Concept of replays in live ImmersiaTV experience.	32
Figure 9 - Configurable portal set-up per omnidirectional scene.....	33
Figure 10 - Wireframe of HMD experience (pilot 3). The magnifying glass represents the user’s point of view.....	36
Figure 11 - HMD view of exploration mode (The Hollow Man)	37
Figure 12 - Tablet view of pilot 3: Trifecta (top left), Case Overview (top right), Profiles (bottom left), Evidence (bottom right)	38
Figure 13: 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.	44
Figure 14: 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.	47
Figure 15: 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.....	49
Figure 16: End-to-end ImmersiaTV workflow for live content	58
Figure 17: Connecting cameras on the cyclocross circuit with the OB-VAN over fibre (Nevion fibre set with break-out boxes).	59
Figure 18 Mapping of WP2 requirements for pilot 1 (off-line documentary).....	80
Figure 19: Mapping of WP2 requirements for pilot 2 (live event).....	81

LIST OF ACRONYMS

Acronym	Description
BNC	Bayonet Neill–Concelman
DOW	Description of work
HD	High Definition
HMD	Head-mounted display
OB	Outside Broadcast
QoE	Quality of Experience
SDI	Serial Digital Interface
VR	Virtual reality
WP	Work Package
XML	eXtensible Markup Language

1. INTRODUCTION

1.1. Purpose and scope of this document

This deliverable addresses the content ideation and production scenarios for pilot 1 and 2 (and to some extent pilot 3) of the ImmersiaTV project, as well as the technical needs and requirement analysis regarding software development needs to create for capturing, live production and delivery of live 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 and technical specifications regarding content production and end-user experience.

The D2.1 and D2.2 insights and requirements, generated based on a very open and human centred design approach, are serving as input for the detailed software requirements and technical specifications regarding content production and end-user experience, as well as the content ideation process, pilot content and production scenarios in D2.3. In Task 3.1, as opposed to the more hybrid approach (generic and ImmersiaTV-specific) in D2.1 and D2.2, the focus is explicitly on the pilot scenarios based on ImmersiaTV concepts.

More specifically, end user experience requirements (D2.1) and professional content format requirements (D2.2), are feeding the content ideation process to define the pilot content scenarios in Task 2.3; while the professional workflow requirements that are relevant within the scope of ImmersiaTV (D2.2) are elaborated in more detail into a list of formal software requirements and technical specifications. These technical requirements are organized according to the different system components (capture, stitching, live production, encoding, distribution, display, quality of experience) of the end-to-end ImmersiaTV workflow, in order to achieve a one-to-one mapping with Task 3.1.

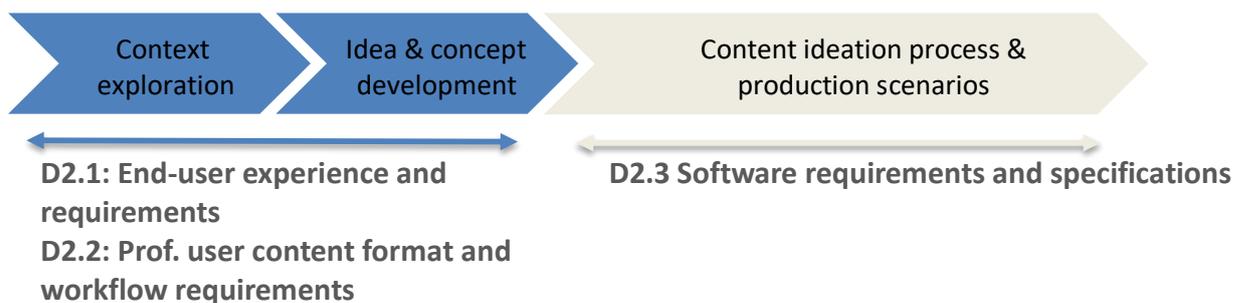


Figure 1 Illustration requirement creation process in WP2

In Section 2.x.4 (pilot 1-3), we provide some example “traces” of how professional content format requirements have rippled through into the pilot scenarios.

In Section 5, a visual overview illustrates how the different WP2 activities are interrelated and how the requirement creation process leads to the D2.3 pilot scenarios and the formal set of requirements.

In D2.3, the requirements are numbered in a different way than for D2.1 and D2.2. In order to have a clear mapping with Task 3.1 (Architecture Design), the requirement numbering includes

the according system component in its name followed by the requirement number for that component, e.g. R-STI-3 (stitching).

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.

The live pilot scenario is introduced and briefly described on page 22:

Pilot 2 (Production led by VRT) *will use the lessons learnt from the first pilot and adapt its offline production pipeline to live scenarios. This pilot will use cyclocross as a case. Cyclocross uses a looped track, which allows for fixed omnidirectional camera points at the most spectacular points in the track. Cyclocross uses a closed track on a mixed underground (dirt, sand, mud, grass, water, ice, snow...), this is a very spectacular sport with thousands of spectators at the race and hundreds of thousands watching it on TV and online. Of all live sports coverage available, we selected cyclocross because of its possibilities to show the sport as if you were on the track with them, or as one of the spectators at the race. This will give us insights in how and when to use omnidirectional video and sound and will also give us some first directions on how to combine it with basic interaction and layering, e.g. to provide more information and background. The complexity of the live event will allow us to test real-time production tools and will teach us what works and what doesn’t as far as user perception, basic interaction and storytelling is concerned.*

The following goal for live production tooling was formulated on page 50:

Live production tooling: automated tools to assist the director in creating the immersive live experience, based on the different content and data feeds. The goal is to create an enhanced experience compared to current solution, without making the director’s job more complex.

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 (Task 2.4), the pilots (Task 4.2), and the technical evaluation (Task 4.3). The relationship between this task and the other related ImmersiaTV tasks is shown below.

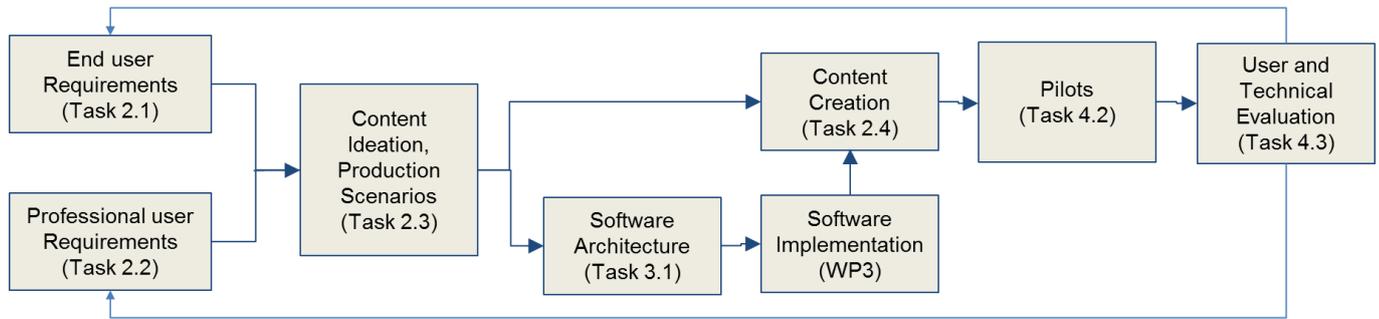


Figure 2: Relation between this deliverable and other WP2 tasks

2. CONTENT IDEATION

2.1. Off-line documentary

In this section, we 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).

2.1.1. 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.

2.1.1.1. 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.

2.1.1.2. 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.

2.1.1.3. The crew is visible in the 360° 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

2.1.1.4. 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

¹ <https://en.wikipedia.org/wiki/Rotoscoping>

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

2.1.1.5. 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

2.1.1.6. 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

2.1.1.7. 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 3).



Figure 3: 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

2.1.2. 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.

2.1.3. 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 2.1.1.6. 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: 360°

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: 360° (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 the main narrative, which 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: 360° (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: 360°

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: 360°

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: 360°

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: 360°

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: 180°+180°

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 (180°) and the teacher side (180°). 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 (180°) and the teacher side (180°). 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: 360°

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.]

2.1.4 Application of content format requirements in pilot 1

Taking the professional content format requirements listed in D2.2 into consideration, the next step was to practically realise these principles in the concrete pilot scenario. In this section, we show how the requirements have been incorporated in the actual shooting of pilot 1.

Regarding *R.2.1.1 The content creator shall take into account that VR experiences are about presence, "being there"* – the main practical consideration was on how to place the 360° cameras in function of two main aspects (in order of importance): story, spatial environment. Our primary concern is to respect whatever the story needs, and to define where the cameras should be from a story point of view. Then, how does this work in relation to the available space? Because the ImmersiaTV production workflow for pilot 1 is based on simultaneous shooting of shots for the different devices, the space has to accommodate for things like hidden cameras and crew, which inherently is limiting the available space. In order to avoid imposing a subjective point of view on the user, we decided to take the camera into a neutral point of narrative action in order to create a unprejudiced view of the scene. In scenes such as the locker room (scene 5) and the classroom (scene 9), the user is put right in the middle of the action.

Regarding *R.2.1.2 The content creator shall be able to use gamification paradigms to improve the storytelling for VR* – this was not implemented in pilot 1 scenario, for multiple reasons. The main focus of the first pilot was to successfully create the fully synchronised content throughout the three devices. In order not to jeopardize the viability of pilot, interactive portals had to be put aside. However, in later versions of the pilot, interaction has been gradually and successfully introduced.

A particular requirement that has been essential throughout the creative process of pilot 1 is *R.2.1.3 The content creator shall make proportional use of VR* – in other words, don't do everything in VR. This became clear quite early in the ideation process when we conducted tests

to see how 360° video could work in combination with traditional TV content. Instead of filming the narrative in the same mind set of directive content, the approach was to identify the added value of VR in each specific scene. In scenes such as the subway tram (scene 4) or the football game (scene 10), we opted to go with the atmosphere of the environment and to stay near the characters. Covering the game itself in 360 (which might look logic at first) would have created a totally different and probably less valuable experience.

This is connected to *R.2.1.4 The content creator shall be able to play with the viewing angle and viewpoints*. By getting insights in the appropriate and proportional use of VR for each scene, we were also able to control the different angles throughout the different scenes. Although a 360 multicam approach would have yielded better results, this requirement was successfully implemented in pilot 1.

Another aspect that is as crucial in traditional 2D storytelling as in 360° video is visual balance and rhythm, which is linked to *R.2.1.6 The content creator shall define the “beats” and find a good balance between static and dynamic shots*. Because camera movements easily can pose nausea-related and also technical problems, we opted for the opposite approach to let the action take place in a moving environment, e.g. subway tram (scene 4) and car (scene 8). In these scenes, the mise-en-scene is actually moving and the camera follows the narrative.

R.2.1.7 The content creator shall take care of the balance between guidance and freely looking around, is a very experimental feature in the context of pilot 1, and for the whole Immersia TV project. Not only are we crafting a TV and 360° narrative, but at the same time the experience on the HMD itself has to be designed very carefully. The content creator can indicate that there is interesting content on another device, but this has to be done in a balanced way which makes it a quite hard challenge. When incorporating guidance elements in the HMD experiences, it became clear that too much emphasis on guiding elements can break the connection between viewer and narrative.

Regarding *R.2.1.8 The content creator shall apply good usage of transitions and cuts in VR storytelling*, we used a fairly common (but effective) way to start and end the story: fading from black and fading to black before cutting directly to the end credits (using the score to correctly time it). For the transitions between scenes, we (initially) opted for luma matte transitions which provided a somewhat *tech* like style of going from one omnidirectional scene to another. In order not to disrupt the user abruptly, by suddenly taking him to another scene, a type of transition that is halfway between a cut and a fade seemed to be the most appropriate one.

R.2.1.10 The content creator shall give the user the choice when to switch between devices, is directly linked to the ImmersiaTV concept, i.e. to have synchronized content across devices. Best example of this in the pilot is the fade to black before the football game scene that both provide a natural transition between two acts of the story, while also allowing the user to transition between devices, if they wish to.

R.2.1.11 The content creator shall not mix TV storytelling with VR storytelling: since we do not want the omnidirectional view to simply be a 360° version of the point of view shown in the TV, we opted for different viewpoints, as often as the situation allowed it. In pilot 1, we always made a clear distinction of what works and is part of the TV storytelling and what belongs in the VR realm.

R.2.1.12 Apply interactivity via portals is not incorporated in pilot 1, but is part of its successor pilot 1.5 based on a multicam approach: a menu with graphical icons on the tablet device serves as portal to different views from multiple omnidirectional cameras. In the HMD view, screenshots of the different camera views are turned into round shapes and distributed in the

360 scene. The user can look around and has the possibility to access other content by triggering that portal.

R.2.1.13 The content creator shall be able to include extra graphical content in the HMD interface has also be implemented in pilot 1 by adding indicative signs of available content on the HMD device to the TV experience.

2.2. Live event

In this section, we introduce the selected use case, the key concepts and a reference scenario for the live ImmersiaTV pilots (pilot 2 and 3). The detailed considerations will form the basis for the production and end user scenarios (Section 3), and technical specifications and software requirements (Section 4) that have to be developed and elaborated in order to create immersive and synchronized multi-device content experiences in a live scenario.

2.2.1. Candidate Pilot Use Cases

From the user requirements (D2.1, D2.2) and feedback gathered during ImmersiaTV demonstration activities at IBC and NEM, the most interesting use cases to be considered for the live pilot scenario are:

- live sports coverage
- live music programme or show

In each case, the aim is to further enhance an already well-viewed popular live event-related programme with complementary and unique omnidirectional content in the head mounted display, where the viewer can experience the event by choosing from different viewpoints that are preselected by the director, at any moment during the event experience. While the viewer should be able to watch the entire event in the HMD if he wants to, the starting point is rather a television broadcast during which the HMD can be put or passed on to get an added-value experience along the event coverage.

This means that for instance even sports that are considered less suited for 360° content experiences, such as football (because of the larger viewing distances), are still suitable candidates for the ImmersiaTV concept, as long as they have the potential of providing interesting omnidirectional shots at specific moments along the play (e.g. near the goal, next to the coach, in the locker room).

Delivery of added value in the HMD while not interfering with the overall event experience that in many cases will be mainly driven by already very good TV content, is another notable facet to take into account. Here, an important aspect lies in the availability of sufficient occasions and time moments during the event to switch devices and story context without breaking the overall “flow” of the story experience.

In accordance with the ImmersiaTV DoW, a synchronized omnidirectional and directive coverage of a **live cyclocross** experience is selected as the reference demonstrator scenario for pilot 2 and will be further elaborated in this deliverable. Cyclocross is a very popular winter sport in Flanders, with one or two television broadcasts per weekend that reach high viewing figures (up to 1,5 million viewers for the most important races) and high market share (up to 80%) in the period of October until February. Races last about one hour, with both the men’s and women’s race being broadcast.

For live pilot 3, an entertainment show, music contest or alike will be selected.

2.2.2. Key pilot concepts

In this section, we introduce the key concepts of the end-to-end workflow related to the live pilot content scenario (see Section 2.2.3) that together will create and determine the overall content experience.

2.2.2.1. Combination of different live 360 camera systems

One of the main goals of the live pilot is to find a good combination of traditional directive cameras and complementary 360 cameras to achieve a coherent and added-value multi-device experience. Specifically for the omnidirectional cameras set-up, the idea is to test and optimize possible combinations of different types of 360 camera and stitching solutions. From a content perspective, the challenge is to assign and adapt the right camera to the right scenes and standpoints. More concretely we consider the use of following live camera equipment:

- integrated Orah 4i cameras developed by VideoStitch
- a new modular distributed camera system developed by iMinds/EDM, which will consist of edge capture, replay and per-camera processing units. The units can be adapted to specific capturing circumstances such as cyclocross tracks. EDM's Studio.One cameras include 1x 360 camera and 8x 4K camera with 170° field of view.
- GoPro/Elmo rigs combined with VahanaVR

For placement in-between a large audience, the Orah 4i camera is best suited. It has a small form factor, and due to its black color it does not attract attention. In addition it only needs a single cable (with a GoPro rig of 6 cameras, 12 cables are needed for live scenarios). The Orah stitching box can be placed up to 100 meters (good Ethernet cable) or more (fibre connection).

