Deliverable

Project Acronym:	**VRTogether**
Grant Agreement number:	762111
Project Title:	*An end-to-end system for the production and delivery of photorealistic social immersive virtual reality experiences*

D2.3- User scenarios, requirements and architecture

Revision: 3.0 - Final version updating the one **officially approved** in M25 (V1.1) during the 2nd year review. **V3.0 has been issued to update** the changes in the requirements and architecture in the last year for exploring a diversity of case studies and improve accessibility to the platform as suggested by the review panel

Authors: Name Surname (institution)

Delivery date: M25 (V1.1 delivered in M25, actual V3.0 delivered in M39

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 762111		
Dissemination Level		
P	Public	
C	Confidential, only for members of the consortium and the Commission Services	

Abstract: This document provides an overview of the user scenarios addressed by the project, compiles the set of requirements that drive the design of the project platform, provides details regarding the pursued system features and provides an initial system architecture design.

Revision History

Revision	Date	Author	Organisation	Description
0.1	16-08-2019	G.Cernigliaro	I2CAT	Table of contents and initial indications for the writers.
0.2	16-08-2019	G.Cernigliaro	I2CAT	First draft of Section 2 and 4
0.3	25-10-2019	V.Lepec	VO	Section 5 Updates
0.4	25-10-2019	G.Calahorra	ENTROPY	Section 2 Production Updates
0.5	30-10-2019	B.Kevelham	ARTANIM	Section 2 Technology Updates
0.6	04-11-2019	P.Cesar	CWI	Section 6 Updates
1.0	05-11-2019	G.Cernigliaro	I2CAT	Unify versions and add last updates
1.01	08-11-2019	T. Koninck	TNO	Last review
1.1	**08-11-2019**	**G.Cernigliaro**	**I2CAT**	**Previously delivered Version (M25)**
2.0	03-11-2020	G.Cernigliaro	I2CAT	Update delivery preparation & Requirements last updates
2.1	10-11-2020	V.Lepec	VO	Architecture last updates
2.2	24-11-2020	G.Cernigliaro	I2CAT	Experiments new intro included
2.3	08-12-2020	P. Cesar	CWI	Update list of experiments
2.9	22-11-2020	G.Cernigliaro	I2CAT	Update Delivery Wrap Up
3.0	18-12-2020	G.Cernigliaro	I2CAT	Last revision, formatting for submission

EXECUTIVE SUMMARY

This document represents the reference for the software integration and the experiments performed in the VR-Together project. It aims to describe the requirements, the architecture and how the experimental work envisaged to implement the main paradigm outlined in VR-Together: the creation of a platform, and the corresponding media content, that allows two, or more users to feel as if they were together in a virtual environment. The togetherness feeling is provided delivering photorealistic content, both for media and end-user representations. The planning introduced in this deliverable is then executed in WP4, and thus the results of content creation, experiments, and pilots will be included in D4.3 and D4.4. In the previous documents D2.1 and D2.2, apart from describing the overall characteristics of the VR-Together platform, like requirements and experiments, we described the scenario, architecture and software platform considered for Pilot 1 and Pilot 2.

The latest, and previously submitted, version of this document (D2.3 – V1.1) updated the previous information, including:

- The production details related to Pilot 3 and the technological development needed to specifically deploy the pilot.
- The updates about the requirements processing
- Architecture and software implementations used for Pilot 3.
- Experiments and testing: the experiments that will pave the way to the pilots and the plan that the consortium has agreed to perform them.

This new version (D2.3 - V3.0) has been issued to finalize the reporting including the information added thanks to the feedback received during the 2nd year review of the VR-Together project, and provides a set of updates regarding:

- **The finalization of the requirements process and the definition of the Proof of Concept requirements (Section 4.3)**
- **The final architecture, including the new components defined during the last year of the project and introduced to provide a working VR-Together Proof of Concept (Section 5.3)**
- **The final list of experiments updated to validate the final delivered technology (Section 6.4.3)**

The document starts with an introduction (Section 1) followed by a short description of the high-level requirements initially defined for the project (Section 2). The requirements are fully described in Annex I, that correspond to Section 2 of the D2.2 version of this document.

Section 3 describes the pilots. Starting with a short description of Pilot 1 and 2 followed by the explanation of the details involved in the production and deployment of Pilot 3, together with the choices regarding the technology involved in it.

Section 4 describes the current requirements of VR-Together that represent the evolution from the high level ones mentioned in Section 8, after the processing detailed in D2.2.

Section 5 introduces the overall architecture for VR-Together, how the different components interact and the hardware topology, taking into consideration the software modules from WP3. The specific implementation details are provided in the WP3 deliverables.

Section 6 outlines all the information related to the distributed lab realized within the VR-Together project. It presents the Advisory board, resulting from the large-scale consultation with stakeholders and experts, supporting the project. Then it describes the existing connected user labs between the Netherlands and France (web-based pipeline) and between Greece and Spain (native pipeline) and the lab nodes at each partner’s premises. Finally, it lists the experiments associated, including the requirements that each experiment considers. The results of the experiments will be reported in D4.5.

Section 0 summarizes the contributions of this deliverable.

Last, Annex I (Section 8) provides the initial status of the VR-Together Requirements. Annex II (9) provides the D2.2 Requirements Matrix and Process status. Annex II (Section 9) shows the requirements considered in Pilot 1 and can be used as a reference after reading. Annex III (Section 10) provides an example of a questionnaire used for experimentation and a detailed description of the Lab nodes in each one of the partners' locations.Finally, Annex IV (11) is used to provide, to the reader, the information about the experiments previously planned for Pilot 1.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Javier	**Lajara**	ENTRO	javier.lajara@entropystudio.net
Guillermo	**Calahorra**	ENTRO	guillermo@entropystudio.net
Pablo	**Cesar**	CWI	p.s.cesar@cwi.nl
Argyris	**Chatzitofis**	CERTH	tofis@iti.gr
Simon	**Gunkel**	TNO	simon.gunkel@tno.nl
Romain	**Bouqueau**	MS	romain.bouqueau@gpac-licensing.com
Vincent	**Lepec**	VIA	v.lepec.ext@viaccess-orca.com
Gianluca	**Cernigliaro**	i2Cat	gianluca.cernigliaro@i2cat.net
Sergi	**Fernandez**	i2Cat	sergi.fernandez@i2cat.net
Mario	**Montagud**	I2CAT	mario.montagud@i2cat.net
Ana	**Revilla**	ENTRO	int@entropystudio.net

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Purpose of this document

The purpose of this public document is to provide the reader with a comprehensive view of the project requirements, the use cases contemplated for the scenarios designed for pilot 3 and the system architecture envisaged to meet the requirements. The document also gathers information regarding the User lab initiative of the project as well as other feedback methods such as experiments, advisory board and others.

1.2. Scope of this document

This document includes the current status of:

- Pilot 3 Production and Technology definition
- Requirements
- Pilot 3 Hardware and Software Architecture Status
- Experiments and Users Labs

1.3. Status of this document

The D2 deliverables follow the project all along its three iterations. Three different versions have been formally submitted to the EC and uploaded in the project website. This D2.3 document is an evolution of the D2.1 and D2.3 documents and provides the status of the third and last iteration.

1.4. Relation with other VR-Together activities

This document gathers the outputs of T2.1, T2.2 and T2.3 and serves as input for WP3 and T2.4. It also provides input to WP4 w.r.t experiment definition and evaluation methodology.

2. Project Goals

The main goal of the VR-Together project is to provide, to the end users, a fully immersive social VR experience able to involve the participants at different sensorial levels. The users are indeed able to be projected in a virtual environment, feel immersed in the three-dimensional space by visualizing their own self representation and, at the same time, see the other users and communicate with them in real time.

In order to provide the aforementioned experience, the consortium is developing a cutting-edge technology able deliver a set of features that covers the full VR-Together pipeline. Such features include, among the others, the reconstruction of the users' body in a volumetric video, the compression and the transmission of the stream, the rendering for the representation of all the media involved, and the orchestration of all the data to be delivered.

In this section of this D2.3 document, the general goals of the VR-Together project are described. The section will strongly refer to Section 2 of the D2.2 (included in this document as Annex I: Project Requirements), where the initial requirements have been listed and fully described. In this version, the requirements are described as high-level goals so that the reader can be involved in the project objectives since the beginning of the document.

2.1. VR-Together general requirements

The objectives of the project can be separated into two main parts, depending on their belonging the main scope or to a specific delivery. The general requirements address those features that are needed along the whole duration of the development, mostly referring the immersive nature of the communication and interaction conceived in VR-Together.

A VR immersive experience mostly focuses on providing a feeling of presence to the users, in terms of self-representation and in terms of how the other participants are perceived. Interaction and communication are fundamental characteristics that need to be defined by specific requirement. For this reason, the VR-Together generic requirements describe concepts as:

- **End-users' representation**: these requirements refer to the reconstructed end-user body, the quality of its representation and the way it is seamlessly blended with the virtual environment.
- **Communication and interaction**: these requirements specify that the communication between users should be natural, without big delays (low latency), and with an acceptable quality for the representation of verbal and non-verbal communication.
- **Technology**: the high-level characteristics of the technology used to provide the experience, together with the definition of the platforms compatible with it, are also defined in the initial general requirements.