GoPro rigs require a computer running VahanaVR quite close to camera through HDMI cables, which is not convenient in crowded places. Compared to Orah 4i, there are no specific advantages related to the use of GoPro rigs except if there is need to beyond 4K@30 fps.

The Studio.One 360° camera is best suited for overview and atmosphere shots, e.g. from a central location on the circuit. The Studio.one 170° cameras provide more detail, and are suitable to capture interesting viewpoints along the circuit where there is not too many crowd.

Both Orah 4i and GoPro/Elmo cameras are not waterproof, a factor that has to be taken into account for the pilot as well. The Studio.One camera can be made relatively water-proof with kitchen foil.

In contrast to pilot 1, simultaneous shooting omnidirectional and (traditional) directive content impose no specific restrictions on how to shoot in each of the envisioned use cases. Specifically for the cyclocross pilot, people at the event itself are already used to the presence of capturing devices and other material to capture the event along the track. Therefore it is anticipated that this will not disturb the HMD experience neither, and no special camera placement requirements have to be taken into account for this matter.

2.2.2.2. New live production tooling

The aim is to demonstrate the ImmersiaTV-enabled equivalent of a live director production tooling environment, which is complementary to existing live production solutions, providing the following main features:

- Preconfiguration and customization of 360 scene compositions for HMD before the event, dynamically adaptable during the event. Within these scene compositions, user interaction points to specific camera viewpoints can be defined (graphical icons, portals)

- Live transitions for omnidirectional and directive content
- Live preview of HMD experience for operator
- The operator is able to provide user exploration and/or director's choice mode to the viewer
- The operator can define content for second screen (e.g. interactive mosaic pointing to omnidirectional or directive viewpoints)

In live pilot 3, an advanced version of the live production tool will be developed with the following two main additions:

- enabling **replays** and **repetitions** or additional off-line content e.g. event highlights. In case specific replays are activated by several HMD viewers at different points in time, these HMDs will not be in sync temporarily.
- allowing different scene lay-out configurations (e.g. location and number of portals/icons, type of interactions) per omnidirectional view

2.2.2.3. Enhanced live event experience

From a content perspective, the main purpose of the live production tooling is to create a synchronized multi-device experience that plays the fine balance between the director's choice in storytelling on the different devices, and free user exploration of the event in the HMD **without breaking the overall event experience**. The aim is to provide seamless transitions between devices, not only technically but also from a content point of view. The use of different devices should give the viewer an enhanced experience compared to traditional tv viewing. In pilot 3, the user will be able to switch on main TV audio and/or binaural audio from the scene location, potentially mixing the two signals. As such the viewer will be able to fine-tune their immersive experience in terms of combined storytelling and feeling of presence at the location.

2.2.2.4. Interaction within and between devices

The targeted platforms for live ImmersiaTV content experiences are:

- Television
- Second screen (tablet or smartphone)
- Head mounted display

The main milestone to achieve is synchronized live content delivered across these three devices, in particular between TV and HMD. The following interaction mechanisms, both between and within devices, can be taken into account to specify the requirements in terms of interaction:

- Head Mounted Display:
 - 8) The viewer is presented 360° content in an HMD with graphical icons used as interaction points in overlay. User interaction with these icons can activate new camera viewpoints e.g. by looking into the icon for a certain time duration, or by clicking a button on the HMD.
 - 9) Portals as part of the HMD experience can be considered as well, but should be used with care (i.e. not overlap with or disturb the main HMD experience). Portals can be interactive or static. In general, graphical icons are preferred over portals in live pilot content experiences. An example portal application can be a window on the TV content. Optionally, looking at it could switch to a TV view in the HMD, with an insert that allows switching back to omnidirectional view. In pilot 3, we envision the positioning of portals in relation to actual placement of omnidirectional cameras.

Real camera coordinates need to be represented in the 360° scene in this case, as illustrated in Figure 4.

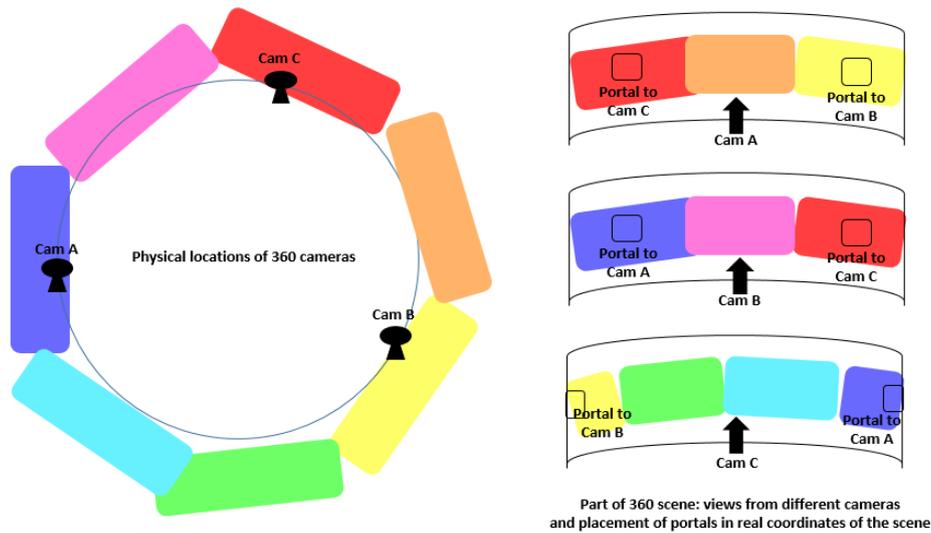


Figure 4 - Assuming cameras are placed in a 360° scene (various colours represent different physical background). If the view from camera A is selected, portals to Cam B and Cam C can be placed on top of their real background (specific angle).

- Second screen (tablet):
 - 10) A tablet user can see both traditional TV content and omnidirectional shots. In the latter case 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
 - 11) A multi-camera selection menu with the available omnidirectional streams on the second screen (tablet). Clicking on one of the omnidirectional streams in the mosaic enable 360 playout of the selected scene on the tablet. There is a possibility to trigger the selected content on the HMD, e.g. as entry content when a viewer puts on the VR goggles.
- Television:
 - 12) The TV shows a notification if there is particularly interesting omnidirectional content in the HMD (director’s choice). This notification also shows up on the tablet. Optionally, this omnidirectional content can be shown as an insert on the television by a specific action on the tablet or HMD (pilot 3).

2.2.2.5. Optimized quality of experience

Quality is one of the most decisive factors in the experience of omnidirectional content. Technical parameters such as bitrate, resolution, frame rates, and artefacts such as coding distortions, video stalling and delays can bring down the targeted quality of experience. A good interaction and feedback mechanism between a real-time logging-based quality metric system on the one hand and the ImmersiaTV encoder at the other hand, should optimize the end user experience. This includes the adaptation of content encoding taking into account the technical limitations of the network and the end-user devices and the stability of the transmission channel. To maximize the quality of experience, the technical parameters should be

continuously adapted depending on the computational load of the display-devices and the network bandwidth and stability.

2.2.3. Live pilot content scenario

2.2.3.1. Live pilot 2

For the live pilot 2, the aim is to demonstrate a synchronized multi-device experience of a Flemish cyclocross in the winter of 2017-2018, one that is already captured and broadcast by VRT for regular television broadcast. In October and November, VRT typically covers 5 cyclocross races, which leaves some room in the final selection of the pilot race, taking into account the weather forecast in order to avoid rainy conditions (used camera systems are not rain resistant). Most of these races have an attractive race circuit containing some spectacular passages or obstacles. These are the points of interest that are particularly interesting for 360 viewing, as these viewpoints can virtually “transport” the viewer from their home environment to the cyclocross race at the most interesting circuit locations. Note that even for spectators on the spot these locations are difficult to reach due to the massive presence of the crowd in such races (especially on points of interests).

How a cyclocross is traditionally captured

A typical cyclocross circuit covers an area of about one square kilometer, and is captured by 15 to 20 (depending on the circuit and the importance of the race) traditional TV cameras (see Figure 5). The live TV broadcast is operated from a central OB van, which is located somewhere on or near the circuit. The video signals of the different HD cameras are transported via triax cable to the CCUs in the control room where HD-SDI streams are generated (BCN connections). From the incoming feeds the operator selects the output TV signal that is transmitted to VRT’s main building (contribution over the air, or via satellite). Besides the producer, the director and his team, other tech personnel in the truck is taking care of the shader, video mixer, sound mixer, .. About five assistants handle the equipment truck. A video assistant helps the director on the field, cameramen handle the cameras along the circuit. In total, a team of 50-60 people take care of the overall capture.

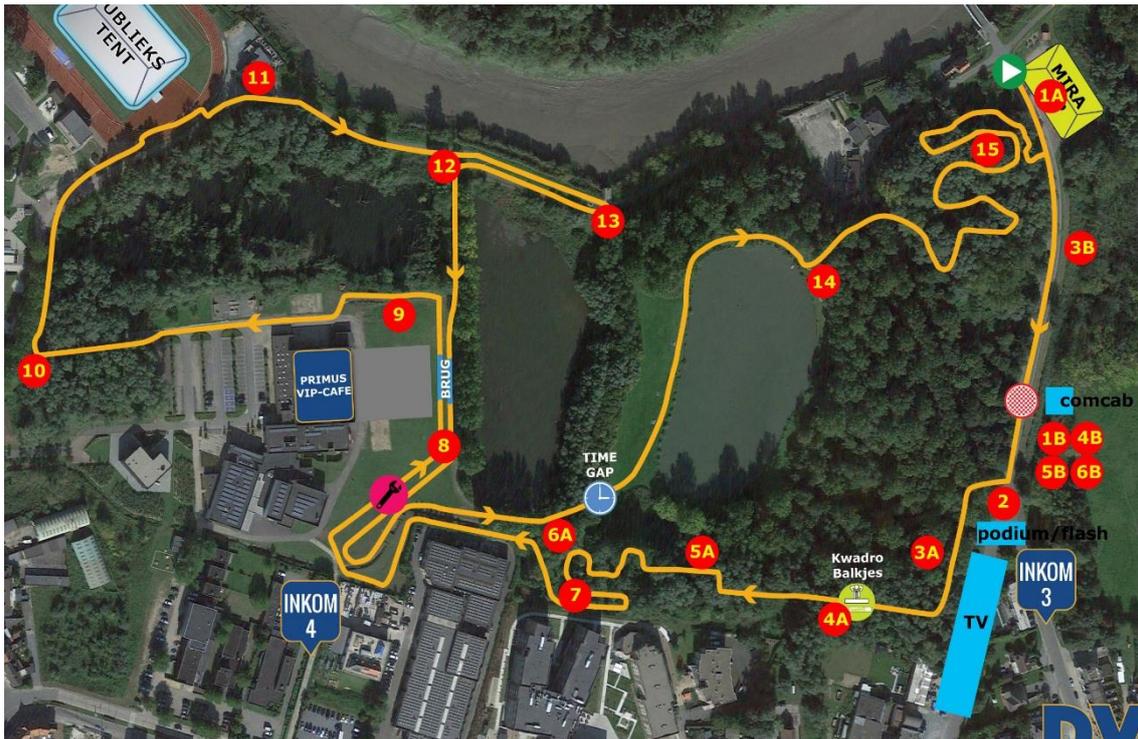


Figure 5: Camera plan for cyclocross capturing (Hamme, Belgium)

The ImmersiaTV cabinet

In the live cyclocross pilot, we plan to add an additional cabinet close by the existing OB truck in order to house the ImmersiaTV crew and the live production and distribution tooling. A safe assumption we can take is that a second operator is anyhow needed when creating additional live content for head mounted display, even if operations are mostly preconfigured. Next to the incoming omnidirectional streams, any directive HD camera stream of the regular tv capturing can be made available for ImmersiaTV purposes. In particular the output TV signal can be very useful to build a bridge between the story on tv and within the HMD. The following ImmersiaTV crew members are foreseen:

- Cameramen setting up and handling the cameras on the field
- Video assistant that can move around the circuit in communication with the director
- Director and team handling live production tooling before and during the race
- Tech people taking care of infrastructure equipment
- Tech people taking care of ImmersiaTV components of end-to-end flow

For the pilot, Nevia²-based fibre sets with break-out boxes both at camera and control car side can be used to transport the ImmersiaTV camera output signals to the car.

Reference cyclocross pilot content scenario

From the end user requirements (D2.1) and professional user requirements (D2.2), we introduce the following *reference pilot content scenario* for a live cyclocross race. In this scenario we

² Nevia – www.nevion.com

particularly focus on the HMD experience that enriches the tv experience. Figure 6 depicts the targeted end user experience.

When choosing to follow the race in the head mounted display, the viewer gets an omnidirectional **entry view** of the action that can either be a preconfigured viewpoint, the director's choice or the previous camera the viewer has watched in the HMD. Ideally, the transition between the tv and the HMD experience should be as seamlessly as possible, also from a content point view.



Figure 6: ImmersiaTV HMD experience for live cyclocross

From the entry view, the user is free to select another viewpoint. In the example, they can choose from 4 cameras in the field that provides (omnidirectional) content at particular interesting or spectacular passages. An additional camera view can be from a drone view giving an elevated overview of part of the circuit. In user interface, the different camera viewpoints are represented by graphical icons, each having a label clarifying from which location on the circuit the content is circuit. In the upper right corner, probably a bit outside of central image in the view, a map further helps the viewer where to situate the different cameras.

The pilots aims to offer exceptional viewpoints in the HMD at the first place, rather than follow the main action of the race as is the case in the tv reportage. The pace of scene changes is also expected to be considerably lower in order to achieve real immersion. The challenge is to balance the user exploration of particular viewpoints for a longer period in time, and at the same time to address adequately the fear of missing out important race events. In order to accommodate this specific challenge, we foresee to insert a **portal view** on the television content (see Figure 7, typically showing the main action or head of the race), which can be activated from the map by user interaction. We should pay attention that this portal content is well placed (e.g. aside from main zone of interest, but still within eye reach) so it does not interfere improperly with the primary omnidirectional video.



Figure 7: Portal showing head of the race (tv) content

In order to further address the **fear of missing out** the main race events or incidents, we can add the following director-driven elements to the HMD experience (see also Section 2.2.2.2):

- Offering the viewer the option to switch between *free exploration* and *director's choice* viewing mode. In the latter case, the director selects the most interesting viewpoint to watch, and presents it to the viewer.
- Indicating which camera is currently offering the most interesting content, by highlighting the icon representing that camera.
- Mixing the commentator's voice with the ambient sound: here the best of both worlds are combined: good race storytelling on tv, presence in the race in HMD. Moreover, as soon as anything particularly interesting happens, the viewer can easily put off the HMD, and follow the action on the main screen.
- Giving the director the ability to switch camera viewpoints on and off, according with the specific race situation (start, finish, interview afterwards). This can avoid that viewers watch uninteresting content and lose their interest in the head mounted display experience.

Other potentially interesting viewpoints include an inside view in the control car, an interview on the circuit by the moving reporter, after-race interviews in changing room,...

2.2.3.2. Live pilot 3

For the live pilot 3, we aim to demonstrate an advanced version of synchronized multi-device experience of a live entertainment show or music event, preferably indoors for practical reasons (in particular to facilitate Studio.One capturing). The selection of this event has to be approved by the production department (TV or Radio), and is not yet decided at this moment. However, this has no consequences on defining the user scenarios, software requirements and technical specifications for live pilot 3.

Generally speaking, the preparation (organization, production planning, rehearsals, ...) and capturing phase of a live entertainment show is largely similar to a live sports event (cyclocross). The main difference is that a live show comes with a detailed call sheet, reflecting the production schedule which follows the different events during the show. Similar to the camera plan for the cyclocross circuit, a detailed venue plan including stage and audience area is used to prepare the capturing of the event (placement of cameras etc..). For the end user, in contrast to pilot 2, the free exploration mode will be emphasized compared to the director’s choice view. The interaction design will be refined (individual portal configuration per scene), and the concept of replays will be introduced.

As a reference scenario for live pilot 3, think of the Eurovision song festival. Figure 8 and Figure 9 depict the targeted end user HMD experience. In addition to the interactive live experience, the viewer has to option to initiate replays of passed songs, hereby mixing live and on-demand content in the HMD experience. In the new view (replay), the scene configuration is different: the viewer has the option to return to the live show or optionally select a replay of another country, which happens to be popular amongst viewers. At any stage of the experience, the viewer can pause the live stream and resume playing at their convenience. In the live experience, if anything spectacular or unexpected would happen e.g. backstage while a viewer is watching from the “best seat”, the director can “enforce” a scene switch (director’s view) if he wants to. Generally speaking however, the spirit of the live pilot 3 is to intervene as little as possible to clear the way for maximal user freedom and interactivity. The whole is provided with an extra degree of immersion by providing ambisonics sound to the HMD viewer.

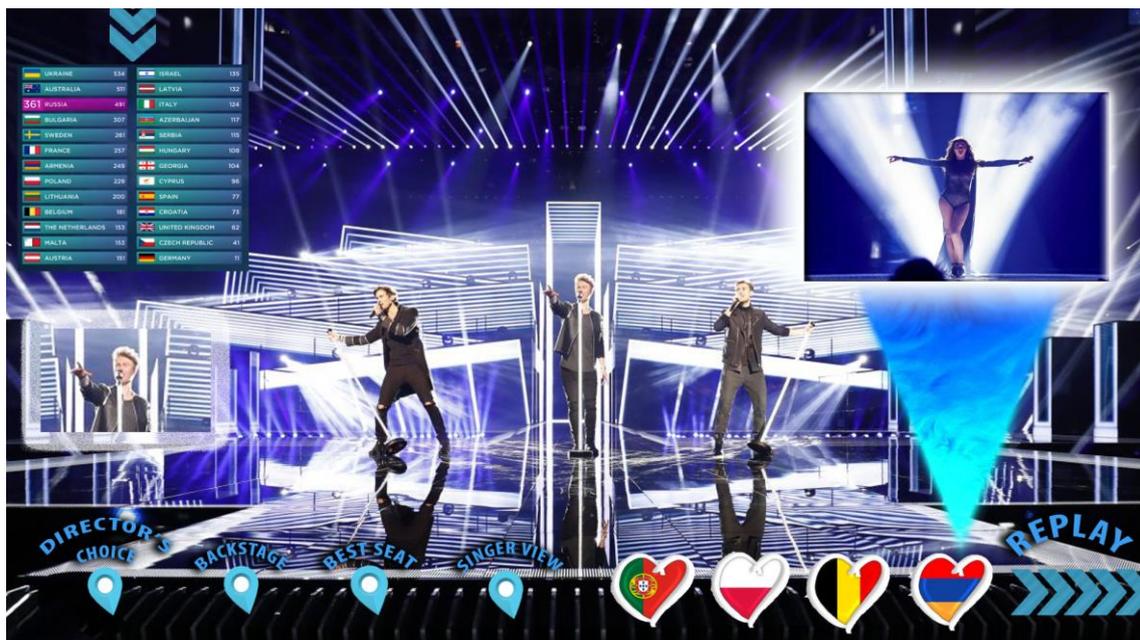


Figure 8 - Concept of replays in live ImmersiaTV experience.



Figure 9 - Configurable portal set-up per omnidirectional scene.

2.2.4 Application of content format requirements in live pilots

2.2.4.1 Live pilot 2

In live pilot 2, the D2.2 content format requirements feed an ideation process that results into a live pilot scenario: a cyclocross race. The ideation process from pilot 1 is clearly different from the one of pilot 2 as a live broadcast has no pre-written storyline, the story unfolds on the moment itself. There are location based conditions that put boundaries in the ideation process on how to capture the live story in the best possible way. Also the nature of the event (sport, music show) will push the ideation in a certain direction.

Cyclocross is a very interesting live pilot case for the ImmersiaTV concept as the viewer can follow the race through a live TV broadcast and at the same time (in the hmd) have the experience *being present* (R.2.2.1) on the very best places at the cyclocross race standing between the loudly yelling spectators (R.2.2.3) around the track. The synchronisation (R.2.2.2 & R2.2.5) between the devices (tv, tablet/hmd) will give the viewer a full coverage of the race, not missing out the best moments.

As part of the ideation process, a mock up interface was created (see Figure 6 and Figure 7) to get a clear view on how the Immersiatv concept can be acknowledged to the viewer. The content format requirements in D2.2 were inspirational in building this mock up and most of the interactive requirements (including the portal concept) have been applied.

In the top right corner (Figure 6) a map shows the spectator from which corresponding camera location he or she is watching the race (R.2.2.4). By interacting with the map (Figure 7) the user can switch to a live feed of the live directive broadcast (R.2.2.5).

On the bottom of the mock up, different icons represent the different camera set ups around the track (R.2.2.7). These icons are portals which the viewer can activate and thus navigate to one of the different camera viewpoints (R2.2.6) around the field. One of the icons named 'director's choice' is self-explanatory, giving the viewer the 360 video feed edit of the director.

In live pilot 3, replays (R.2.2.9) are offered and mixed with the live experience (Figure 8 and Figure 9). Portal set-up can be configured per omnidirectional scene (R.2.2.10). The director can force specific omnidirectional views (R.2.2.12), or the sharing of HMD point of view to the television (R.2.2.10). From this, the content format requirements of D2.2 are clearly mapped to the live pilot content scenario.

2.3. Pilot 3 off-line: exploration mode (whodunnit)

2.3.1. Candidate Pilot Use Cases

The selected off-line pilot 3 use case is a whodunnit multi-device content experience, based on exploration mode. Each device (TV, HMD, tablet) has its own dedicated function in the overall experience. In contrast with broadcast mode (pilot 1 and 2, and live pilot 3), in exploration mode the TV scenes are decoupled from the HMD/tablet scenes in time, i.e. they do not overlap. Exploration mode is activated by breaking out from the main TV storyline to the second screen devices. By applying this concept, professional storytelling is mixed with user exploration capabilities, and the goal is to find the ideal balance between them. Social interaction between viewers beyond jointly watching TV content is particularly encouraged by HMD-tablet interplay. Exploration mode also adds some gamification techniques to the ImmersiaTV experience.

Based on D2.2 findings, a whodunnit scenario is ideally suited to work out and test the exploration mode concept. In this section, the key pilot concepts are explained and the whodunnit script is elaborated.

Note that in contrast with broadcast mode, exploration mode inherently (by design) does not pose concurrent directive and 360° shooting challenges.

2.3.2. Key pilot concepts

The off-line scenario of pilot 3 introduces a new mode called Exploration Mode, in which the following concepts are introduced and elaborated.

2.3.2.1. Devices and behaviour

- **TV:** displays main thread of the movie and leads users across the interactive story. Three different options of TV experience can be considered while exploration mode on HMD/tablet is activated:
 - **Video:** Looped video clip. This mode is preferable. The image may be either a looped video clip or even a cropped video of the omnidirectional scene – in this case the virtual camera will be moving slowly around the scene using a defined path. The path around the omnidirectional scene can be pre-defined by an external tool.
 - **Static image:** for example a wide angle view on the scene. This is a simpler case of using looped video and may be useful in some scenarios.
 - **VR view:** the same view as the HMD viewer is shown on television screen
- **HMD:** when exploration mode is active, the person wearing the HMD is able to look around in an (interactive) 360° scene and can select objects.
 - The omnidirectional scene visible in HMD is animated (video clip). It can be either static or dynamic: by “static” we mean for instance a view of the room

where a crime was committed, in which only a few elements are in motion (e.g. leaves waving outside the window). “Dynamic” means, for example, the the visualization of the actual crime.

- During exploration, some objects can highlight when the viewer is focusing on them and they can be picked up by clicking an HMD button, or an additional video can be played. Then the object appears on the tablet (e.g. in the list of Evidence) with their additional description.
- The time duration of the scene in Exploration Mode is limited. This means a user only has a predefined time window for exploring the scene. Optionally he or she can be informed about the remaining time (by displaying it as a portal or beeping with decreasing number of beeps every half minute, etc.).
- Although a user cannot pause or rewind the scene to the beginning in a standard scenario, in some other cases this might be useful. As we cannot foresee all possible scenarios now, the functionality of pausing/resuming and replaying should be defined and implemented.
- All HMD displays are synchronized in Exploration Mode, which means that all people wearing VR glasses are at the same moment in the storyline. This will prevent unwanted time differences when crucial action is happening.
- **Tablet:** Users with tablets can examine objects selected in the scenes watched in HMD. Each object is depicted by a real or rendered image and described by a few lines of text. The user goes through the list of collected objects as well as the object descriptions of objects and shares his “intelligence” with the HMD user. The tablet also allows to switch the TV view from virtual camera/static image to VR view (screen cast). If more people are in exploration mode with HMD, the tablet menu should give the possibility to switch between different VR views from HMDs.

2.3.2.2. Roles (social viewing)

By default, we assume that the minimal number of people using exploration mode is 2, however we can define 3 roles for people. If there are more people “playing” exploration mode, they take one of the following roles:

- **Explorer:** he or she is exploring a 360° scene in HMD and can pick objects. The view from the person’s HMD may be streamed on television so other people are able to see what he or she sees in HMD. Usually one person uses an HMD which simplifies things, but if more people have HMDs, each of them may pick up objects and add them to the tablet list.
- **Assistant:** a person using a tablet to examine objects picked up by the Explorer. The assistant analyses collected objects and confers with the HMD users. He/she can also switch the TV view between static and VR modes, and chooses which HMD view to mirror in case there are multiple explorers.
- **Observer:** person not being directly engaged into Exploration Mode. He or she can observe the “game” and give advice.

2.3.2.3. Privacy issues

The only privacy issue we foresee is sharing one’s HMD view with Assistant and Observers. The person being in Exploration Mode as an Explorer may not necessarily want to share their experience. On the other hand, we may assume that people playing Exploration Mode should be eager to cooperate with Assistants in order to resolve the whodunnit, and agree in advance with sharing their VR view in advance. The software on HMD should allow to share the view with

others. For this, the proper button should be activated in tablet view in order to show the VR view on TV.

2.3.3. Pilot 3 content scenario

We have prepared a detailed script for a whodunnit called “The hollow man”, which implements the concept of exploration mode. In order to maintain legibility of this deliverable, the elaborated script is added as a separate document (pdf-file) to this deliverable: IMMERSIATV-PILOT 03.pdf.

Figure 10 and Figure 11 show the wireframes illustrating the Exploration Mode HMD experience based on the whodunnit scenario. If the viewer detects an object (in the example a gun) that hides a portal, a sphere-like image pops up in the omnidirectional view. By staring at the image for a couple of seconds, the portal is activated and plays the hidden content. The content’s narrative and related information is simultaneously loaded into the tablet’s “notebook”. A notification icon (top right) is signalling the viewer that new information is transferred to the tablet.

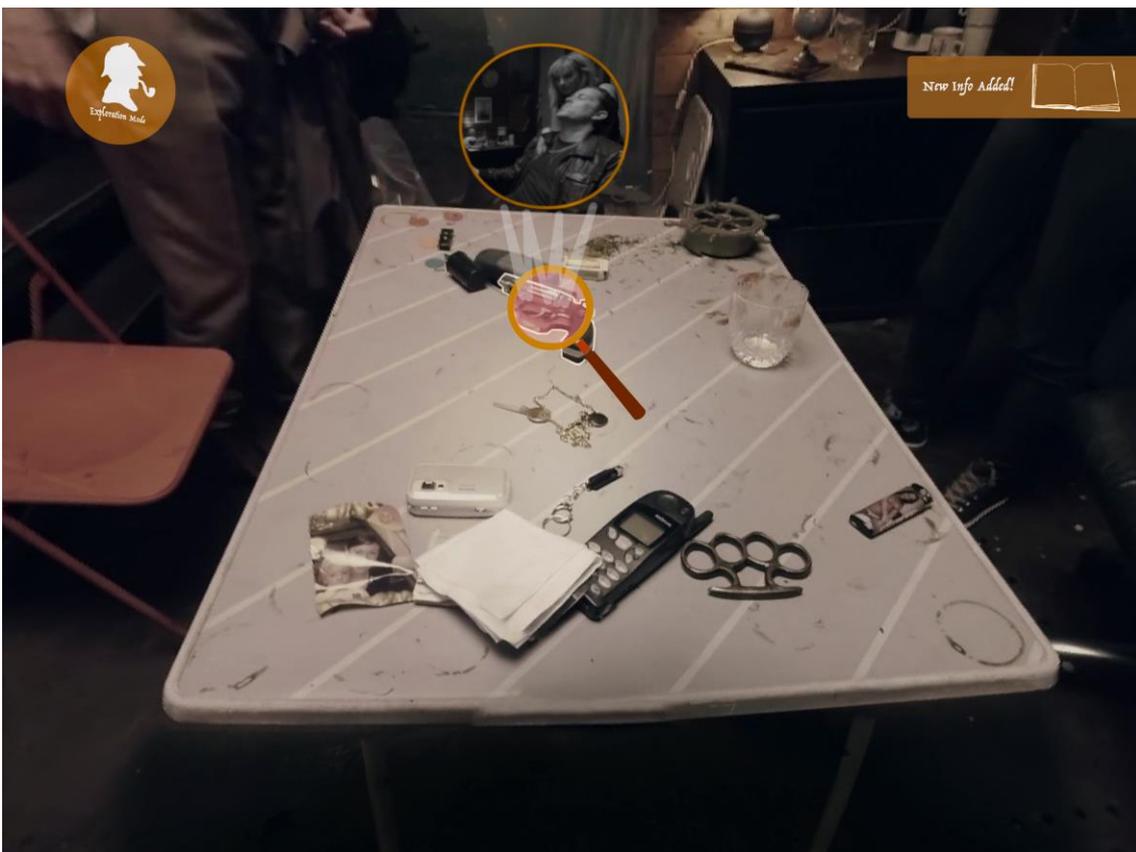


Figure 10 - Wireframe of HMD experience (pilot 3). The magnifying glass represents the user’s point of view.