2.2. Pilot depending goals

Together with the general project requirements, that affect the technology along the whole project development, the consortium has initially defined also a set of three intermediate deliveries, defined as Pilots. Each Pilot represents a milestone for the whole progress of the project and shows the technological advances reached. In the next sub-section, the specific goals of each Pilot are briefly introduced. As for the general requirements, the reader can have a detailed description of the initial requirements, defined pilot by pilot, in Annex I: Project Requirements.

2.2.1. Pilot 1

The goal of Pilot 1 has been to provide an initial feeling of immersion to the user. The number of supported users is indeed two, so that self and others representation can be already appreciated and, at the same time the easiness of the communication and interaction can be sensed.

The Pilot 1 initial requirements focus, then, mostly on the representations of all the participants, being these end users and other characters of the experience provided.

2.2.2. Pilot 2

Pilot 2 has been defined as the second step of the development. At this stage most of the features of the previous pilot will be improved and new features will be introduced.

Among the new features, the most important are represented by the possibility of a multi-user experience (more than 2 end users can participate at the same time) and by the presence of a live component.

2.2.3. Pilot 3

Finally, Pilot 3 will provide the full experience at its best in terms of technical development and providing, to the users, the most immersive experience among the three available.

By the point of view of the technological advances, the users will be able to experience the best possible volumetric reconstruction of themselves and of other users, with the lowest possible latency and with the possibility of involving a number of participants higher than 4.

By the point of view of the immersion, the users, in Pilot 3, will be able to finally interact not only with other users, but also with the environment, being capable to interact with objects at different levels and actively changing the path of the story.

3. Pilot Scenarios

Figure 1 VR-Together Pilots Calendar

3.1. Pilots 1 and 2

This section summarizes pilot 1 and 2, focusing specially on the plot and the technology behind their post-production. A more detailed description and analysis of both pilots can be found in the deliverables D2.1 and D2.2.

A brief description of the pilot 3 is, as well, presented in this deliverable. This pilot is the final chapter of the narrative and tries to summarize technologically the objectives of the project.

3.1.1. 3.1.1 Plot

The story around the three pilots covers the murder of Ms. Armova, British socialite, who is found dead one night in her apartment. Pilots 1 and 2 focus the story on the interrogation of the main suspects and the police investigation after the murder, respectively. It was decided at the beginning of the project that each pilot would cover a different treatment of the technology developed, thus the story and plot would also be adapted depending on these technological requirements.

Pilot 1 focuses on the interrogation of the two main suspects of the crime. Ryan Zeller, suspicious "party boy", saw the victim before she died. On the other side, Christine Gerard was the victim's personal assistant, and she was she last person who saw her alive. Since the main plot of the pilot focuses on this situation, two different situations were filmed. Two users are located within a police interrogation building, seeing each other in different points of the room. Each user will only be able to watch a specific interrogation (Ryan Zeller or Christine Gerard).

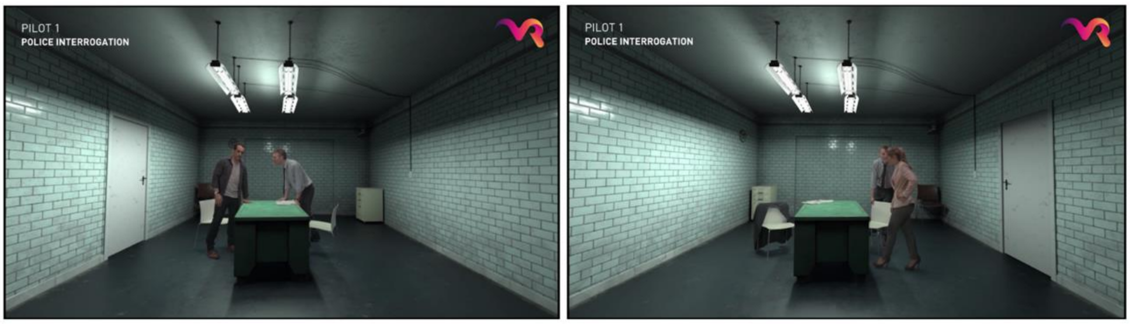


Figure 2 Left: Ryan Zeller interrogation. Right: Christine Gerard interrogation.

Once the users have watched the interrogation, they both will need to share what they have seen and what they have listened. They will need to pay attention to different details to try to decide who they think is the responsible for the death of Ms. Armova. Alongside the official interrogations, a TV in front of the users will show them different video footage that might help them solve the crime. Steve Galache and Almudena Calvo play Ryan Zeller and Christine Gerard, respectively. Jonathan David Mellor plays the role of the police officer, who will also appear on the second pilot (since he leads the team trying to solve this case). A dissemination video was made, explaining the project, showing the shooting process and how the post-production was done. The video can be watched by clicking here. A post in the official VR-Together website was written explaining the shooting process. This post can be read by clicking here.

The story of the **second pilot** takes place days after Ms. Armova's death. The pilot mixes two different scenarios. The users become part of the audience of a famous TV Show, where the host explains the situation of the murder of Ms. Armova and how the police is trying to solve the crime. There are two experts who collaborate within this TV show: a technology expert and a TV reporter who is outside the crime scene, trying to get information about the case.

Figure 3 TV scenario with users and live connection (right)

The TV host will give the floor to the reporter, answering his questions about the case. The police officer from the first pilot will make an appearance, and the reporter will try get him answering all the questions related to the case. Interactivity is one of the main keys for the second pilot, where the audience will be able to ask the host different questions in a live version of the pilot. In the television program there are 4 users who can interact in real time, asking the presenter. These users are a mix of fake users and users scanned in real time. Deliverable 4.3 shows a better detailed version of the pilot 2. A post in the official VR-Together website was written explaining the shooting process. This post can be found by clicking here.

3.1.2. Technology

Pilot 1

The goal of the first pilot was proposed within an offline environment, trying to demonstrate that a mix of innovative media developed exclusively for the VR-Together project can produce a more intimate and binding activity than more traditional content production pipelines, based on omnidirectional content (orchestrating point clouds, 3D Mesh based models and multiple video sources).

We compared different capture and production techniques (video, point cloud capture, high-end motion capture) as well as combinations of them to determine quantitative balances among the different formats available (video, point clouds, time-varying meshes, dynamic meshes, motion data). A police story was ideal to demonstrate the innovative character of the project. The goal was to demonstrate that the orchestrated delivery of the VR-Together media format, combining several video sources, point cloud and 3D mesh representations improves closeness between the 2 end users.

The VR content displayed is stored in the end-user device, categorizing the first pilot as “offline content”. Some achievements that made the experience successful were the following: the detail in the facial expression in both the end-user’s and the character representation, the illumination was consistent in the whole experience and the scene contains rendered characters, among others achieved goals.

Pilot 2

For the second pilot, different aspects already developed in pilot 1 have been used and improved. End users must be able to see themselves and other users’ representation, and they must be able to hear the sounds made by another user in the virtual space of VR-Together. Because of this, the pilot focuses on the latency, finding a system which allows an end user to have a network latency allowing for seamless and natural communication and interaction with other users.

We are also working in capturing the visual representation of the user, developing a VR-Together platform's capturing setup. The level of detail of end-user representation allows the recognition of facial expressions. The representations of the rendered characters inside the virtual space has parallax and depth to allow for a 3D representation, so the end user can see it. The VR-Together native platform processes end-user’s live coloured 3D point cloud to reconstruct a 3D time-varying mesh in real-time.

Another goal is to achieve a hyper-realistic and optimized scenario so that each end user can create a reconstruction of the virtual space of VR-Together.

Figure 4 TV scenario with users and live connection (right)

It is very important the integration and synchronization of the content generated in time relative to the content created previously. End users in distributed locations can share a virtual space and can see the same VR content. This photorealistic VR content is prepared in different formats. End users, scene of action and characters are projected in the virtual space of VR-Together using different media formats. The resulting VR image should be a blend of different formats. Deliverable 2.2 details different aspects of the pilots 1 and 2 regarding, among others, the general requirements, scenario details and an extended calendar that illustrate each task.

3.2. Pilot 3

Pilot 3 embodies the work realized during the three years of the project. It includes all the features developed for Pilot 1 and Pilot 2 and adds components as interactivity and more users, scanned in real time that are part of the story, choosing different endings.

3.2.1. Plot

The story will take place inside Armova's apartment, where Sarge and the users will be looking for evidences that can confirm some hypothesis regarding the murder. 5 users will have the possibility to have an active role in this piece, using different objects to interact within the environment. Figure 6 summarize the type of action that users can execute. Rachel Long will also make an appearance in this final episode: a technology expert that will show how to represent a hologram of Elena Armova. An interrogatory between Sarge and Elena will take place, trying to solve the mystery around the crime, and answering the question brought in the previous two episodes: who was the murderer? Was it Zeller or Gerard? At the end of the experience, users will have the possibility of choosing which way they want to take: to follow Ms. Armova and Rachel or to follow Sarge.