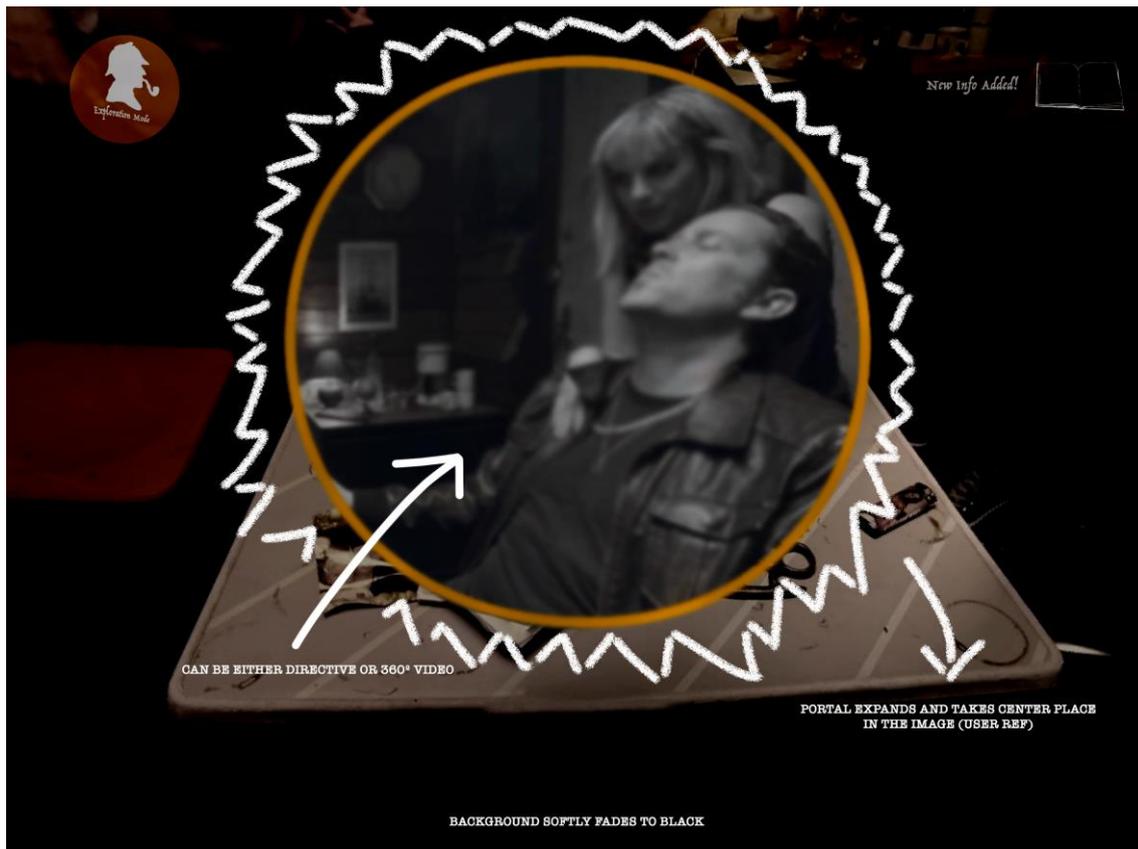


Figure 11 - HMD view of exploration mode (The Hollow Man)

In the tablet view, different tabs with distinct information will be presented to the viewer:

- “Trifecta”: quest to find the culprit and solve the crime step-by-step
- “Case Overview”: a structured view on the puzzle to solve the crime
- Profiles: in-depth information on possible suspects, their alibis, etc..
- Evidence: structured list of suspicious objects and revealing elements, when/where they are found and by whom

Each tab is dynamically updated, based on (inter)actions of the HMD viewer(s).

TRIFECTA CASE OVERVIEW PROFILES EVIDENCE

Culprit trifecta

(This is what this looks like MIDWAY of the Pilot)

Eugene Legrand

CASE OVERVIEW PROFILES EVIDENCE TRIFECTA

The Puzzle

(This is what this looks like at the END of the Pilot)

Eugene Legrand

PROFILES CASE OVERVIEW EVIDENCE TRIFECTA

(This is what this looks like at the START of the Pilot)

Anton Smith
43 years old
dead brother
Has saved Mr. Ample for 21 years

ALIBI

UNAVAILABLE

Angela Smith
35 years old
kind and
Has saved Mr. Ample for 18 years

ALIBI

UNAVAILABLE

Eugene Legrand PAGE 1/2

EVIDENCE CASE OVERVIEW PROFILES TRIFECTA

(This is what this looks like at the START of the Pilot)

FOUND IN: UNCOVERED **BY:** UNCOVERED

Botulinum Toxin

FOUND IN: UNCOVERED **BY:** UNCOVERED

ARSENIC. POISON!
Arsenic

FOUND IN: UNCOVERED **BY:** UNCOVERED

Cyanide

Eugene Legrand PAGE 1/2

Figure 12 - Tablet view of pilot 3: Trifecta (top left), Case Overview (top right), Profiles (bottom left), Evidence (bottom right)

2.3.4. Application of content format requirements in pilot 3 (Exploration Mode)

In this section, we show how the content format requirements (see Deliverable 2.2, Section 3.4) have been incorporated in the actual script of the off-line pilot 3.

R.2.3.1 – seamless storytelling across devices: the “break outs” to exploration mode are introduced with care. The first time exploration mode is applied, an “inciting incident” story element is used to trigger the curiosity of the viewer, making him eager to switch devices to continue the storyline.

R.2.3.2 – simple and coherent interface: the design of the detective’s notebook (Suspects/Profiles/Evidence bar) is very intuitive and serves the purpose of the whodunnit. The same applies for the portals introduced in the HMD view.

R.2.3.3 – lead device vs exploration mode devices: The lead device of The Hollow Man whodunnit clearly is the television, where the investigation and denouement takes place. Exploration mode is introduced with care, mainly to provide flashbacks taking the viewer back in time to the moment the crime was committed, without disturbing the flow of the narrative.

R.2.3.4 – Each device has its own relevance and added value: the HMD experience brings a subjective perspective (through the eyes of the detective, Mr. and Mrs. Smith and Eleanor) to the whodunnit story, giving the viewer a more active role in the story. By looking into specific suspicious objects (the books, whisky glass, ...) additional information pops up in the detective’s notebook (tablet) or additional content is shown in a portal inside the HMD.

R.2.3.5 – Creation of break-out segments where the viewer enters exploration mode: this has to be done at script level. 4 break-out segments have been defined in the script, which is specifically designed for exploration mode.

R.2.3.6 – Exploration mode is properly signalled to the viewer by TV content: this has to be added in post production stage.

R.2.3.7 – Careful application of exploration mode (frequency, timing): limited to 4 break-out segments following the rhythm of the story and respecting the main story arc. Each break-out segment brings a specific added value.

R.2.3.8 – Storyline on main device (TV) is paused when exploration mode is activated: once in exploration mode, the television shows what the HMD viewer sees.

R.2.3.9 – Exploration mode (pilot 3) and broadcast mode (pilot 1) should not be mixed: there is no HMD content played when the main storyline is told on the television and vice versa. In exploration mode, the television just shows the HMD content to non-HMD viewers. However, the detective’s notebook (tablet) responds to content shown on both television and user interaction (e.g. looking into an object) on the HMD. This is a classical second screen extension of broadcast content, and does not correspond to broadcast mode.

R.2.3.10 – User interaction is only applied in exploration mode: interaction is implemented by interactive portals (HMD) and information on the tablet.

R.2.3.11 – Interactions between HMD and tablet in exploration mode: in the HMD the viewer explores the crime scene, and by looking into objects, information will appear on the tablet.

R.2.3.12 – Return from exploration mode to main storyline: the timing boundaries of exploration mode can be defined in post production.

R.2.3.13 – Branched storytelling in exploration mode: for reasons of complexity, branched storytelling is not applied in the basic pilot 3 script. Giving it a specific purpose in the storytelling, would take it too far in the scripted whodunnit.

R.2.3.14 & R.2.3.15– Social viewing aspects have to be taken into account: interaction between viewers is expected to be stimulated based on the interplay between HMD and tablet (detective’s notebook), and by showing the HMD view on the television during exploration mode.

3. USER SCENARIOS

3.1. Off-line documentary

3.1.1. Pre-production

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.

3.1.1.1. 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

R-STORY-1 The content creator can create the main storyline

R-STORY-2 The content creator can define the main and side characters

R-STORY-3 The content creator can define the detailed story structure

R-STORY-4 The content creator can define the sub-storylines for TV, tablet and HMD

R-STORY-5 The content creator can define the user interaction design

(For each scene:)

R-STORY-6 The content creator can define the multi-platform logic

R-STORY-7 The content creator can define the viewer perspective(s)

R-STORY-8 The content creator can define the detailed script

R-STORY-9 The content creator can define if it uses Omnidirectional and/or directive content

R-STORY-10 The content creator can define the viewing angle

R-STORY-11 The content creator can define the interaction points: portal, AR object, caption, graphics, ...

R-STORY-12 The content creator can define transition between scenes

R-STORY-13 The content creator can specify use of audio for guidance and transitions

R-STORY-14 The content creator can indicate use of camera movements

R-STORY-15 The content creator can indicate forced exploration mode where applicable

R-STORY-16 The content creator can save resulting format script as master template

3.1.1.2. 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

R-PREPROD-1 The content creator can add on-location research material as placeholder in format script

R-PREPROD-2 The content creator can perform first VR preview of relevant scenes

R-PREPROD-3 The content creator can define the shooting plan

R-PREPROD-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.

3.1.2. 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 2.2.2.2 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 **¡Error! No se encuentra el origen de la referencia.**).

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.

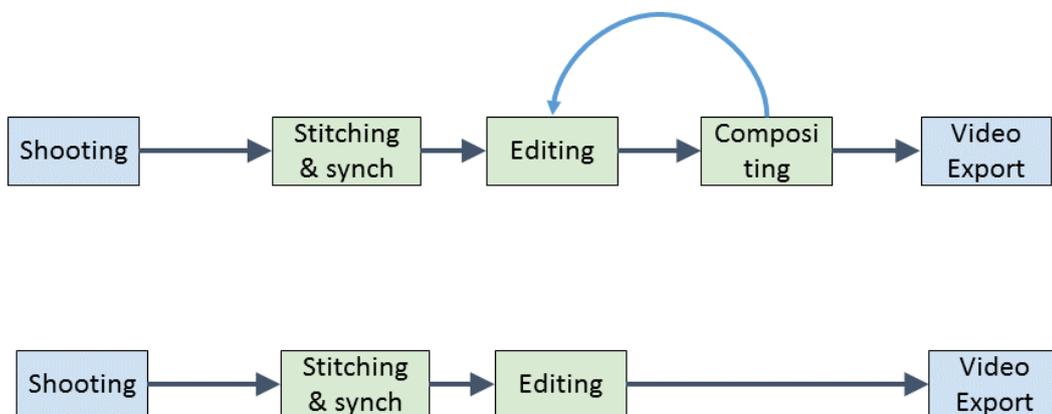


Figure 13: ImmersiaTV post-production workflow. On top, the default process of omnidirectional video post-production. This involves a significant amount of compositing, i.e., advanced post-production. At the bottom, a

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.

3.1.2.1. 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:

R-EDIT-1 The content creator can visualize the raw material across the different end-user devices.

R-EDIT-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.

R-EDIT-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.

R-EDIT-4 The content creator can use Windows and OS X.

R-EDIT-5 The content creator can make use of an advanced mode in a compositing software (Nuke, Adobe After Effects).

R-EDIT-6 The content creator can introduce interactivity within the editor timeline through *conditional transitions* between shots and scenes

R-EDIT-7 The content creator can select, within the editor timeline, which video assets are visible within the TV, the tablet and the HMD

R-EDIT-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 2.2.2.2) 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

R-EDIT-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. The software reads the black video as the part of the video that is invisible and white for the part that is visible (see **¡Error! No se encuentra el origen de la referencia.**).

- 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 **¡Error! No se encuentra el origen de la referencia.**, 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:

R-EDIT-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:

R-EDIT-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 **¡Error! No se encuentra el origen de la referencia.**)

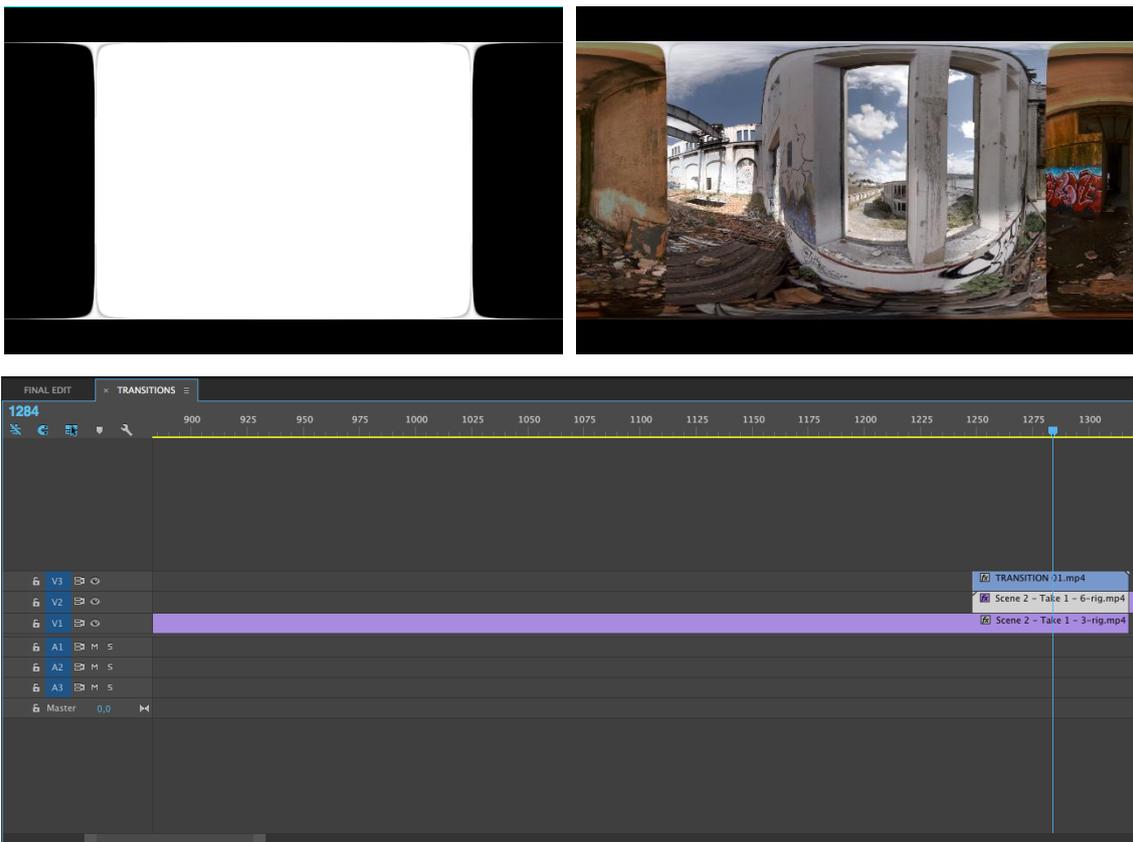


Figure 14: Interactive transitions between omnidirectional videos using matte. On top left, a frame of a black and white video matte video. On top right, the result of applying such mask on two omnidirectional videos. On bottom, the edition timeline corresponding to the output shown on top.

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.