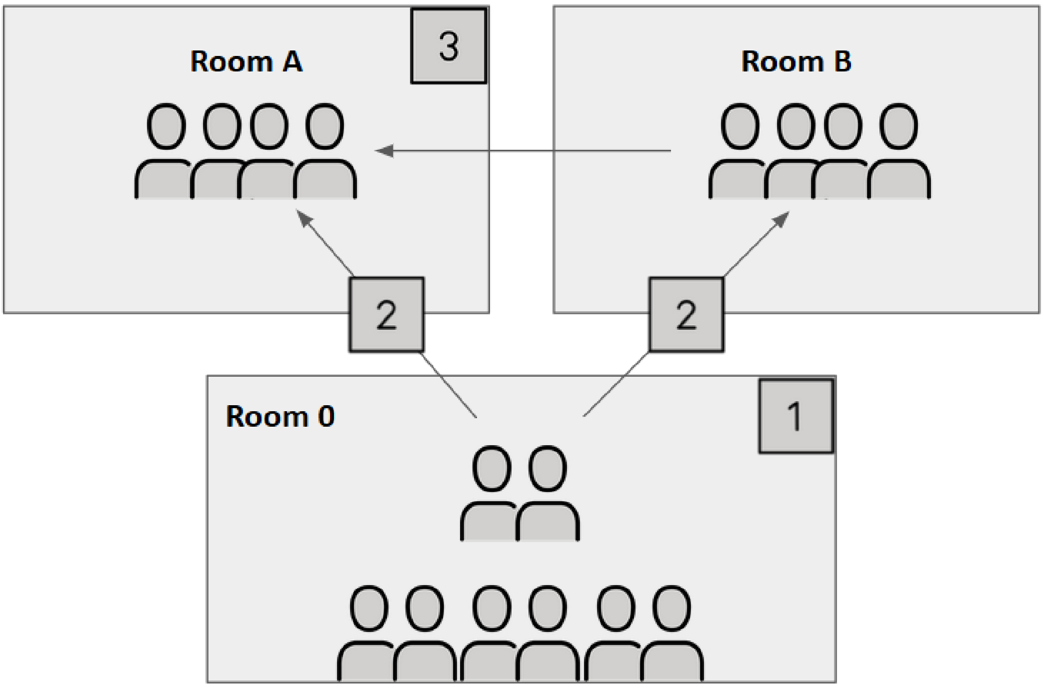


Figure 5 TV scenario with users and live connection (right)

The structure of the pilot is illustrated through the previous figure. It all starts in the room 0, where the main plot is presented. When users have the chance to choose which way they want to go, they will have to decide to enter room A or room B. Different situations will trigger the development of the story, and users from room B will have to change to room A in order to end the story up.

In the first room, users will have the possibility of choosing the way they want to take, whether following Ms. Armova and Rachel or Sarge. Sarge and Ms. Armova will have, eventually, a conversation about the previous night. Users would be able to develop simple actions, such as: switching the lights, picking up objects, pressing buttons, etc.

In the kitchen, three users will follow Sarge and Evans. Evans would examine different clues and will maintain a conversation with Sarge the whole time. In this space, users would be able to insert a card in some places, to press buttons or to talk to Sarge. In the other room, Armova's bedroom, 2 users will follow Armova, Rachel and the other tech, who will examine different clues as well. As in the other room, users will be able to pick a card or to talk to different characters.

3.2.2. Production Plan

Different options have been proposed to keep the story going. Planned for October 2019, the third pilot will close the story of Ms. Armova's tragic death. Using new hardware technology, the pilot will be necessary to understand what happened that night. The representations of the rendered characters, inside the virtual space of VR-Together Pilot 3, must be able to retarget their gaze according to the end-user's viewpoint. The representations of the rendered pre-rigged characters, inside the virtual space of VR-Together Pilot 3, must be able to retarget pointing gestures.

The pre-production phase for this pilot was expected to last several weeks, from July to November. In this stage, tasks such as the script development or the concept development are performed. Specifically, these two actions were discussed with all the members of the consortium, asking for any specific requirements in terms of artistic and cinematographic direction of the plot. The other tasks related to the first phase of the whole production are related to the selection of the crew and the appropriate clothes for the shooting. The proper shooting is expected to happen between the third and fourth week of November, when every aspect related to the pre-production is arranged and solved. At the same time, the study of the environment and the 3D scanning are expected to happen.

Once the shooting has been done, the post-production phase will take place. The motion capture cleaning tasks will happen at the same time as the scan rigging, the motion retargeting and the chroma cleaning, all of them crucial at any post-production process. Lastly, the 3D modelling and texturing (for both scenario and objects) will be happening during the whole process, leaving two weeks extra at the end for detail correction.

The following table summarizes the outlines for the pilot 3.

SCENARIO	The pilot 3 takes place in Elena Armova´s house: 3 rooms + 2 groups
ROOM 1 1 GROUP (5 users)	In the first sequence we are in the living room of the House. There are 5 users placed in specific positions in the room. Characters: Sarge, Rachel (Sarge's assistant), Elena Armova (AI). Sarge says that they must find clues in the house and talk directly with the users asking to doing actions.

	USER ACTIONS: 1.- To turn on/off the lights. A user is located close to a switch in the wall. 2.- To take one flashlight and charge it with a battery. 3.- To take a remote control and push a button. 4.- To open a drawer and push a button inside 5. To answer questions of Sarge: YES / NO Elena goes out to other room. Sarge tells two users to go with him and the rest with Elena to find clues
ROOM 2 (kitchen and Armova's bedroom) GROUP 1: Sarge's (2 users) GROUP 2: Armova's (3 users)	Group 1 (2 users) enters Room 2 with Sarge Group 2 (3 users) enters with Elena and Rachel to Room 2 They perform actions similarly as they do in the first room, but within a different context. USER ACTIONS 1. To answer questions of Sarge/ Elena (depending on the group): YES / NO 2. To answer questions of Sarge/Armova (depending on the group). They can answer something long enough so Sarge can answer: "It is impossible to understand you! Such a bad English! Don't waste my time!" 3. To press something 4. To grab a paper 5. To open a drawer 6. To insert a DVD in a player 7. To find something under furniture
ROOM 3 (living room) GROUP 1: Sarge's (2 users) GROUP 2: Armova's (3 users)	The two groups get mixed in a third room and they experiment different endings depending on what group they were coming from.

Figure 6 Summary of outlines for pilot 3

There will be a total of 5 characters in the third pilot. Each character will be played by an actor/actress that is already selected or will be chosen from a casting process. The individual characteristics of each one are the following:

- **Sarge**. Played by Jonathan D. Mellor.
  - Fully body scan and rigging
  - Scanning and facial rigging
    - Facial expressions: eyebrow and mouth movement
  - Motion animations – MOCAP
  - Develop actions of all kinds
    - Taking objects with his hands, walking, turning himself and looking directly to people, talking to the other protagonists and interacting verbally with users
- **Armova**. TBA, decided through casting.
  - Fully body scan and rigging
  - Scanning and facial rigging
    - Facial expressions: eyebrow and mouth movement
  - Motion animations – MOCAP
  - Develop actions of all kinds
    - Taking objects with his hands, walking, turning himself and looking directly to people, talking to the other protagonists and interacting verbally with users
- **Rachel**. Player by Ana Revilla
  - Fully body scan and rigging
  - Scanning and facial rigging
    - Facial expressions: Mouth movement
  - Motion animations – MOCAP
  - Develop actions of all kinds
    - Taking objects with his hands, walking, talking to the other protagonists
- **Evans**. TBA, decided through casting.
  - Fully body scan and rigging
  - Motion animations – MOCAP
  - Develop actions:
    - Taking things with his hands
    - Walking
    - Talking to the other protagonists
- **Tech. 2**. Played by Guillermo Calahorra
  - Fully body scan and rigging
  - Motion animations – MOCAP
  - Develop actions:
    - Taking things with his hands
    - Walking
    - Talk to the other protagonists

In the third pilot, users interact with the environment and with the main characters. A system that allows an end user to have a network low latency allowing a natural communication and interactions is then the main focus. Work has been done to capture the visual representation of the user and its facial expressions. The volumetric capture process allows representations of the rendered characters.

To summarize, the developments of pilot 3 are:

- Facial Expressions. Photorealistic detail to see facial expressions (users and Characters).
- Passive watch. Users can see the content in a passive way.
- Active watch. Character can participate in the action.
- Movement. 6DoF movement.
- Derived actions. user actions can change the plot. We are preparing 2 different endings.
- Pattern Recognition. Multi modal pattern recognition tools can be used and integrated into the plot.
- Pointing
- Talk
- Physical actions. Users can trigger story actions by performing simple physical actions.
- Interactive storytelling
- Interactive characters

Some of the steps that need an extra attention are the following, including an estimated time for their resolution:

- Casting: Elena Armova and Evans (approximately 3 weeks)
- Scanning and rigging of characters (1 week)
- 3D Characters (approximately 3 months)
- Photogrammetry of the three different rooms (approximately 3 months)
- Shooting: 2 weeks of pre-production and 3 days shooting
- MOCAP: 4 months
- Post-production: regarding sound, integration and compositing. 4 months approximately.

To end with the needed requirements for pilot 3, the following table sums up all the details that are mandatory to offer the greatest experience in terms of quality and content.

Title	Description
VR Experience	An end user should have an experience that visually and acoustically allows to perceive and understand the other participants' body language expressions
VR Content Visual Quality	The virtual character representation must be detailed enough to allow for the recognition of facial expressions
VR experience	The level of detail of end-user's representation in the virtual space of VR-Together must allow the recognition of facial expressions
VR content	The representations of the rendered characters, inside the virtual space of VR-Together pilot 3, must be able to retarget their gaze according to the end-user's viewpoint

VR content	The representations of the rendered pre-rigged characters, inside the virtual space of VR-Together pilot 3, must be able to retarget pointing gestures
Active watch	The end user inside the virtual space of VR-Together pilot must be able to become a character within the storyline that is being projected
Movement	The end user inside the virtual space of VR-Together pilot 3 must be able to move. 6DoF
Derived actions	The end-user's actions inside the virtual space of VR-Together pilot 3 must lead to changes in the storyline that is being projected
Pattern recognition interaction	The VR-Together pilot 3 platform must support multi modal pattern recognition mechanics for changing the storyline according to user's choices
Pointing interaction	The VR-Together pilot 3 must be able to recognize pointing gestures of end users and change the storyline accordingly
Speech interaction	The VR-Together pilot 3 platform must be able to recognize the speech of end users and change the storyline accordingly
Interactive character	The system for pilot 3 should integrate and use interactive character animation
Configuration	The VR-Together platform orchestration component (for pilot 2 and 3) must manage sessions where at least 2 end users participate in a virtual space
Configuration	The VR-Together platform orchestration component (for pilot 2 and 3) should support at least two parallel sessions
VR scenario	The VR-Together pilot 3 platform must allow 10 end users to simultaneously be in the same virtual space

Table 1 Requirements for VR-Together pilots

3.2.3. Technological considerations

3.2.3.1. Character representations

Pilot 1 and 2 have seen a variety of content formats being used for characters (live or recorded) being a part of the experiences. Experiment Artanim-2.1 executed during the second year of the project showed that in the context and setting of the pilot 1 showcase, users mostly preferred virtual characters to be represented by billboard videos or full 3D characters driven by motion capture recordings. Participants in the experience however did note that where realism and expressiveness are concerned, the 3D characters lagged somewhat behind.

When considering the options for pilot 3, there are a variety of constraints and requirements to consider. As specified in deliverable D2.2, a pilot 3 specific requirement is:

P3	11	Interactive Character	The system will integrate interactive character animation techniques

The highly interactive scenario written for this pilot further solidifies this element. The characters will directly interact with the users of the experience, they will go through different scenario paths based on those interactions, and further will on occasion interact with the exact same virtual-physical object with which the users interact. Unlike the in-one-shot recording and playback of content as was a part of pilot 1, this requires the real-time and seamless blending of various animation, and their on-the-fly adaptation to the current experience state.

While with a very careful consideration of stereo billboard recording and processing some form of an interactive experience could have been created, the conclusion of experiment Artanim-2.1 as reported in D4.4 provides another downside: *"It is important to emphasize that the VR technique that relies on a mixture of video and 3D graphics imposes constraints on the design of the experience"*. The wide variety of a user's possible viewpoints in pilot 3 may break the illusion of a coherently integrated 3D experience when relying on billboards for characters.

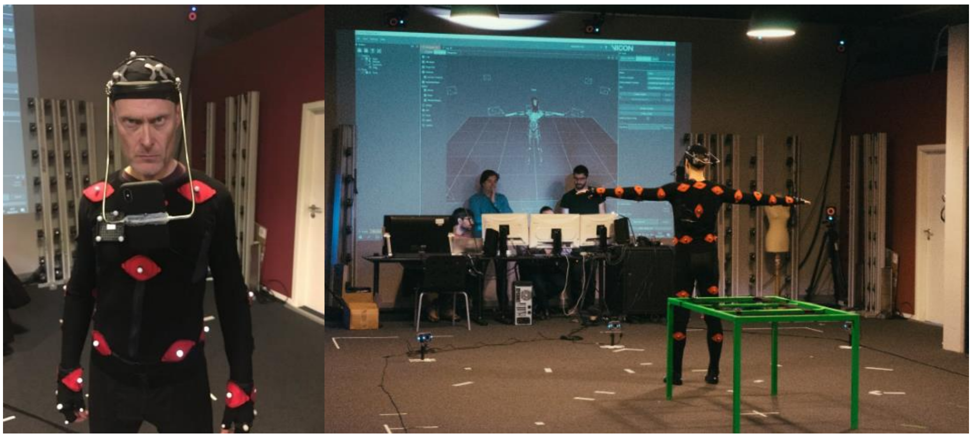


Figure 7 Pilot 3 will use 3D characters animated with motion capture data, to be redirected where suitable.

With that in mind the choice for a fully 3D character representation driven by pre-recorded motion capture data becomes evident and that is our choice going forward. The choice for a 3D character representation also allows us to address the following functional platform requirements as specified in D2.2.

FR	32	Gaze	The representations of the rendered characters inside the virtual space of VR-Together MUST be able to retarget their gaze according to the end-user's viewpoint
FR	33	Pointing gestures	The representations of the rendered characters inside the virtual space of VR-Together MUST be able to retarget pointing gestures

To address the quality concerns raised before, the creation of the 3D rigged characters with appropriate facial blend-shapes will be outsourced to a professional service. This should significantly raise the quality of the end-result in comparison with the 3D avatars as available in pilot 1. And in addition, the motion capture solutions used have seen several updates since pilot 1, which should allow for an improved end-result as well, particularly where the character's hands and fingers are concerned.

3.2.3.2. Interaction controls

Pilot 3 specifies several interactions with virtual objects, in particular triggers:

- *"Hit that light switch", "Turn it off"*
- *"Hold down button three for a few seconds"*

and grasp:

- *"Put the batteries in that torch there"*
- User X opens the drawer, removes the remote

This implies at least the ability to track the hand positions and gestures for a user. This information is however unavailable in both the Point Cloud and TVM user representation. The solution we have settled upon is to use standard VR controllers such as the Oculus Touch. These controllers will naturally provide us with tracked hand positions, and through standard button interaction can give us gesture information such as "pointing" for triggers and "grasping" for object manipulation.

Figure 8 Controllers will be used to determine hand position and orientation as well as interaction gestures

In addition, VR headset positional information will be used for more passive interaction such as the direction of gaze or pointing gestures from virtual characters to users as described above.

3.2.3.3. Client synchronization

Pilots 1 and 2 had a limited number of synchronization points, i.e. moments where all clients/servers needed to be synchronized on a particular event. Such events could be experience starting point, the play-out of certain content streams or other such elements in the experience. Pilot 3 sees this need for synchronization increase significantly.

First there is the "player data" in direct relation to the interaction controls. This includes for example the position and orientation of a player's head and hands, as well as any gestures made. Then there are the objects a user interacts with. If one user picks up an object, this should be clearly and naturally visible to all other players. And of course, there is a multitude of events which will happen. A non-exhaustive list of examples is:

- User A grasped object X with Left Hand.
- User B pressed light switch Y.
- Start animation timeline "Elena Armova AI Launch".
- User C said "No".

For some of these events, "someone" needs to be in control, i.e. be the authority who decides what the next step in the experience will be. This is a fairly common problem in multiplayer games, which often see one of two solutions:

- **An authoritative server**. I.e. a dedicated server running the same experience as all clients run, but with the final say in all events which happen.
- **An authoritative client**. I.e. a regular client which get trusted with being the authority for the experience, getting the final say in all events which happen. This is commonly referred to as a "master client" as well.

An authoritative server implies that for every experience we wish to run, a server instance needs to be deployed. When considering the Unity eco-system this generally implies a solution – provided by Unity or a third party – which launches a server in the cloud. While such solutions are available or about to become available, this becomes problematic in cases where local demonstrations are required, such as at trade shows or conferences for example. One possibility that was considered was to extend the Orchestrator which is a part of our platform with the ability to act as an authoritative server. This was dismissed for two reasons: first, the development required to extend the Orchestrator to fill this role and possess enough knowledge of the experience to be useful would be prohibitively expensive in terms of effort. Second, content authoring in tight synchronization with a Unity-based client would also be potential bottleneck.