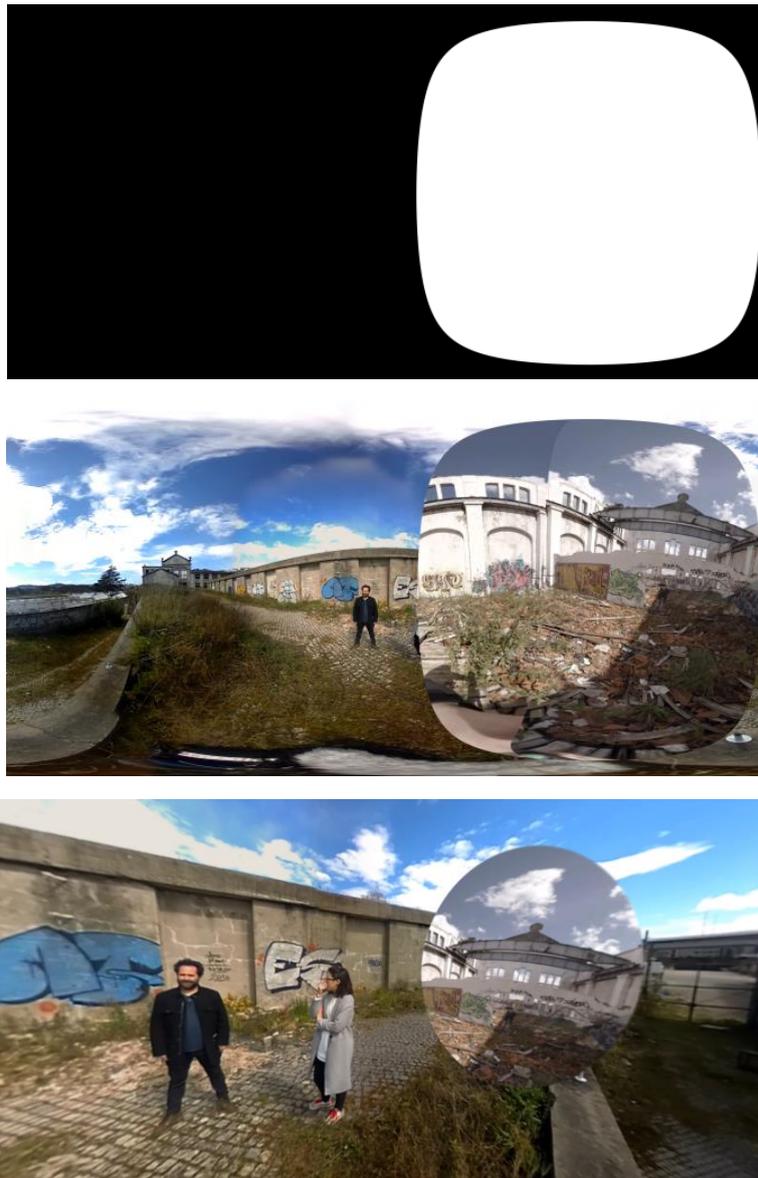


Figure 15: Inserting portals by applying video masks in video. On Top, Black-and-White Image Generated in a compositing software (Adobe After Effects) with a third party plugin (Mettle’s Skybox), in equirectangular projection. In the middle, the black and white video matte applied to two omnidirectional 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.

3.1.2.2. Pre-visualisation requirements

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

R-EDIT-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 **¡Error! No se encuentra el origen de la referencia.**)

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

R-EDIT-13 The content creator will be able to pre-visualize transitions and interactive transitions within the editing software

R-EDIT-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.

3.1.2.3. 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:

R-EDIT-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:

R-EDIT-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*).

R-EDIT-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.

R-EDIT-18 It will be possible to define a label specifying the destination for each sequence

R-EDIT-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.

3.1.3. Content playback

The user scenario introduced in section **¡Error! No se encuentra el origen de la referencia.** 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 **¡Error! No se encuentra el origen de la referencia..** In the following subsections the major functional requirements are listed and elaborated.

3.1.3.1. 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:

R-PLAY-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.

3.1.3.2. 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:

R-PLAY-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.

R-PLAY-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.**

R-PLAY-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).

R-PLAY-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 **¡Error! No se encuentra el origen de la referencia.**)

R-PLAY-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).

3.1.3.3. 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.

3.1.3.4. 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:

R-PLAY-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.

R-PLAY-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.

3.1.3.5. 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.

R-PLAY-9 Basic audio control in the end-user devices

3.1.3.6. 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 functionality 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:

R-PLAY-10 Real time communication channel between devices: sending messages from one device to another

R-PLAY-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.

3.1.3.7. 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:

R-PLAY-12 The end-user can capture screen casts and share them with other devices

R-PLAY-13 The end-user can capture screen casts and share them through social media

3.1.4. Encoding

As a general rule, codec parameters (video format, bit rate, frame rate) for pilot 1 are determined by the capabilities of the reception device, such as a typical HMD, a tablet or a smartphone. The goal in ImmersiaTV is to minimize the number of transcoding steps needed. Where possible, the influence of each encoding step on the quality should be minimized. At

capture stage, guidelines for off-line recording are provided in Deliverable 3.1 (Table 7, Section 3.4.2.1).

3.2. Live event

3.2.1. Production scenario

The live pilot 2 scenario defined in Section 2.2.3.1 requires a production workflow that follows the following professional user scenario:

Professional User scenario: The Sports department of the public broadcaster VRT is preparing for the next live coverage of a cyclocross. The race contains some really spectacular passages, therefore the producer decides to go for a multi-device coverage using the ImmersiaTV platform such that the viewer gets an enhanced experience, including the feeling of being there in the heat of the race enjoying the best standpoints on the circuit.

One month ahead, a technical visit on the race circuit is performed to examine and define the different camera standpoints. Next to circa 15 HD cameras for TV capturing, the best spots for omnidirectional capturing are selected. The technical responsible and the director debate and agree upon the technical and creative possibilities to place the different cameras, with a joint storytelling on TV and HMD in mind.

In the run-up towards the cyclocross, further preparations are being made: planning of activities, personnel staffing, transport, technical equipment details, etc.. One or two days before the race cabling, power infrastructure and control car (OB van) are installed. As an extension of the OB van, a specific ImmersiaTV cabinet is constructed and the different ImmersiaTV components such as live production tooling and streaming encoder are already installed.

Early in the morning of the day of the professional race all the cameras are built and placed, both the TV cameras and the additional omnidirectional cameras. Two Orah 4i's are placed at respectively a steep descent, and a decisive "sandy" passage. These two highlight spots are located at distant points from each other. The other omnidirectional cameras (e.g. one additional Orah 4i, two Studio.One 170° cameras) are placed on carefully chosen points of interest, such that all together the "flow of the race" is also captured more or less completely through the circuit. However, the frequency of omnidirectional follow-up shots remains (on purpose) significantly less than for the tv coverage. The Studio.one 360 camera is placed in the tv control car, to give the end user a unique view of how the race is directed. A drone camera or a camera placed on a very high crane provides an elevated omnidirectional view of the spectacle. Once installed, the delays between the different cameras (stitched output) are defined using clapperboard.

After the briefing, the first races of the day, which are not broadcast, take place giving the opportunity for try-outs by the production crew: fine-tuning of camera positions, quality of camera (stitched) images, director actions, ...

The experience created for the head mounted display is done by a second director, in addition to the traditional tv content director. The second director can "shape" the user experience by defining one or more 360 scene compositions for the head mounted display view, that typically consist of a primary omnidirectional video plus additional interactive icons, graphics and one or more portals. He assigns positions and size of inserts, defines if these elements are referenced to the user or the world, and connect the graphical icons to the respective directive or omnidirectional cameras. If portals are used, either directly or after activation triggered by user interaction, the director defines what portion of omnidirectional or directive video is visualised

into the portal(s) and how the cropped video is shaped. Predefined templates makes this pre-configuration process an easy job.

To prepare the “director’s view” mode, the director can test and preview transitions between two omnidirectional shots during the pre-races, and also fine-tune the 360-based storytelling that optimally follows the rhythm of the race.

When the professional race starts, the director for HMD view is fully prepared for the job. He selects a preconfigured scene composition to run the *user exploration* mode, and from then on mainly concentrates on the *director’s choice* storyline. At the output side, the different used audiovisual streams (one coming from the tv director) and live production actions (metadata) are encoded and streamed. The tv stream is created by the tv director.

Occasionally, the director signals the viewer if one camera is particularly interesting, by highlighting the icon. Optionally, important race evolutions can be “pushed” as portal content containing the directed “head of the race” tv content or a selected ISO camera feed, even if the user had not switched on the portal. When the race is finished, the director switches to a second scene composition, containing a 360 overview shot of the moving crowd, a portal showing an interview with the race winner and one viewpoint icon for the commentator cabin view.

Finally, after the broadcast is finished all cameras and cable infrastructure are cleared away.

For live pilot 3, staging a live entertainment event, the main extensions of the production scenario are:

- The director is able to record event highlights, edit these highlights and add event replays to the interactive omnidirectional experience.
- The director can configure the scene layout (portals, interactions) for each omnidirectional camera individually
- The director can generate ambisonic sound

These extensions are also applicable to a live sports (cyclocross) scenario.

3.2.2. Consumption scenario

The scenario below describes the end user scenario for live pilot 2.

End User scenario: Tom and Laura are two cyclocross fans that enjoy each weekend in winter the television broadcasts of their favourite sport. This time, they are especially excited because they will be able to experience a live cyclocross experience that is fully synchronized on their tv and head mounted display. Their expectations are high: for the first time, Tom and Laura will be able to enjoy best of both worlds: TV storytelling which gives the best possible view on any detail or event about the race, and the feeling of presence on the circuit at some of the best possible locations – all this at the same time! Laura’s mother prefers the tablet as a second screen device, where she can pick a viewpoint of choice during the race.

A few minutes before the race starts, Tom switches on his connected television and selects the cyclocross event to start playing. He makes sure his head mounted display is ready and within reach. He watches the start and first round of the race on his television, and then opens the ImmersiaTV app on his mobile phone for watching in HMD. Tom is eager to discover some of the interesting locations of the race, but prefers to follow the “flow of the race” so for the time being he selects the “director’s choice” icon, to be sure not to miss out anything important. He likes the new experience, it feels like an in-between experience of being at the race location (but being able to “hop around” more easily and frequently), and being comfortably at home missing out nothing. At half race, Tom gets curious how such a multi-device experience is actually

directed. He switches to the user exploration view and triggers the “control car” icon to get a first-row view of the director and his team covering the race. After a while, he sees in the right corner portal that a decisive action is taking place. Tom puts off the HMD, and watches the race final on the television big screen. After the race, he puts the HMD on again, to enjoy the post-analysis in the commentator cabin in 360.

Laura and her mother are also watching the race. In the past, they used to go to cyclocross races regularly, but last years due to the popularity of the sport it became a bit too crowded and best positions became too difficult to claim. With ImmersiaTV, they can enjoy a similar “being there” experience, from their living room. After a couple of laps, the race positions are fixed and Laura gets attracted to explore some of the spectacular views in the HMD. She first activates the drone view to get a 360 overview of the race circuit. Meanwhile, her mother explores the race viewpoints on her tablet which are shown as a two by two mosaic. At one spot, there is a crash so she opens the viewpoint and moves her tablet to look around for which riders are impacted by this crash. Laura had seen a sign of the director on TV and was in the same view and they discuss what’s happening with each other. Laura passes the HMD to her mother for having a better view.

Further on, Laura gets intrigued by specific difficult passage, and decides to “stay” there for a while to see how the riders, now scattered across the circuit, are handling this race obstacle. In the background she hears both the ambience and the commentator voice, so she remains perfectly aware of the race evolutions even if she stays at the same place. At the end of the race, they watch the final sprint together on the television.

For live pilot 3, the viewer will benefit from the following additional functionalities:

- The viewer can pause and resume play-out
- The viewer will be presented a list of additional offline content (e.g. replays)
- The viewer can switch between live and offline content
- The viewer will be notified if new additional content is available
- The viewer will enjoy ambisonic sound
- The viewer can publish their actual view on the 360° content on social media

These additional functionalities can be applied in both live entertainment and sports events.

3.2.3. Technical specification and requirement analysis

In order to demonstrate the key concepts outlined in Section 2.2.2, to realise the live pilot content scenario (Section 2.2.3) and to enable the production and consumption scenarios (Section 3.1 and 0), the technical specifications and different software requirements of the end-to-end ImmersiaTV workflow for live omnidirectional content production need to be specified. In this section, we will define these specifications and requirements for both pilot 2 and pilot 3 each of the building blocks depicted in Figure 16:

- Capture
- Stitching
- Live production system
- Encoding
- Distribution
- Reception, Interaction and Display
- Quality of Experience

These list of specs and requirements are the outcome result of internal technical brainstorms between project partners. More details of technical implementations and functionalities of these building blocks can be found in Deliverable 3.1.

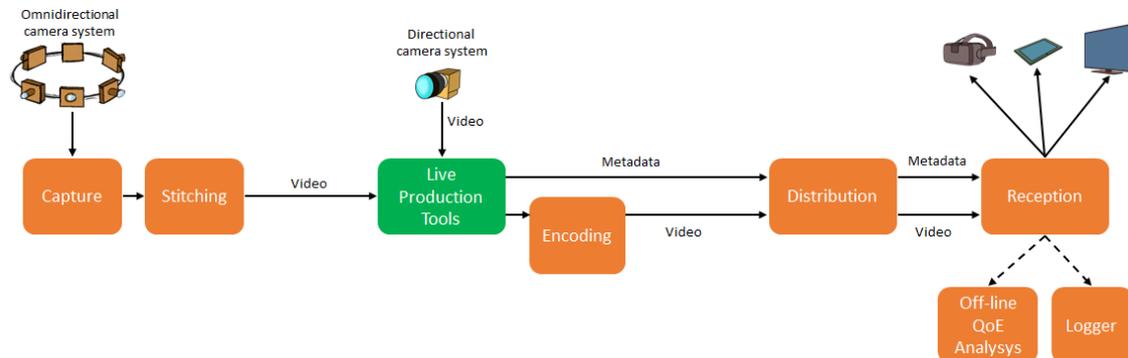


Figure 16: End-to-end ImmersiaTV workflow for live content

3.2.3.1. CAPTURE

The cyclocross pilot, including preparatory testing events, will support various use cases and different camera systems. This implies the following requirements:

- **R-CAP-1 Omnidirectional and directional cameras:** Orah 4i, GoPro rigs and EDM Studio. One cameras (both 360 rigs and up to 8x 170 degree semi-omnidirectional high resolution cameras). The directional cameras used for the cyclocross pilot are Grass Valley LDK 8000 cameras (HD 1080i).
- **R-CAP-2 Camera distances:** to deal with a variety of distance between cameras – from a few tens of meters (e.g. when filming a music contest) up to 1 kilometer (cyclocross competition). Fibre optics routing supported with adapter can cover the large distances.
- **R-CAP-3 Synchronisation of camera streams:** the various streams resulting from different camera set-ups may have different latencies, each omnidirectional system typically has its own latency constraint. These streams need to be resynchronised, this will happen in the Cinegy server (live production tool)
- **R-CAP-4 Cabling infrastructure:**
 - o **Power:** Power generator and power cables
 - o **Ethernet cables:** for larger distances fiber optics routing can be used with adapters
 - o **Fibre sets:** Connection cameras/capture server to control car: nevion fibre sets with break-out boxes (one in the field, one in the control car) carrying 4x HD-SDI and 1x Ethernet. The Ethernet connection can also be used to remotely control the Studio. One cameras via RESTful APIs. Different Nevion fibre sets can connect different cameras, such that cameras can be sufficiently distributed. An example cabling scheme based on the Nevion fiber set is illustrated in Figure 17.
 - o **Triax cables:** to connect the directive (tv) cameras with the respective CCU's in the control car.
 - o **BNC HD-SDI cables:** HD-SDI cabling of up to 130m from Studio. One semi-omnidirectional cameras to their capture and processing server. (depending on

cable quality - here supposed Belden 1694A ³, with thicker high quality HD-SDI cables, or multiple cables per camera, larger distances can be covered)

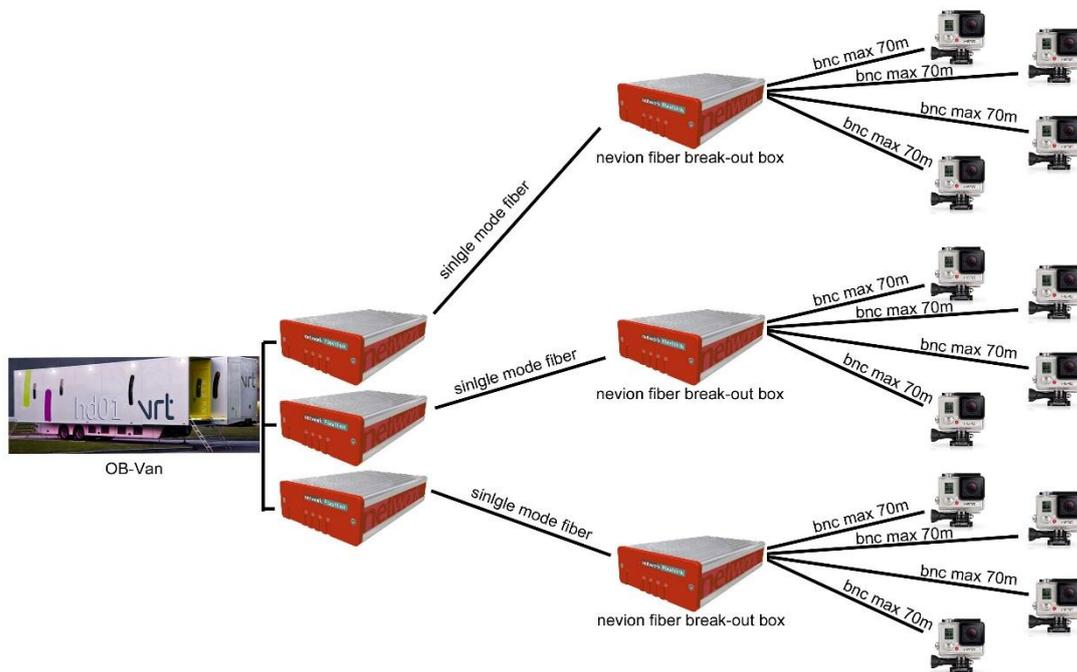


Figure 17: Connecting cameras on the cyclocross circuit with the OB-VAN over fibre (Nevion fibre set with break-out boxes).