The authoritative client approach is attractive because it simplifies both development and

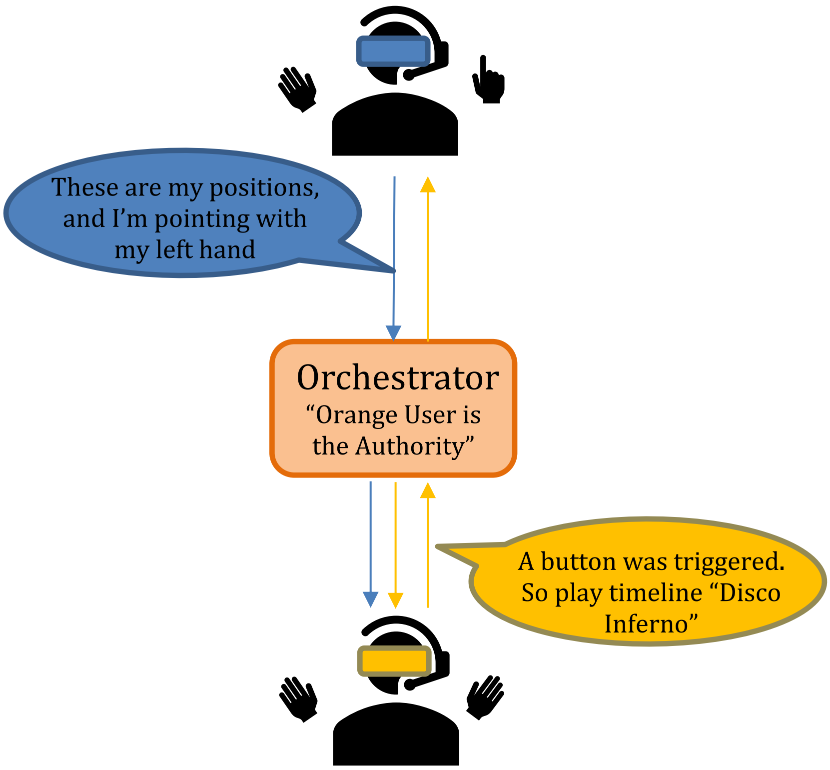


Figure 9 Authoritative client solution using the Orchestrator as the communication platform

deployment. No additional infrastructure is required, and development is all part of the same project. When evaluating possible solutions, particularly Unity's built-in networking, there were several downsides; the offered solution is marked deprecated, and a reliable connection requires the use of a relay server adding up to a second of latency. With that in mind and given the limited set of different synchronization actions required, we have opted to use the Orchestrator as a client communication platform as illustrated in Figure 9. In contrast to the previous case, now the Orchestrator – already a part of our platform – can take on the additional role of exchanging data and events between clients and selecting a specific client to act as the authoritative entity.

4. Requirements Matrix and Process

VR-Together is a software platform for an end-to-end communication pipeline between end users in virtual reality. As mentioned in Section 2, the software is described by several requirements that define its functionalities and characteristics. Section 2, together with Annex I: Project Requirements, show the initial ideas behind the functionalities that the platform has to deliver and the first set of requirements describing such functionalities. During the development of the project, both in terms of functionalities definition and technological development, the requirements have evolved based on a set of actions taken by the consortium, such as experiments, surveys and feedback from experts after experiencing demos of the platform functionalities.

The requirements definition has been evolving along the project and each update has been i) the outcome of a peer reviewing by all the partners and ii) fully tracked in the document *VR-Together Requirements Matrix*[1].

All the above-mentioned information, has been extensively reported in the previous version of this document (D2.2), that can also be found as an annex in this deliverable (Annex II: D2.2 Requirements Matrix and Process). The annex also shows the requirements matrix updated at the moment of the delivery of the D.2.2 document.

In this section we describe:

- The adopted requirements definition process
- The status of the requirements at the moment of the release of this document

4.1. Requirements definition process

The Requirements Matrix generation has been defined since the beginning following a clear distinction between *functional* and *non-functional* requirements, defined as:

- Functional requirements (FR): Define what the system must accomplish or must be able to do.
- Non-functional requirements[2] (NFR): The required overall attributes of the system, including portability, reliability, efficiency, human engineering, testability, understanding, and modifiability.

The above differentiation has been used to state clearly which requirements define the main functionalities of the platform and which one, instead, define the technology and the performance needed to provide the aforementioned functionalities. In order to state the priority of each requirement, the MoSCoW prioritisation has been adopted. The MoSCoW method implies that, in order to give an idea of the importance of the functionality described, each requirement specifies, in the text, one of the following words: “must”, “should”, “could” or “won’t”.

MUST	Requirements labelled as ***MUST*** have to be included in the solution to be a success. Think of MUST as a requirement that without it the result is considered a failure.

[1] VR-Together Requirements Matrix https://docs.google.com/spreadsheets/d/12IF74tYmMmCin5_hAgm0PL4ajia6Ii-WQ3fS-NWS5q0/edit?ts=5c87c8f2#gid=196101635

[2] A. Davis (1993). Software Requirements: Objects, Functions and States. Prentice Hall.

SHOULD	***SHOULD*** requirements are as important as *MUST*, although *SHOULD* requirements are often not as critical or have workarounds, allowing another way of satisfying the requirement. They are important and of high value to the user but even without them the system could still be considered a success.
COULD	Requirements labelled as ***COULD*** are less critical and often seen as 'nice to have'.
WON'T	***WON'T*** requirements are either least-critical or not appropriate at that time.

The process to define the requirements has been revisited during the second year of the project, putting in place a peer reviewing flow that has involved all the partners. The reviewing has focused on the following tasks:

- **Quality of the requirements**: the partners have proceeded in checking the validity of the requirements, the structure of the text, the redundancy to identify double requirements to deprecate
- **Clustering of the requirements**: the requirements have been grouped in subsets to facilitate the identification of the functionality and the reviewing
- **Feasibility Analysis**: the functionality and the performances have been analysed
- **Experiments Linking:** Requirements and experiments have been connected identifying from which experiments the requirements depend on.

Several examples of the process have been already depicted in the document D2.2 (in this document as an annex in Section 9). In addition, all the iterations with the ongoing results of the process mentioned above, can be found in the document *VR-Together Requirements Matrix*[3].

4.2. Requirements status

At the moment of the delivery of this document, the process has produced the final version of the requirements affecting Pilot 2 and the definition of the linking between requirements and experiments.

The Production and Technology description, provided in Section 3 of this document, have been defined according the General VR-Together requirements affecting the technology along the whole project and, at the same time, taking into account the requirements strictly affecting Pilot 3 (summarized in Table 2).

ID	Component	Title	Description
FR.17.0	VRT	VR Experience	An end user SHOULD have an experience that visually and acoustically allows to perceive and understand the other participants' body language expressions.

[3] VR-Together Requirements Matrix https://docs.google.com/spreadsheets/d/12IF74tYmMmCin5_hAgm0PL4ajia6li-WQ3fS-NWS5q0/edit?ts=5c87c8f2#gid=196101635

FR.27.1	VRT	VR Content Visual Quality	The virtual character representation MUST be detailed enough to allow for the recognition of facial expressions.
FR.28.0	VRT	VR experience	The level of detail of end-user representation in the virtual space of VR-Together MUST allow the recognition of facial expressions.
FR.32.1	VRT	VR Content	The representations of the rendered characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget their gaze according to the end-user's viewpoint
FR.33.1	VRT	VR Content	The representations of the rendered pre-rigged characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget pointing gestures
FR.41.1	OR	Active watch	The end user inside the virtual space of VR-Together Pilot 3 MUST be able to become a character within the storyline that is being projected
FR.42.1	VRT	Movement	The end user inside the virtual space of VR-Together Pilot 3 MUST be able to move (translation). 6DoF
FR.43.1	OR	Derived actions	The end-user's actions inside the virtual space of VR-Together Pilot 3 MUST lead to changes in the storyline that is being projected
FR.44.1	VRT	Pattern recognition interaction	The VR-Together Pilot 3 platform MUST support multi modal pattern recognition mechanics for changing the storyline according to user's choices
FR.45.1	VRT	Pointing interaction	The VR-Together Pilot 3 platform MUST be able to recognize pointing gestures of end users and change the storyline accordingly
FR.46.1	VRT	Speech interaction	The VR-Together Pilot 3 platform MUST be able to recognize the speech of end users and change the storyline accordingly
FR.48.1	VRT	Interactive character	The system for Pilot 3 SHOULD integrate and use interactive character animation
FR.119.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, MUST manage sessions where at least 2 end users participate in a virtual space.
FR.121.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, SHOULD support at least two parallel sessions.
FR.130.0	OR	VR Scenario	The VR-Together Pilot 3 platform MUST allow for 10 end users to simultaneously be in the same virtual space.

Table 2: Set of requirements strictly affecting Pilot 3

The third year of the development of the project will start finalizing those requirements that have not been going through the processing already performed and tracked in the document *VR-Together Requirements Matrix*[4].

The following two tables (Table 3 and Table 4) show, respectively, the status of the Requirements Matrix at the moment of the release of this document and the table that links requirements and experiments.

[4] VR-Together Requirements Matrix https://docs.google.com/spreadsheets/d/12IF74tYmMmCin5_hAgm0PL4ajia6Ii-WQ3fS-NWS5q0/edit?ts=5c87c8f2#gid=196101635

ID	Type	No	Version	Component	Title	Description
FR.1.0	FR	1	0	PL	Self-representation	An end user MUST be able to see his own representation in the virtual space of VR-Together
FR.2.0	FR	2	0	PL	Users audio representation	An end user MUST be able to hear the sounds made by another user in the virtual space of VR Together
FR.3.0	FR	3	0	PL	Users representation	An end user MUST be able to see the visual representation of another user in the virtual space of VR Together
FR.4.0	FR	4	0	CA	Audio Capturing setup	A location where the VR-Together platform's capturing setup is deployed MUST capture the audio generated by the user
FR.5.0	FR	5	0	CA	Visual Capturing setup	A location where VR-Together platform's capturing setup is deployed MUST capture the visual representation of the user
FR.8.0	FR	8	0	DE	Latency	An end user MUST have a network latency allowing for seamless and natural communication and interaction with other users in the virtual space of VR-Together
FR.9.1	FR	9	1	OR	VR Scenario	An end-user client MUST be able to create a reconstruction of the virtual space of VR-Together.
FR.11.0	FR	11	0	VRT	VR content formats	End users SHOULD be able to see different examples of VR content formats
FR.12.1	FR	12	1	VRT	VR content visual quality	End users MUST be able to see photorealistic VR contents
FR.13.0	FR	13	0	VRT	Synchronization	End users in distributed locations sharing a virtual space MUST be able to see the same VR content at the same time
FR.15.1	FR	15	1	PL	VR content visual quality	End users SHOULD see other users seamlessly blended in the virtual space of VR-Together. The Seamlessness evaluation will be performed by TBD.
FR.16.0	FR	16	0	VRT	VR Experience	End users SHOULD feel comfort in being immersed in the virtual space of VR-Together, at least for the duration of the pilot experience
FR.17.0	FR	17	0	VRT	VR Experience	An end user SHOULD have an experience that visually and acoustically allows to perceive and understand the other participants' body language expressions.
FR.18.0	FR	18	0	PL	VR Content	The VR audio content MUST be directional giving the perception of point sources within the virtual space of VR-Together.
FR.20.0	FR	20	0	VRT	End-user devices	End users MUST be able to access the VR-Together platform by using commercially available HMDs and capture systems
FR.22.0	FR	22	0	PL	VR content visual quality	End users, scene of action and characters SHOULD be able to be projected in the virtual space of VR-Together using different media formats. The resulting VR image should be a blend of different formats.
FR.23.1	FR	23	1	DE	Networks	The data transmission within VR-Together MUST be using commercial communication (e.g. MPEG-DASH) and media delivery networks (e.g. CDNs)
FR.24.0	FR	24	0	EE	Networks	Media streams SHOULD provide adaptive quality to network, device and interface capabilities

FR.25.0	FR	25	0	VRT	Web interface	End users MUST be able to access the VR-Together platform using a web application.
FR.26.0	FR	26	0	VRT	Native interface	End users MUST be able to access VR-Together platform using a native application
FR.27.1	FR	27	1	VRT	VR Content Visual Quality	The virtual character representation MUST be detailed enough to allow for the recognition of facial expressions.
FR.28.0	FR	28	0	VRT	VR experience	The level of detail of end-user representation in the virtual space of VR-Together MUST allow the recognition of facial expressions.
FR.30.1	FR	30	1	PL	VR Content	The VR-Together platform MUST be able to display the VR content which, depending on the configuration, can be either i) local or ii) stored in a network server
FR.31.0	FR	31	0	VRT	VR Content	Illumination MUST be consistent in the whole experience
FR.32.1	FR	32	1	VRT	VR Content	The representations of the rendered characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget their gaze according to the end-user's viewpoint
FR.33.1	FR	33	1	VRT	VR Content	The representations of the rendered pre-rigged characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget pointing gestures
FR.35.0	FR	35	0	PL	VR content visual quality	The representations of the rendered characters inside the virtual space of VR-Together MUST have parallax and depth to allow for a 3D representation.
FR.36.0	FR	36	0	PL	VR content visual quality	The end user inside the virtual space of VR-Together MUST be able to perceive the 3D appearance of the characters (parallax, depth)
FR.37.0	FR	37	0	VRT	VR Experience	The end user inside the virtual space of VR-Together MUST be able to rotate their head and have certain level of translation capacity while seated (3DoF+)
FR.41.1	FR	41	1	OR	Active watch	The end user inside the virtual space of VR-Together Pilot 3 MUST be able to become a character within the storyline that is being projected
FR.42.1	FR	42	1	VRT	Movement	The end user inside the virtual space of VR-Together Pilot 3 MUST be able to move (translation). 6DoF
FR.43.1	FR	43	1	OR	Derived actions	The end-user's actions inside the virtual space of VR-Together Pilot 3 MUST lead to changes in the storyline that is being projected
FR.44.1	FR	44	1	VRT	Pattern recognition interaction	The VR-Together Pilot 3 platform MUST support multi modal pattern recognition mechanics for changing the storyline according to user's choices
FR.45.1	FR	45	1	VRT	Pointing interaction	The VR-Together Pilot 3 platform MUST be able to recognize pointing gestures of end users and change the storyline accordingly
FR.46.1	FR	46	1	VRT	Speech interaction	The VR-Together Pilot 3 platform MUST be able to recognize the speech of end users and change the storyline accordingly
FR.48.1	FR	48	1	VRT	Interactive character	The system for Pilot 3 SHOULD integrate and use interactive character animation

FR.49.0	FR	49	0	VRT	Networks	The VR-Together platform MUST support bandwidth configuration options for the end user
FR.50.0	FR	50	0	VRT	Networks	The VR-Together platform MUST support delay constraint configuration options for the end user
FR.51.1	FR	51	1	VRT	Self-representation	The VR-Together platform MUST support self-representation projection configuration options for the end user.
FR.54.0	FR	54	0	VRT	VR Scenario	The VR-Together platform MUST allow one end user to see a dynamic projection of another end-user's body representation within the virtual space.
FR.55.0	FR	55	0	VRT	VR Scenario	The VR-Together platform MUST allow one end user to see the projection of another end-user's body representation positioned at various distances within the virtual space.
FR.57.0	FR	57	0	CA	RGB-D capture	The VR-Together hardware capturing component/system MUST capture RGB-D data from 4 RGB-D devices connected to 4 capturing nodes (RGB-D nodes)
FR.58.0	FR	58	0	CA	RGB-D Capture	The VR-Together hardware capturing component/system RGB-D devices SHOULD be automatically calibrated (extrinsic calibration).
FR.59.0	FR	59	0	CA	RGB-D Capture	The RGB-D frames from the RGB-D nodes MUST be synchronized and grouped in a central node, resulting in a RGB-D group frame.
FR.60.0	FR	60	0	CA	VR content formats	The VR-Together platform MUST process end-user’s live coloured 3D point cloud to reconstruct a 3D time-varying mesh in real-time.
NF.66.1	NF	66	1	CA	Latency	The input image captured by the hardware sensors of the capturing component MUST use a framerate of at least 25 fps.
NF.67.0	NF	67	0	CA	Face capture	The VR-Together hardware capturing component/system MUST capture the end-user’s face from at least two different sides.
NF.72.0	NF	72	0	EE	Compression	The VR-Together platform MUST be able to achieve a compression ratio of up to 1:10 in point cloud streams
NF.73.1	NF	73	1		Latency	The VR-Together platform MUST achieve an end to end (capture to projection) latency that is lower than TBD.
NF.74.1	NF	74	1	EE	Compression	The VR-Together platform SHOULD support point cloud compression of arbitrary topology (Topology TBD).
NF.75.0	NF	75	0	VRT	Quality assesment	The VR-Together platform SHOULD be able to evaluate the expected quality of experience according to the objective metrics TBD
NF.77.0	NF	77	0	EE	Compression	The VR-Together platform SHOULD be able to achieve a compression ratio of up to 1:30 for textured mesh (3D geometry and textures) content
NF.78.0	NF	78	0	EE	Compression	The VR-Together platform MUST support compression for textured 3D time varying mesh content of arbitrary topology.
NF.92.1	NF	92	1	OR	Configuration	The VR-Together platform orchestration module MUST be able to configure the native end-user play-out component
NF.95.0	NF	95	0	PL	VR content visual quality	The VR-Together platform MUST support playback of end-user's representation of at least 960x540 pixels