For live pilot 3, the following specific requirements regarding capturing apply:

- **R-CAP-5 Capturing format:** at least 4K 360° video in equirectangular format, at 25fps or higher.
- **R-CAP-6 Timestamping:** timestamping for synchronisation with other cameras and audio.
- **R-CAP-7 Remote operation:** remote operation of camera shading, recordings, streaming.
- **R-CAP-8 Ambisonic audio capturing:** regarding audio, use the ambisonic capabilities of the orah camera, mixed with lavalier microphones in each actor/performer.
- **R-CAP-9 Non-intrusive camera rig:** keep the camera rig as little intrusive as possible, so both the directive cameras and 360° rig can have the most flexible space to work in.

3.2.3.2. STITCHING

The general specifications for stitching are as follows:

³ <http://www.belden.com/techdatas/english/1694a.pdf>

- **R-STI-1 Stitching output formats:** Be able to output real-time 4K 30 fps video streams, either compressed (H.264 RTMP) or uncompressed (provided by EDM's stitching server, or Orah's Stitching Box HDMI out, or Vahana VR output ports)
- **R-STI-2 VahanaVR stitching:** one Vahana VR instance per GoPro/Elmo rig, about 300ms of latency
- **R-STI-3 Orah stitching:** One Orah Stitching Box per Orah 4i camera, 3 seconds of latency
- **R-STI-4 Studio.One stitching:** the AZilPix Studio.One server performs 360 stitching in real-time with UHD 30fps output tested for live 360 streaming before. Camera to screen latency is in the order of 2 to 3 frames.

As far as stitching is concerned, live pilot 3 poses no additional requirements.

3.2.3.3. LIVE PRODUCTION TOOL

In the live production tool, the different captured omnidirectional and directive content streams are ingested and synchronized with each other. From these input streams the director creates a multi-device experience consisting of one or more omnidirectional videos, directive shots and interaction points for the HMD viewing experience in the form of graphical icons or portals allowing the viewer to interact with the content. As a background reference, we refer to Annex 1 of the first D2.3 deliverable instance for preliminary requirements for live production tooling.

- **R-PROD-1 Ingest:** this describes the ingest side of the Live Production block. The possible sources include streams from omnidirectional cameras, streams from directive cameras, and optionally video files.
 - HD SDI signals to be converted to RTMP
 - Compatibility with VahanaVR and Orah4i RTMP streams
 - RESTful API to control the Studio.One camera
 - Configurable number of sources (RTP, RTMP, files)
 - RTMP to RTP (Cinegy Live format) and RTSP (live preview) rewrapping support
- **R-PROD-2 Synchronisation of streams (Cinegy Transport):** configurable service that lists the RTMP streams to be processed and specifies the required delay. Cinegy Transport can add a per-stream delay by rewriting the time stamp with a common value, based on UTC added with a preconfigured (measured) delay. Optionally the stream can also be physically delayed – i.e. re-broadcasted with defined delay to ensure both logical (metadata) and physical (delivery) streams are in sync.
- **R-PROD-3 Live content edition (Cinegy Live):** the live production operator workstation will have Cinegy Live VR application running that is Windows OS-based and provides the following functionality:
 - **Sources:** display of all incoming streams
 - **Preset 360 scene composition:** One or more preconfigured 360 scene compositions (i.e. one or more omnidirectional videos and directive shots with interaction points) that are prepared beforehand and are available as ImmersiaTV scene XML format. This file initializes the live content delivery. The format is defined in D3.1. The operator can at least:
 - define a set of graphical icons and portals in the scene
 - select portions of omnidirectional and directive video for use in portals
 - assign position, size, shape and user/world reference of the insert

- assign input streams to graphical icons representing cameras
- define user interaction patterns e.g. switch from one scene composition to another one, triggered by user input
- **Scene changes:** the director can define required scene changes which activates an automatic scene update action, e.g. removing a camera icon, changing the actual source for a portal, .. . An updated ImmersiaTV scene XML is generated in real-time and in sync with the media streams.
- **Transitions:** the director can activate a transition between omnidirectional and/or directive streams. A set of quick access scene transitions is available. Transitions typically apply for the *director's view* mode, but can also relate to the entry view of the *user exploration* mode.
- **Live Preview:** the director can locally preview scene modifications in real-time (or sufficiently low latency).
- **Content for second screen (tablet):** the director can preconfigure and change the second screen experience, e.g. a mosaic consisting of a number of camera viewpoints that can be activated
- **Mixing TV audio and omnidirectional audio into a stereo stream:** this advanced feature will be included in pilot 3, first tests can be performed during pilot 2.
- **R-PROD-5 Distribution:** collection and packaging of all streams and metadata to send towards MPEG-DASH streaming server that
 - accepts RTMP streams as live sources
 - supports MPEG-DASH events mechanism
 - allows scene profile modifications and scene updates
 - (optionally) performs external encoding to package the incoming streams as segmented MP4 files. However this packaging can be performed by Cinegy Transport in the final version.

For **pilot 2**, we will primarily focus on providing the *director's choice* viewing scenario, and on achieving **good interface usability** of the corresponding live editor software. More advanced features such as the interactive *user exploration mode* for HMD and tablet, using of omnidirectional audio and mixing with TV audio, sharing of content between tablet and HMD will be dealt with in pilot 3.

For **pilot 3**, live production tools should give the operator more flexibility in defining the user experience on-the-fly, allowing quick reaction to last-minute changes and events.

The main novelty of pilot 3 is to introduce user exploration mode experience. This implies the ability to select scenes containing non-live content (replays, highlights) that can be recorded parts of the current live event or extra content prepared beforehand. The following additional requirements apply:

- **R-PROD-6 Individual scene configuration:** placement of portals individually for each omnidirectional camera. Each camera that produces a live stream is considered as a separate scene that can be activated by the user or suggested by the live production tools operator. The operator should be able to define the location of the portals for each

- scene individually based on the content (for example, to avoid portals hiding some important view/object). When a new scene is activated the locations of the portals and content should be updated according to the settings for the new scene.
- **R-PROD-7 Recording of replays:** recording highlights and dynamically adding them to the list. The live production tool should allow recording of the current live event and should enable the recorded content for re-use as VOD content within the same session. The operator should be able to select recorded segments of the current live session, to specify possible interaction points for this new scene and make it available for the player users.
 - **R-PROD-8 Publishing of replays:** publishing a list of additional offline content - replays, pre-recorded content. The live production tool should allow the operator to define VOD content as accessible scenes additionally to the live content. Non-live media should be either recorded segments of the current live session or external additional content.
 - **R-PROD-9 Generation of ambisonic sound:** to enhance the immersive experience of the user, audio for each scene should optionally include ambisonic audio from the location. Audio should be captured by the camera and be available for the playback in the player. Optionally live production tools should allow mixing the audio from an additional source (directional TV commentator voice, for example) into the ambisonic audio for each scene.

3.2.3.4. ENCODING

The encoding process has to be performed inline within the live production pipeline. As a general rule, codec parameters (video format, bit rate, frame rate) are determined by the capabilities of the reception device, such as a typical HMD, a tablet or a smartphone. The goal in ImmersiaTV is to minimize the number of transcoding steps needed. Where possible, the influence of each encoding step on the quality should be minimized. Note that at capture stage, for the live pilot, the codec parameters of the used cameras prevail, but as said, the main bottleneck are the reception device capabilities.

The encoding and decoding subsystem is designed to compress the video stream in order to reduce the required bandwidth for transmission (broadcasting) and the required disk space to store the content. The encoder takes as an input the stitched omnidirectional video stream from the capture system, remaps it (if required) to a particular projection type and sends the stream to the distribution system.

For Pilot 2 an AVC-based encoding strategy will be adopted as this is fully supported by the devices used for playout. In parallel, an HEVC based encoding strategy will be investigated in view of improving visual quality and reducing required bandwidth. Due to limitations on the playout side, this coding strategy is not anticipated to be fully supported in Pilot 2. The detailed upstream and downstream external requirements regarding an AVC and HEVC encoding strategy can be found in Deliverable 3.1 including detailed encoding design requirements.

- **R-ENC-1 Upstream external requirements for AVC-based solution:** fully integrated with VahanaVR platform, requirements are those specified within VahanaVR

- **R-ENC-2 Downstream external requirements:** determined by capabilities of reception device
 - o **Supported codec:** H.264/AVC
 - o **Codec parameters:** bit rate, frame rate, frame size, chroma sub-sampling
 - o **Supported projections:** Cubic 2x3 rotated
- **R-ENC-3 Encoding design requirements**
 - o Real-time hardware (GPU: NV NVIDIA GeForce GTX 1080 or similar) supporting encoding of single video stream
 - o Ubuntu Linux Operating System
 - o YUV-based encoder input
 - o Input interface: on same platform as the stitching software
 - o Output interface: same as platform hosting the stitching software
 - o Per encoding session configuration

In pilot 3, a HEVC-based encoding strategy will be deployed. The encoding/decoding module will also take into account QoE data (in particular, objective quality metric and visual attention) to adjust the encoding settings and performance. The additional requirements for live pilot 3 are:

- **R-ENC-4 Encoder Upstream external requirements for HEVC-based solution:** fully integrated with VahanaVR platform, requirements are those specified within VahanaVR
- **R-ENC-5 Encoder Downstream external requirements:** determined by capabilities of reception device
 - o **Supported codec:** H.265/HEVC
 - o **Supported projections:** Equirectangular and Cubic 2x3 rotated
- **R-ENC-6 Encoding design requirements:** similar as for live pilot 2
- **R-ENC-7 Compression requirements:** the encoder is to compress 4k omnidirectional video at up to 30 frames per second
- **R-ENC-8 Decoder requirements:** the decoder is to decompress 4k omnidirectional video at up to 30 frames per second
- **R-ENC-9 Objective QoE metric:** the encoder takes as an input an objective QoE metric from the QoE module
- **R-ENC-10 QoE Visual attention info:** the encoder takes as an input from the QoE module visual attention information
- **R-ENC-11 QoE adjusted Encoding parameters:** the encoder uses the objective QoE metric to adjust the encoding parameters
- **R-ENC-12 QoE adaptive pre-filtering:** the encoder uses visual attention information to apply adaptive pre-filtering

3.2.3.5. DISTRIBUTION

The distribution encoder is responsible to collect all streams and metadata produced in Live Production, and perform real-time packaging and delivery of these streams through the internet up to the consumers' local network, where synchronized multi-device playout will take place. Streaming delivery is based on the MPEG-DASH standard.

The distribution encoder-related requirements for pilot 2 are:

- **R-DIST-1 Live Preview (operator):** different RTSP streams needs to be provided to preview contents in ImmersiaTV preview player.

- **R-DIST-2 MPEG-DASH streams:** live streams should be available on a server with metadata signaling the different contents to show at specific times.
- **R-DIST-3 Metadata updates:** the metadata XML file should be accessible on the same server with the media content. The files are pointed by a relative path to itself. The metadata XML file will be refreshed in a timely manner in order to be consumed by the ImmersiaTV player such that the changes are reflected before they happen (future time). This means that the metadata refresh interval should be less than the delay introduced by the distribution pipeline.
- **R-DIST-4 Live transitions:** transitions will be selected from a list, as indicated in the XML metadata file. The transitions will be produced in real time in the ImmersiaTV player.

Delivery and reception module must be compatible with client. The additional requirements for live pilot 3 are:

- **R-DIST-5 Social networks binding:** In order to publish stored movement around the 360° scene on social media portals (e.g. Facebook) this module needs to support this functionality.
- **R-DIST-6 Ambisonics audio packaging and distribution in a compatible MPEG-DASH format:** ambisonics audio delivered from production tools to the client will need adaptation of MPEG-DASH format, i.e. packaging of ambisonics b-format to be streamed and adapted in order to deliver 4 channels that can be mapped into a 360° scene.
- **R-DIST-7 H.265/HEVC MPEG-DASH for native player media consumption:** transcode and produce h.265 streams for higher resolutions, this will result in lower bitrates and higher qualitative results.
- **R-DIST-8 H.264/AVC and H.265/HEVC mixed MPEG-DASH distribution:** both codecs will be allowed per one MPEG-DASH stream, so the web player will reproduce the H.264/AVC stream and the native player can get advantage of the more optimized compression of H.265/HEVC.
- **R-DIST-9 Automatic conversion from equirectangular to cubemap representation:** to avoid redundancy in the 360) video distribution, a more optimized representation will be produced from equirectangular.

Similar to R-REC-26 (off-line pilot 3), ambisonics audio parsing in both web player and native player is a distribution requirement for live pilot 3; the MPEG-DASH demuxer and decoder must process this streamed format.

3.2.3.6. RECEPTION, INTERACTION AND DISPLAY

This component deals with the reception of the streams from the Internet, their redistribution in the home network, and the integration with interactive input of the viewer. The ImmersiaTV Interaction and Display software (ImmersiaTV player) runs on the receiver devices, and

integrates the audio, video and data streams into a coherent omnidirectional scene by parsing the user input and adapting the expected output appropriately. The specific requirements of Reception, Interaction and Display in the context of pilot 2 are:

- **R-REC-1 Master device:** TV should be a master device for synchronization in LAN.
- **R-REC-2 Adaptive streaming:** the ImmersiaTV player should adaptively select stream resolution and bitrate provided by the MPEG-DASH server, taking into account the hardware capabilities of playout device and available network bandwidth.
- **R-REC-3 Refresh:** Player should refresh scene layout (XML metadata refreshing), taking into account changes on the server side.
- **R-REC-4 Transition implementation:** Player should implement a list of possible transitions for opening and closing of portals or graphical icons.