NF.96.1	NF	96	1	PL	VR Content Visual Quality	The VR-Together platform MUST support playback of end users represention at a framerate of at least TBD fps.
NF.99.0	NF	99	0	PL	WebVR	The VR-Together play-out component's web player MUST operate in a browser that supports WebVR and A-frame.
NF.104.1	NF	104	1	PL	VR Content Format	The VR-Together play-out component's native player MUST support the reproduction of hybrid VR contents (TBD what is the hybrid VR content) in virtual space.
NF.107.0	NF	107	0	PL	VR Content Visual Quality	The VR-Together play-out component's native player SHOULD be able to alter the lighting of specific objects within the virtual space, on the basis of custom shaders.
NF.109.1	NF	109	1	PL	VR Content Visual Quality	The VR-Together play-out component's native player stereo effective display resolution MUST be up to 4K.
NF.110.1	NF	110	1	VRT	Latency	The VR-Together play-out component's native player self representation projection MUST have latency under TBD.
NF.111.1	NF	111	1	PL	Synchronization	The VR-Together play-out component's native player SHOULD support synchronization between different input formats with less than TBD of delay
NF.112.1	NF	112	1	PL	Synchronization	The VR-Together play-out component's different players SHOULD support synchronization of frame accurate with a delay lower than TBD.
FR.115.0	FR	115	0	CA	Quality Assesment	The VR-Together platform capturing component SHOULD record and store the recordings of the HMD for further furture analysis purposes.
FR.117.0	FR	117	0	OR	Configuration	The VR-Together platform orchestration component MUST support remote operation.
FR.118.0	FR	118	0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 1, MUST manage sessions where 2 end users participate in a virtual space.
FR.119.0	FR	119	0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, MUST manage sessions where at least 2 end users participate in a virtual space.
FR.120.0	FR	120	0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 1, SHOULD support at least one session.
FR.121.0	FR	121	0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, SHOULD support at least two parallel sessions.
NF.122.0	NF	122	0	CA	RGB-D Capture Framerate	The VR-Together hardware capturing component/system MUST achieve a capture rate of at least 25 fps.
NF.123.0	NF	123	0	CA	Latency	The VR-Together platform MUST perform the People live 3d reconstruction with a delay lower than 80ms.
FR.124.0	FR	124	0	CA	VR experience	The VR-Together hardware capturing component/system MUST store the captured end-user's face data. The information must be stored (on disk or in memory) and must be accessible in real-time by the face inpainting algorithm.
FR.125.0	FR	125	0	CA	Benchmarking	The VR-Together platform MUST record the postion of the end user in the 3D scene at regular time intervals.

FR.126.0	FR	126	0	CA	Benchmarking	The VR-Together platform MUST record the viewport video visualized by each end user with timestamp.
FR.127.0	FR	127	0	CA	Benchmarking	The VR-Together platform MUST record the audio information (speech) from the end user with timestamp.
FR.128.0	FR	128	0	OR	VR Scenario	The VR-Together Pilot 1 platform MUST allow for 2 end users to simultaneously be in the same virtual space.
FR.129.0	FR	129	0	OR	VR Scenario	The VR-Together Pilot 2 platform MUST allow for 4 end users to simultaneously be in the same virtual space.
FR.130.0	FR	130	0	OR	VR Scenario	The VR-Together Pilot 3 platform MUST allow for 10 end users to simultaneously be in the same virtual space.
FR.132.0	FR	132	0	VRT	Compression	The VR-Together platform MUST support TVM compression configuration options for the end user.
FR.134.0	FR	134	0	PL	VR Content Visual Quality	The VR-Together play-out component's native player SHOULD be able to alter the lighting of specific objects within the virtual space, on the basis of custom shaders.
FR.135.0	FR	135	0	PL	VR Experience	The VR-Together play-out component's native player MUST be able to reproduce content adapted to 3DoF or 3DoF+ movements.
FR.136.0	FR	136	0	EE	Compression	The VR-Together platform MUST use typical browser supported audio encoding.
FR.137.0	FR	137	0	EE	Compression	The VR-Together platform MUST use typical browser supported video encoding.
FR.138.0	FR	138	0	EE	Compression	The VR-Together platform MUST use typical browser supported audio encapsulation.
FR.139.0	FR	139	0	EE	Compression	The VR-Together platform MUST use typical browser supported video encapsulation.
FR.140.0	FR	140	0	PL	VR Content	The VR-Together platform web player MUST support playback of 2D VR video content.
FR.141.0	FR	141	0	PL	VR Content	The VR-Together platform web player MUST support playback of 2D end-user representation projection.
FR.142.0	FR	142	0	PL	VR Content	The VR-Together play-out component platform SHOULD support spatial audio.
FR.143.0	FR	143	0	PL	VR Content	The VR-Together play-out component MUST support input of separate VR content and end-user representations streams.
FR.144.0	FR	144	0	PL	Network	The VR-Together play-out component's web player SHOULD support content bandwidth adaptation.
FR.145.0	FR	145	0	CA	Synchronization	The VR-Together platform capturing component MUST timestamp media content in relation to a platform-wide common clock.
FR.146.0	FR	146	0	PL	VR Content	The native player MUST support play-out of content for Point Clouds.
FR.147.0	FR	147	0	PL	VR Content	The native player MUST support play-out of content for Static/Dynamic meshes.

FR.148.0	FR	148	0	PL	VR Content	The native player MUST support play-out of content for mono/stereo 2d video.
FR.149.0	FR	149	0	PL	VR Content	The native player MUST support play-out of content for 360 video.
NF.150.0	NF	150	0	PL	VR Content Visual Quality	The VR-Together play-out component's native player mono effective display resolution MUST be up to 2K.
NF.151.0	NF	151	0	OR	Configuration	The VR-Together platform orchestration module SHOULD be able to configure the web-client end-user play-out component.
FR.152.0	FR	152	0	PL	Network	The Point Clouds (PC) used to represent the VR together end user content COULD be transmitted using an an adaptive bitrate streaming technique.
NF.153.0	NF	153	0	PL	Network	The adaptive bitrate streaming technique used SHOULD be the Dynamic Adaptive Streaming over HTTP (DASH)
NF.154.0	NF	154	0	PL	Network	NF.154 The DASH adapatation set SHOULD include a low quality version of the PC user representation, for the DASH client to download it when the bandwidth conditions are bad
NF.155.0	NF	155	0	PL	Network	NF.155 The DASH server COULD provide (as an option) a tile-based adaptation set, for the DASH client to download only the portion of the PC that falls in her/his field of view.
FR.156.0	FR	156	0	VRT	VR content	End users SHOULD be able to see photorealistic Live content (mono or stereoscopic video) in the VR environment

Table 3: Current status of the Requirements Matrix

Experiment Id	Linked Non-Functional Requirements	Linked Functional Requirements
i2CAT-2.1	NF.104.1, NF.107.0, NF.109.1, NF.150.0, NF.66.1, NF.96.1, NF.122.0	FR.1.0,FR.2.0,FR.3.0,FR.5.0,FR.15.1, FR.16.0, FR.17.0, FR.27.1, FR.28.0, FR.31.0, FR.37.0
Artanim-2.1	NF.96.1, NF.104.1, NF.107.0, NF.109.1, NF.122.0, NF.150.0	FR.3.0, FR.9.1, FR.11.0, FR.12.1, FR.13.0, FR.22.0, FR.27.1, FR.31.0, FR.35.0, FR.36.0, FR.37.0, FR.135.0
CERTH-2.1	NF.96.1, NF.122.0 NF.55.0, NF57.0, NF.58.0, NF.59.0, NF.77.0, NF.78.0, NF.123.0	FR5.0. FR12.0, FR15.0, FR.35.0, FR.147.0, FR.119.0, FR129.0
CWI-2.1	NF.75.0, NF.95.0, NF.72.0, NF.74.1, NF.77.0, NF.78.0, NF.154.0	FR.1.0, FR.3.0, FR.5.0, FR.16.0, FR.20.0
CERTH-2.2	NF.66.1, NF.67.0, NF.96.1, NF.122.0	FR.1.0, FR.5.0, FR.12.1, FR.16.0,FR.57.0
CWI-2.2	NF.75.0, NF.95.0, NF.152.0, NF.153.0, NF.154.0, NF.155.0	FR.3.0, FR.5.0, FR.16.0
CERTH-2.3	NF.75.0, NF.66.0, NF73.0, NF77.0, NF78.0, NF123.0	FR.1.0, FR.5., FR.16.00, FR.60.0
VO-2.1	NF.66.1, NF.92.1, NF.151.0	FR.1.0, FR.5.0, FR.49.0, FR.3.0, FR.23.1, FR.25.0
TNO-2.3	NF.66.1, NF.73.1, NF.123.0, NF.153.0, NF.151.0	FR.8.0, FR.13.0, FR.15.1, FR.49.0, FR.50.0, FR.51.1, FR.54.0 and FR.129.0
Artanim-2.2	N/A	FR.1.0, FR.5.0, FR.16.0
CERTH-2.4	NF.67.0, NF.75.0	FR.3.0
TNO-2.1	NF.66.1, NF.67.0, NF.75.0, NF.96.1, NF.122.0, NF.123.0	FR.1.0, FR.5.0, FR.12.1, FR.50.0, FR.51.1
TNO-2.2:	NF.66.1, NF.67.0, NF.73.1, NF.75.0, NF.96.1, NF.110.1, NF.111.1, NF.112.1, NF.122.0, NF.123.0	FR.1.0, FR.5.0, FR.8.0, FR.12.1, FR.13.0, FR.15.1, FR.50.0, FR.51.1
CERTH-2.5	NF.55.0, NF57.0, NF.58.0, NF.59.0, NF.77.0, NF.78.0, NF.123.0	FR.13.0, FR5.0, FR12.0, FR15.0, FR.35.0, FR.147.0, FR.119.0, FR129.0