Pilot 3 extends the functionalities for the live scenario. One of the changes will be additional offline content such as replays that is available next to the live streams. This requires the following new functionalities:

- **R-REC-5 Displaying list of additional offline content**
- **R-REC-6 Mixing live and offline content**
- **R-REC-7 Pausing, resuming, synchronizing to live**
- **R-REC-8 Notifying about new additional content available**
- **R-REC-9 Highlighting of interesting viewpoints recommended by director:** no direct push, user is free to select suggested viewpoint or not.
- **R-REC-10 Storing path of user head movement in order to publish on social media:** during live streaming, the user should have the possibility to store his or her actual view of the 360° scene and then to publish a 360° video with the recorded path (based on tracked head movement) on social media portals which support it (e.g. Facebook)
- **R-REC-11 Support for hotspots with tooltips distribution:** a new kind of interactivity based on portals with additional information based on actual camera locations. This requirement is related to *R-PROD-6 Individual scene configuration*.

3.2.3.7. QUALITY OF EXPERIENCE

The QoE component aims to assess the perceived quality of the immersive experience at the end user's side, and to advise and steer encoding parameter adjustment to improve quality whenever necessary. To be effective, quality measurements should be well-correlated to human perception. The QoE metric takes into account distortion and content features (server side), and also considers the effect of delay in viewing experiences (end user side). The specific requirements of the QoE module for pilot 2 can be listed as follows:

- **R-QoE-1 Logging information:** the QoE module requires a number of logging information as input to evaluate quality (a list of QoE parameters has to be provided).

- **R-QoE-2 Server implementation:** the QoE module will be implemented on a server. This can be the same server as where the Encoder is running or a separate server close to the Encoder.
- **R-QoE-3 Timing signals:** at the client side, access to timing signals (time stamps, net clock data) is required (before and after delay) to extract start and length of the delay (stalling).
- **R-QoE-4 Side channel:** a side channel between clients and the server is required to deliver necessary data (e.g. delay data) for QoE optimisation
- **R-QoE-5 Feedback communication:** a communication channel between QoE and Encoder is required to send QoE feedback for steering encoding parameters. The feedback will be used in Phase 3 (pilot 3): an Encoder Feedback module establishes a connection with Encoder in order to transmit the data computed in the QoE Analysis to the Encoder sub-system.

For pilot 3, the requirements regarding the communication channel between the QoE module and the DASH server is also further elaborated:

- **R-QoE-6 Communication channel between QoE and DASH server:** to acquire HEVC bit streams and store them for decoding. If the videos are not already stored on the QoE server, the QoE module needs to initiate DASH sessions with the CDN, acquire an HEVC bit stream and store this bit stream for off-line decoding and image quality assessment.

3.3. Refined off-line

3.3.1. Production scenario

The off-line pilot 3 scenario defined in Section 2.3 requires a production workflow that follows the following professional user scenario:

Professional User scenario:

The media department of a broadcast company was very pleased with the multi-device VR documentary of the young football kid David of FC Porto, and has commissioned a new whodunnit fiction programme, also to be produced in VR.

A producer (Matt) has received the order to create the whodunnit on multiple devices using omnidirectional content, with the request of having dedicated user exploration with the second screen devices without jeopardizing the storytelling on television. Matt decides to apply exploration mode to enhance the whodunnit with subjective storytelling elements, and asks scenarist Louise to write a script. Louise uses the “exploration mode” scripting template in her scripting tool to elaborate the scenario breakdown, the main characters and events, and detailed script. The template allows her to indicate how the content on the different devices follow on and interact with each other along the storyline. She decides not to opt for branched storytelling in this case.

During capturing, the director’s assistant Julia can link the captured shots for the television and HMD to the structured script, and annotate captured content. The audience engineer on set makes sure two different audio tracks are recorded: one binaural to capture the generic

ambience, and one directive to capture the noises and voices. There is no need to capture directive and VR content simultaneously, which will improve the cinematographic quality of the shooting.

After capturing, editor Tom creates a new project selecting “exploration mode” as project mode setting. This imposes specific timeline behaviour for TV, HMD and tablet content, e.g. TV content and HMD content cannot overlap. For each of the intended interactions between HMD and tablet, Tom specifies the specific user action in the HMD as conditional event upon which events on the tablet can be triggered, e.g. popping up of extra information which is added as a still picture on the timeline. He specifies the areas of interaction in the HMD, that activates tablet events or portals inside the HMD with additional content. Tom sets a specific minimum and maximum duration for each of the break-out segments in exploration mode.

In audio post production, spatial and binaural audio are mixed and added to the project timeline as well.

Once the post production is finished, Tom can visualise one last time the final edited content in the end user devices, and test whether the interaction mechanisms within and between devices work as expected. By publishing the exploration mode-based project, the ImmersiaTV player can play out the whodunnit to the end user using a multi-device setup.

3.3.2. Consumption scenario

End User scenario:

Emma, Liam, Oliver and Charlotte are spending the evening together and decide to watch “The Hollow Man”, a multi-device whodunnit experience. They have 2 HMD devices and one tablet at their disposal. Emma (Observer) prefers to watch the television, and Oliver (Assistant) steps forward as would-be detective playing the tablet. Liam and Charlotte are the two Explorers: they will explore the 360° scenes, activate portals and pick objects to be examined by Oliver.

The prologue and introduction scene is enjoyed by all four on television, before switching to exploration mode. Oliver has selected the HMD view of Liam to be replicated on television to start. Liam and Charlotte are looking around through the eyes of doctor Walnut, fairly quickly discovering the stack of books, opening a portal showing Mr. Ample organizing the books in his peculiar way, by page number. Then Charlotte discovers the food tray, now glowing up inside the HMD. She also discovers the whisky glass, and Emma asks Liam to look for the glass as well so she can see it on the television. Meanwhile Oliver is gathering the pieces of evidence on the detective’s notebook on his tablet. He asks the Explorers if there is more evidence in the room. Their answer is negative, but they are overlooking the new book in the centre of the desk.

Now, exploration mode is off again and the story continues on television where the book “The Hollow Men” is now revealed. Oliver still keeps an eye on the detective’s notebook, and discovers additional information about the book.

In the subsequent exploration modes, Emma asks Oliver to switch to Charlotte as main Explorer as she seems better at looking around while observing the crime scene. More portals and extra information is popping up, triggering a lively discussion between the 4 viewers. Before entering the final scene on TV, they have an animated discussion on who finally did murder Mr. Ample. It seems to be Eleanor: they have seen her putting arsenic on the top corners of every page of the book, and is holding a knife in her hand at the end. Did she have an accomplice or not?

The truth is finally revealed by inspector Legrand: it was indeed Eleanor.

In the aftermath of watching the whodunnit, the discussion continues. Emma is still confused that neither Mr. or Mrs. Smith had something to do with murder. They all gather around Oliver's detective notebook to further discuss the denouement.

3.3.3. Technical specification and requirement analysis

This section defines the technical specification and software requirements for the off-line pilot 3 which introduces Exploration Mode. For the modules Capturing, Stitching, Encoding, Distribution and Quality of Experience, the requirements for live and off-line pilot 3 are the same. These requirements are already listed and described in Section 3.2.3. Specific new requirements are introduced in this section of Editing & Compositing and Reception/Interaction/Display.

3.3.3.1. CAPTURING

New requirements: R-CAP-5 to R-CAP-9.

3.3.3.2. STITCHING

No new requirements for pilot 3.

3.3.3.3. EDITING AND COMPOSITING

The new concepts related to Exploration Mode as applied in the script and the new features of the player need support on the content creation side. The content creator must be able to prepare content that fully utilizes the defined functionalities. As a main novelty of pilot 3 it should be possible to define scenes, where exploration mode is available. The length of such a scene is the maximum time the user can watch it, but there should be an option to leave earlier and progress with the story.

- ***R-EDIT-20 Selecting Exploration Mode for the project settings***
- ***R-EDIT-21 Creating user action for going from exploration mode to the next scene***

In exploration mode there are more advanced ways of content playout and these should be defined during production. Some video clips may start at the moment of activating a portal by the user. This means that they are not in sync with other content.

- ***R-EDIT-22 Adding unsynchronized content (to be started on user action)***

The content creator should be able to plan extended interactions between devices. For example when the HMD user selects and watches additional video, additional content defined by the creator (e.g. as an icon) may appear on a tablet. It may stay visible till the end of the content.

- ***R-EDIT-23 Defining interactions between devices***: an event on one device triggers an action on another
- ***R-EDIT-24 Defining user events that need to be stored for actions in following scenes***: this requirement refers to actions performed by the viewer that have to be remembered so the additional info on the tablet can still be displayed even after scene changes. The content creator should be able to indicate which user actions have permanent effect.

The script expects such icons or other additional information to be presented on the tablet in a specific mode ('detective's notebook'). A dedicated tool to design such a view may be needed.

- ***R-EDIT-25 Designing a customized tablet view (e.g. list of objects found)***

3.3.3.4. ENCODING

New requirements: R-ENC-4 to R-ENC-12.

3.3.3.5. DISTRIBUTION

New requirements: R-DIST-5 to R-DIST-9.

3.3.3.6. RECEPTION, INTERACTION AND DISPLAY

In off-line pilot 3, the player also introduces Exploration Mode. The new concepts related to this new mode have been explained in Section 2.3.2.

The corresponding requirements are the following:

- ***R-REC-12 Enabling Exploration Mode on the client***
- ***R-REC-13 Different devices can display different non-synchronized content***
- ***R-REC-14 TV displays VR view from HMD***
- ***R-REC-15 Sharing the path of head movement of a user.*** This is required for the VR view on television (screen cast).
- ***R-REC-16 HMD may display: Video or Static image (e.g. map)***
- ***R-REC-17 All HMD displays should be synchronized in time***
- ***R-REC-18 Support for video looping on all devices***

R-REC-19 Tablet in Exploration mode enables to show additional information. This view may contain the list of objects, their textual descriptions or photos. The time duration of the scene in Exploration Mode is limited. This means a user has only a (pre)defined time for exploring the scene:

- ***R-REC-20 HMD content is limited in time, then user is moving to another scene***

The Exploration Mode needs extended interactivity, which means that the story and scenes should follow the user's actions. These actions trigger playback of specific content on specific devices. The user should have the possibility to manually control the rhythm and speed of the experience by pausing and resuming the exploration experience.

- ***R-REC-21 Client extends interactivity between devices***
- ***R-REC-22 Functionality of pausing and resuming of the content***
- ***R-REC-23 Functionality of jumping to another scene basing on users actions***

During Exploration Mode, some objects in the HMD scene can highlight when a user is focusing on them and they can be selected by clicking the HMD button. Then the object appears on the Tablet list with their additional description.

- ***R-REC-24 Highlighting areas where objects are located in the scene (HMD)***
- ***R-REC-25 Selecting objects from the scene in HMD to be displayed on tablet.***

Pilot 3 introduces two important requirements related to audio, which were not fully covered in the previous pilots. These requirements are not related to Broadcast mode or Exploration mode, but should be supported in general. The first one is to support ambisonics spatial audio and extend user experience.

- ***R-REC-26 Supporting ambisonics sound***

The second one is for synchronization between devices running in different networks - for instance TV is traditional broadcasted (terrestrial, cable or satellite television), and content for tablet and HMD is delivered via Internet. The system should offer ways to synchronize streams in both networks using audio.

- ***R-REC-27 Synchronization across devices based on audio***

3.3.3.7. QUALITY OF EXPERIENCE

New requirements: R-QoE-6 and R-QoE-7 (see also Section 3.2.3.7).

4. OVERVIEW OF REQUIREMENTS AND SPECIFICATIONS

4.1. Off-line documentary

From the analysis of the user scenarios in previous sections, as well as the general requirements of D2.1 and D2.2, here we have the resulting list of specific requirements categorized according to the different stages of the workflow.

Story preparation

R-STORY-1 The content creator can create the main storyline

R-STORY-2 The content creator can define the main and side characters

R-STORY-3 The content creator can define the detailed story structure

R-STORY-4 The content creator can define the sub-storylines for TV, tablet and HMD

R-STORY-5 The content creator can define the user interaction design

(For each scene:)

R-STORY-6 The content creator can define the multi-platform logic

R-STORY-7 The content creator can define the viewer perspective(s)

R-STORY-8 The content creator can define the detailed script

R-STORY-9 The content creator can define if it uses Omnidirectional and/or directive content

R-STORY-10 The content creator can define the viewing angle

R-STORY-11 The content creator can define the interaction points: portal, AR object, caption, graphics, ...

R-STORY-12 The content creator can define transition between scenes

R-STORY-13 The content creator can specify use of audio for guidance and transitions

R-STORY-14 The content creator can indicate use of camera movements

R-STORY-15 The content creator can indicate forced exploration mode where applicable

R-STORY-16 The content creator can save resulting format script as master template

Production preparation

R-PREPROD-1 The content creator can add on-location research material as placeholder in format script

R-PREPROD-2 The content creator can perform first VR preview of relevant scenes

R-PREPROD-3 The content creator can define the shooting plan

R-PREPROD-4 The content creator can define the VR/directive capturing strategy

Edition and Compositing

R-EDIT-1 The content creator can visualize the raw material across the different end-user devices

R-EDIT-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

R-EDIT-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

R-EDIT-4 The content creator can use Windows and OS X

R-EDIT-5 The content creator can use of an advanced mode in a compositing software (Nuke, Adobe After Effects)

R-EDIT-6 The content creator can introduce interactivity within the editor timeline through *conditional transitions* between shots and scenes

R-EDIT-7 The content creator can select, within the editor timeline, which video assets are visible within the TV, the tablet and the HMD

R-EDIT-8 The content creator can also create *ImmersiaTV scene typologies*, i.e., interaction between devices, through *conditional transitions* within the editor timeline

R-EDIT-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

R-EDIT-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.

R-EDIT-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

R-EDIT-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.

R-EDIT-13 The content creator will be able to visualize transitions and interactive transitions will be visible within the editing software

R-EDIT-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.

R-EDIT-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:

R-EDIT-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).

R-EDIT-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.

R-EDIT-18 It will be possible to define a label specifying the destination for each sequence

R-EDIT-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:

Content Playback

R-PLAY-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.

R-PLAY-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 signaled within the stream itself, or it might be transmitted using a parallel communication channel.

R-PLAY-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.

R-PLAY-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).

R-PLAY-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).

R-PLAY-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.

R-PLAY-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.

R-PLAY-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.

R-PLAY-9 Basic audio control in the end-user devices

R-PLAY-10 Real time communication channel between devices: It will be needed to send messages from one device to another

R-PLAY-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.

R-PLAY-12 The end-user can capture screen casts and share them with other devices

R-PLAY-13 The end-user can capture screen casts and share them through social media

4.2. Live event

From the analysis of the production and consumption scenarios in previous sections, as well as the general requirements of D2.1 and D2.2, the identified technical specifications and software requirements for the live pilot are listed in Section 0 and summarized below.

Capture

R-CAP-1 Omnidirectional and directional cameras: Orah 4i, GoPro rigs and EDM Studio. One cameras, Grass Valley LDK 8000 cameras for directional recording.

R-CAP-2 Camera distances: from a few tens of meters up to 1 kilometre.

R-CAP-3 Synchronisation of camera streams (in Cinegy server): different camera set-ups may have different latencies and need to be resynchronised.

R-CAP-4 Cabling infrastructure: power, Ethernet cables, fibre set connection to the control car carrying 4x HD-SDI and 1x Ethernet, triax cables, BNC HD-SDI cables up to 130m

Additional requirements for (live) pilot 3:

R-CAP-5 Capturing format: at least 4K 360° video in equirectangular format, at 25fps or higher.

R-CAP-6 Timestamping: timestamping for synchronisation with other cameras and audio.

R-CAP-7 Remote operation: remote operation of camera shading, recordings, streaming.

R-CAP-8 Ambisonic audio capturing: regarding audio, use the ambisonic capabilities of the Orah camera, mixed with Lavalier microphones in each actor/performer.

R-CAP-9 Non-intrusive camera rig: keep the camera rig as little intrusive as possible, so both the directive cameras and 360° rig can have the most flexible space to work in.

Stitching

R-STI-1 Stitching output formats: be able to output real-time 4K 30 fps video streams, either compressed (H.264 RTMP) or uncompressed

R-STI-2 VahanaVR stitching: one Vahana VR instance per GoPro/Elmo rig, about 300ms of latency

R-STI-3 Orah stitching: one Orah Stitching Box per Orah 4i camera, 3 seconds of latency

R-STI-4 Studio.One stitching. The AZilPix Studio.One server performs 360 stitching in real-time with UHD 30fps output. Camera to screen latency is in the order of 2 to 3 frames.

Live production tool

R-PROD-1 Ingest: the possible sources include streams from omnidirectional cameras, streams from directive cameras, and optionally video files.

- HD SDI signals to be converted to RTMP
- Compatibility with VahanaVR and Orah4i RTMP streams
- RESTful API to control the Studio.One camera
- Configurable number of sources (RTP, RTMP, files)
- RTMP to RTP (Cinegy Live format) and RTSP (live preview) rewrapping support

R-PROD-2 Synchronisation of streams (Cinegy Transport): configurable service that lists the RTMP streams to be processed and specifies the required delay.

R-PROD-3 Live content edition (Cinegy Live): the live production operator workstation will have Cinegy Live VR application running that is Windows OS-based and provides the following functionality:

- Sources: display of all incoming streams
- Preset 360 scene composition: One or more preconfigured 360 scene compositions. The operator can at least:
 - define a set of graphical icons and portals in the scene
 - select portions of omnidirectional and directive video for use in portals
 - assign position, size, shape and user/world reference of the insert
 - assign input streams to graphical icons representing cameras
 - define user interaction patterns e.g. switch from one scene composition to another one, triggered by user input

- Scene changes: the director can define required scene changes which activates an automatic scene update action
- Transitions: the director can activate a transition between omnidirectional and/or directive streams. A set of quick access scene transitions is available.
- Live Preview: the director can locally preview scene modifications in real-time (or sufficiently low latency).
- Content for second screen (tablet): the director can preconfigure and change the second screen experience
- Mixing TV audio and omnidirectional audio into a stereo stream (pilot 3).

Additional requirements for live pilot 3:

R-PROD-5 Distribution: collection and packaging of all streams and metadata to send towards MPEG-DASH streaming server that

- accepts RTMP streams as live sources
- supports MPEG-DASH events mechanism
- allows scene profile modifications and scene updates

R-PROD-6 Individual scene configuration: placement of portals individually for each omnidirectional camera.

R-PROD-7 Recording of replays: recording highlights and dynamically adding them to the list.

R-PROD-8 Publishing of replays: publishing a list of additional offline content - replays, pre-recorded content.

R-PROD-9 Generation of ambisonic sound

Encoding

R-ENC-1 Upstream external requirements for AVC-based solution: fully integrated with VahanaVR platform, requirements are those specified within VahanaVR

R-ENC-2 Downstream external requirements: determined by capabilities of reception device, supported codec H.264/AVC, supported projection Cubic 2x3 rotated

R-ENC-3 Encoding design requirements: real-time hardware supporting single video stream encoding, Ubuntu Linux OS, YUV encoder input, input and output interface on stitching software host platform, per encoding session configuration

Additional requirements for (live) pilot 3:

R-ENC-4 Encoder Upstream external requirements for HEVC-based solution: fully integrated with VahanaVR platform, requirements are those specified within VahanaVR

R-ENC-5 Encoder Downstream external requirements: determined by capabilities of reception device

Supported codec: H.265/HEVC

Supported projections: Equirectangular and Cubic 2x3 rotated

R-ENC-6 Encoding design requirements: similar as for live pilot 2

R-ENC-7 Compression requirements: the encoder is to compress 4k omnidirectional video at up to 30 frames per second

R-ENC-8 Decoder requirements: the decoder is to decompress 4k omnidirectional video at up to 30 frames per second

R-ENC-9 Objective QoE metric: the encoder takes as an input an objective QoE metric from the QoE module

R-ENC-10 QoE Visual attention info: the encoder takes as an input from the QoE module visual attention information

R-ENC-11 QoE adjusted Encoding parameters: the encoder uses the objective QoE metric to adjust the encoding parameters

R-ENC-12 QoE adaptive pre-filtering: the encoder uses visual attention information to apply adaptive pre-filtering

Distribution

R-DIST-1 Live Preview (operator): different RTSP streams needs to be provided to preview contents in ImmersiaTV preview player.

R-DIST-2 MPEG-DASH streams: live streams should be available on a server with metadata signaling the different contents to show at specific times.

R-DIST-3 Metadata updates: the metadata XML file should be accessible on the same server with the media content (relative paths). The metadata refresh interval should be less than the delay introduced by the distribution pipeline.

R-DIST-4 Live transitions: transitions will be selected from a list, as indicated in the XML metadata file. The transitions will be produced in real time in the ImmersiaTV player.

Additional requirements for (live) pilot 3:

R-DIST-5 Social networks binding: In order to publish stored movement around the 360° scene on social media portals (e.g. Facebook)

R-DIST-6 Ambisonics audio packaging and distribution in a compatible MPEG-DASH format

R-DIST-7 H.265/HEVC MPEG-DASH for native player media consumption

R-DIST-8 H.264/AVC and H.265/HEVC mixed MPEG-DASH distribution

R-DIST-9 Automatic conversion from equirectangular to cubemap representation

Reception, Interaction and Display

R-REC-1 Master device: TV should be a master device for synchronization in LAN.

R-REC-2 Adaptive streaming: the ImmersiaTV player should adaptively select stream resolution and bitrate provided by the MPEG-DASH server, taking into account the hardware capabilities of playout device and available network bandwidth.

R-REC-3 Refresh: Player should refresh scene layout (XML metadata refreshing), taking into account changes on the server side.

R-REC-4 Transition implementation: Player should implement a list of possible transitions for opening and closing of portals or graphical icons

Additional requirements for live pilot 3 are:

R-REC-5 Displaying list of additional offline content

R-REC-6 Mixing live and offline content

R-REC-7 Pausing, resuming, synchronizing to live

R-REC-8 Notifying about new additional content available.

R-REC-9 Highlighting of interesting viewpoints recommended by director

R-REC-10 Storing path of user head movement in order to publish on social media

R-REC-11 Support for hotspots with tooltips distribution: a new kind of interactivity based on portals with additional information based on actual camera locations. This requirement is related to *R-PROD-6 Individual scene configuration*.

Quality of Experience

R-QoE-1 Logging information: input to evaluate quality.

R-QoE-2 Server implementation: the QoE module will be implemented on a server (same as Encoder or separate server).

R-QoE-3 Access to timing signals at the client side: time stamps, net clock data to extract start and duration of delay (stalling).

R-QoE-4 Side channel between clients and server to deliver necessary data (e.g. delay data) for QoE optimisation

R-QoE-5 Feedback communication between QoE and Encoder to send QoE feedback for steering encoding parameters. The feedback will be used in Phase 3 (pilot 3).

The additional requirements for (live) pilot 3 are:

R-QoE-6 Communication channel between QoE and DASH server: to acquire HEVC bitstreams and store them for decoding.

4.3. Exploration mode

The identified technical specifications and software requirements that are specific are listed in Section 3.3.3 and summarized below. Only the requirements for Editing and Reception/Interaction/Display are listed here, for the other modules (capturing, stitching, encoding, distribution, QoE) the off-line pilot 3 requirements are similar to the live pilot 3 reqs.

Editing

R-EDIT-20 Selecting Exploration Mode for the project settings

R-EDIT-21 Creating user action for going from exploration mode to the next scene

R-EDIT-22 Adding unsynchronized content (to be started on user action)

R-EDIT-23 Defining interactions between devices: an event on one device triggers an action on another

R-EDIT-24 Defining user events that need to be stored for actions in following scenes

R-EDIT-25 Designing a customized tablet view (e.g. list of objects found)

Reception, Interaction and Display

R-REC-12 Enabling Exploration Mode on the client

R-REC-13 Different devices can display different non-synchronized content

R-REC-14 TV displays VR view from HMD

R-REC-15 Sharing the path of head movement of a user: VR view on television (screen cast).

R-REC-16 HMD may display: Video or Static image (e.g. map)

R-REC-17 All HMD displays should be synchronized in time

R-REC-18 Support for video looping on all devices

R-REC-19 Tablet in Exploration mode enables to show additional information. This view may contain the list of objects, their textual descriptions or photos.

R-REC-20 HMD content is limited in time, then user is moving to another scene

R-REC-21 Client extends interactivity between devices

R-REC-22 Functionality of pausing and resuming of the content

R-REC-23 Functionality of jumping to another scene basing on users actions

R-REC-24 Highlighting areas where objects are located in the scene (HMD)

R-REC-25 Selecting objects from the scene in HMD to be displayed on tablet.

R-REC-26 Supporting ambisonics sound

R-REC-27 Synchronization across devices based on audio

5. MAPPING OF WP2 REQUIREMENTS

In this section, we clarify the relation between the sets of requirements defined in D2.1, D2.2 and D2.3 for the different pilot iterations.

5.1. Off-line documentary (pilot 1)

Mapping of WP2 requirements for PILOT 1: off-line documentary

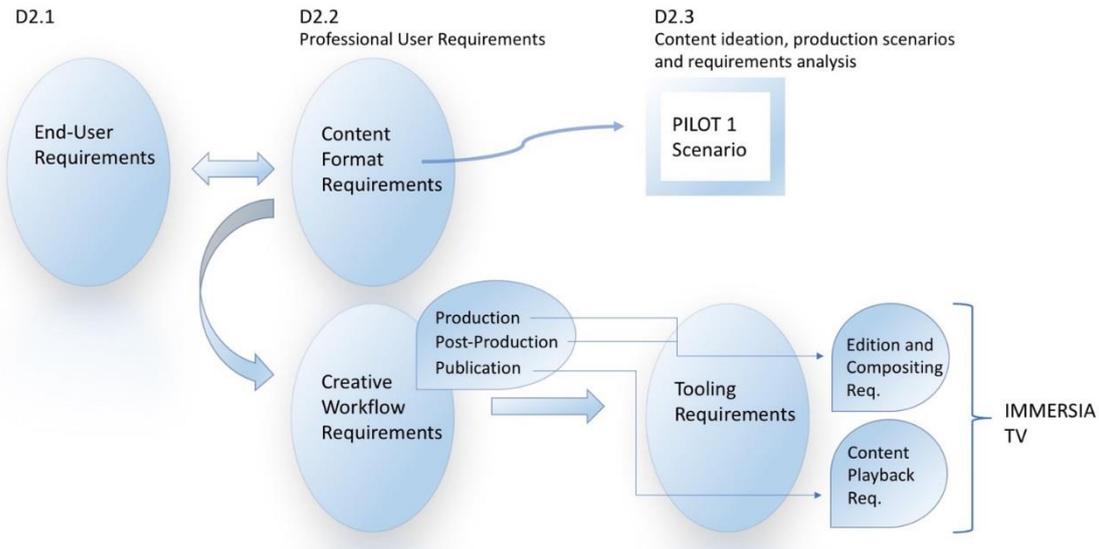


Figure 18 Mapping of WP2 requirements for pilot 1 (off-line documentary)

In pilot 1, end user (experience) requirements and professional user requirements have been gathered in an open approach, as explained in the introduction, including both generic requirements and specific requirements linked to ImmersiaTV concepts that have been carefully introduced in the co-creation workshops.

First, as shown in Figure 12, there is a clear interlink between the end user requirements (D2.1) and professional content format requirements (D2.2). These links are provided in detail in Section 7 of D2.2. They indicate largely similar generic requirements and notable associations around synchronisation and interactivity. These content format/experience-related requirements are input for ideation process in D2.3 leading to pilot content scenarios.

In parallel with content format requirements, professional user requirements regarding creative workflow have been gathered, and structured according to the workflow stages of content professionals: preproduction, production, post-production and publication. Crosslinks between content format and creative workflow requirements evidently exist and are also documented in D2.2 Section 7.

The creative workflow requirements are input for the definition of the formal list of detailed technical requirements in D2.3. Here we focus on the detailed software requirements of the needed ImmersiaTV components to be developed in WP3, in particular the software building blocks “edition/compositing” and “content playback”. “Story preparation” and “production preparation” have been documented as well, but have a lower priority in terms of implementation within the ImmersiaTV project.

For pilot 1, the main mapping between D2.2 creative workflow requirements and D2.3 software tooling requirements (see also D2.2 Section 7) can be summarized as follows:

- “production” and “post-production” map into “editing and compositing”
- “publication” map into “content playback”

The D2.3 requirements are more detailed, so typically D2.2 creative workflow requirement fans out into multiple software requirements.

Note: a similar mapping figure and reasoning as above applies for the Exploration Mode-based whodunnit in pilot 3.

5.2 Live event (pilot 2 and 3)

Mapping of WP2 requirements for PILOT 2: live event

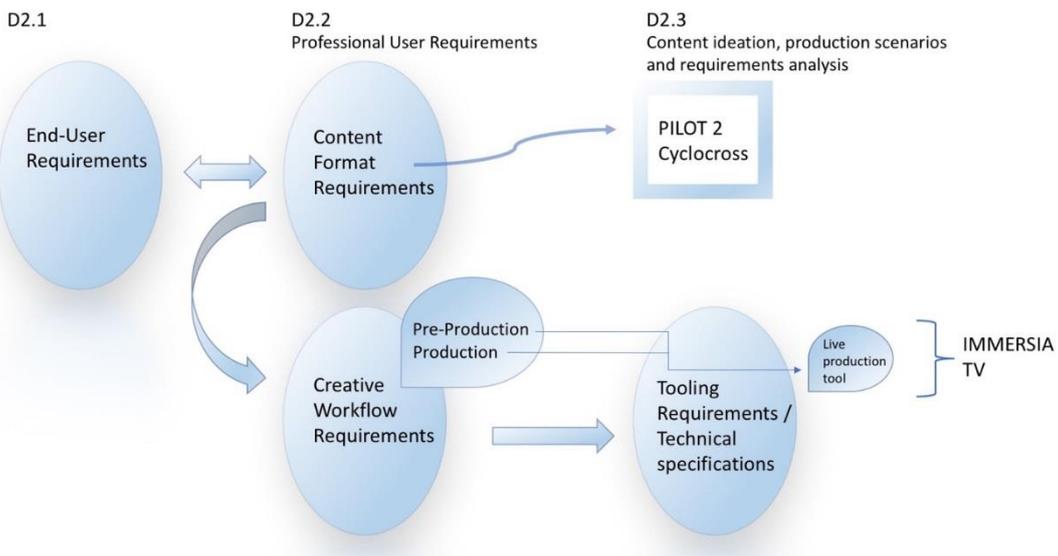


Figure 19: Mapping of WP2 requirements for pilot 2 (live event)

Similar to the off-line pilot, the set of requirements defined in D2.1, D2.2 and D2.3 are mapped as shown in Figure 19. In contrast, the tooling requirements focus on the live production tool which can be directly mapped to the production (and preproduction) related creative workflow requirements. The exact mappings are documented in Section 7 of D2.2.

Technical specifications on capture, stitching, encoding, distribution have been identified in separate technical workshops with ImmersiaTV partners and are mostly not directly linked to the requirements identified with professional users.