CWI-2.5	NF.96.1	FR.1.0, FR.16.0
EXP-VO-3.1	NF.66.1, NF.92.1, NF.151.0	FR.1.0, FR.5.0, FR.49.0, FR.3.0, FR.23.1, FR.25.0
CERTH-3.1	NF.66.1, NF.92.1, NF.151.0	FR.1.0, FR.5.0, FR.49.0, FR.3.0, FR.23.1, FR.25.0
CERTH-3.2	NF.96.1, NF.122.0 NF.55.0, NF57.0, NF.58.0, NF.59.0, NF.77.0, NF.78.0, NF.123.0	FR5.0. FR12.0, FR15.0, FR.35.0, FR.147.0, FR.119.0, FR129.0
CERTH-3.3	NF.75.0, NF.66.0, NF73.0, NF77.0, NF78.0, NF123.0	FR.1.0, FR.5., FR.16.00, FR.60.0
i2CAT-3.1	NF.67.0, NF.96.1, NF.122.0	FR.12.1, FR.57.0
i2CAT-3.2	NF.111.1, NF.112.1	FR.1.0, FR.16.0, FR.59.0, FR.145.0
i2CAT-3.3	NF.111.1, NF.112.1	FR.1.0, FR.16.0, FR.59.0, FR.145.0
I2CAT-3.4	NF.66.1, NF.73.1, NF.123.0, NF.153.0, NF.151.0	FR.8.0, FR.13.0, FR.15.1, FR.49.0, FR.50.0, FR.51.1, FR.54.0 and FR.129.0
CWI-3.1	NF.75.0, NF.95.0, NF.72.0, NF.74.1, NF.77.0, NF.78.0	FR.3.0, FR.5.0, FR.16.0, FR.20.0
CWI-3.2	NF.74.1, NF.78.0, NF.155.0, NF.152.0, NF.153.0, NF.154.0	FR.23.1
CWI-3.3	NF.73.1	FR.8.0, FR.13.0, FR.50.0
TNO-3.1	NF.99.0, NF.151.0	FR.25.0, FR.140.0, FR.141.0, FR.144.0

Table 4: Linking between experiments and Requirements

4.3. Requirements status Update for Final Delivery

During the last year of development, shaping the final VR-Together product, the technology has been adapted to the needs of the market, in particular to the specific necessity of bringing remote people together, in an immersive and more natural way; issue particularly highlighted by the current situation due to the spreading of the COVID-19 virus and the consequent lockdown and restrictions raised in many countries.

For the above reasons the VR-Together consortium has attacked the problem improving the platform and including specific functionalities with the following objectives:

- Have a scalable solution that do not requires a volumetric video capturing system
- Provide access to users with traditional 2D webcams
- Provide access to users with 2D screens (laptops and desktop PCs)
- Reach a broader audience with lower end devices

The following list of requirements represents the last version, updated from the one presented in Table 3, where i) a new set of requirements have been introduced in order to define the technical objectives of the **Proof of Concept** including the aforementioned new functionalities and ii) the old requirements specifically defined for the volumetric video representations have been updated according to the presence of the new available representations.

4.3.1. Proof of Concept Requirements Analysis and Definition

The definition of the new requirements, together with the adaptation of the previous ones, has followed a specific, and reviewed, process similar to every updates of the requirements produced in VR-Together (see D2.1 and D2.2). For this particular case, the previously available requirements have been

- Initially analysed, in order to decide if the definition needed an update or not.
- The requirements identified as to be updated have been provided with a new description
- A new category has been defined (Non-Volumetric Users)
- The new functionalities that were not mentioned or considered have been listed as new requirements
- The specific non-functional requirements have been included for each functionality

Table 5 shows the new requirements updated according to the aforementioned explanation.

ID	Component	Title	Previous Description	New Description Needed?	New Description
FR.1.0	PL	Self representation	An end user MUST be able to see his own representation in the virtual space of VR-Together	Yes	An end user **with volumetric capturing system** MUST be able to see his own representation in the virtual space of VR-Together
FR.2.0	PL	Users audio representation	An end user MUST be able to hear the sounds made by another user in the virtual space of VR Together	Yes	An end user MUST be able to hear the sounds made by **other users with audio chat activated** in the virtual space of VR Together
FR.3.0	PL	Users representation	An end user MUST be able to see the visual representation of another user in the virtual space of VR Together	Yes	An end user MUST be able to see the visual representation of another user **with volumetric or 2D capturing system** in the virtual space of VR Together
FR.4.0	CA	Audio Capturing setup	A location where the VR-Together platform's capturing setup is deployed MUST capture the audio generated by the user	Yes	A location where the VR-Together platform's capturing setup is deployed MUST capture the audio generated by the user **if the user representation allows it**
FR.5.0	CA	Visual Capturing setup	A location where VR-Together platform's capturing setup is deployed MUST capture the visual representation of the user	Yes	A location where VR-Together platform's capturing setup is deployed MUST capture the visual representation of the user **if the user representation allows it**
FR.8.0	DE	Latency	An end user MUST have a network latency allowing for seamless and natural communication and interaction with other users in the virtual space of VR-Together	No	N/A
FR.9.1	OR	VR Scenario	An end-user client MUST be able to create a reconstruction of the virtual space of VR-Together.	No	N/A

FR.11.0	VRT	VR content formats	End users SHOULD be able to see different examples of VR content formats	No	N/A
FR.12.1	VRT	VR content visual quality	End users MUST be able to see photorealistic VR contents	No	N/A
FR.13.0	VRT	Synchronization	End users in distributed locations sharing a virtual space MUST be able to see the same VR content at the same time	No	N/A
FR.15.1	PL	VR content visual quality	End users SHOULD see other users seamlessly blended in the virtual space of VR-Together. The Seamlessness evaluation will be performed by TBD.	No	N/A
FR.16.0	VRT	VR Experience	End users SHOULD feel comfort in being immersed in the virtual space of VR-Together, at least for the duration of the pilot experience	No	N/A
FR.17.0	VRT	VR Experience	An end user SHOULD have an experience that visually and acoustically allows to perceive and understand the other participants' body language expressions.	No	N/A
FR.18.0	PL	VR Content	The VR audio content MUST be directional giving the perception of point sources within the virtual space of VR-Together.	No	N/A

FR.20.0	VRT	End-user devices	End users MUST be able to access the VR-Together platform by using commercially available HMDs and capture systems	No	N/A
FR.22.0	PL	VR content visual quality	End users, scene of action and characters SHOULD be able to be projected in the virtual space of VR-Together using different media formats. The resulting VR image should be a blend of different formats.	No	N/A
FR.23.1	DE	Networks	The data transmission within VR-Together MUST be using commercial communication (e.g. MPEG-DASH) and media delivery networks (e.g. CDNs)	No	N/A
FR.24.0	EE	Networks	Media streams SHOULD provide adaptive quality to network, device and interface capabilities	No	N/A
FR.25.0	VRT	Web interface	End users MUST be able to access the VR-Together platform using a web application.	No	N/A
FR.26.0	VRT	Native interface	End users MUST be able to access VR-Together platform using a native application	No	N/A
FR.27.1	VRT	VR Content Visual Quality	The virtual character representation MUST be detailed enough to allow for the recognition of facial expressions.	No	N/A

FR.28.0	VRT	VR experience	The level of detail of end-user representation in the virtual space of VR-Together MUST allow the recognition of facial expressions.	Yes	The level of detail of **volumetric** end-user representation in the virtual space of VR-Together MUST allow the recognition of facial expressions.
FR.30.1	PL	VR Content	The VR-Together platform MUST be able to display the VR content which, depending on the configuration, can be either i) local or ii) stored in a network server	No	N/A
FR.31.0	VRT	VR Content	Illumination MUST be consistent in the whole experience	No	N/A
FR.32.1	VRT	VR Content	The representations of the rendered characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget their gaze according to the end-user's viewpoint	No	N/A
FR.33.1	VRT	VR Content	The representations of the rendered pre-rigged characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget pointing gestures	No	N/A
FR.35.0	PL	VR content visual quality	The representations of the rendered characters inside the virtual space of VR-Together MUST have parallax and depth to allow for a 3D representation.	No	N/A

FR.36.0	PL	VR content visual quality	The end-user inside the virtual space of VR-Together MUST be able to perceive the 3D appearance of the characters (parallax, depth)	No	N/A
FR.37.0	VRT	VR Experience	The end-user inside the virtual space of VR-Together MUST be able to rotate their head and have certain level of translation capacity while seated (3DoF+)	No	N/A
FR.41.1	OR	Active watch	The end-user inside the virtual space of VR-Together Pilot 3 MUST be able to become a character within the storyline that is being projected	No	N/A
FR.42.1	VRT	Movement	The end-user inside the virtual space of VR-Together Pilot 3 MUST be able to move (translation). 6DoF	No	N/A
FR.43.1	OR	Derived actions	The end-user's actions inside the virtual space of VR-Together Pilot 3 MUST lead to changes in the storyline that is being projected	No	N/A
FR.44.1	VRT	Pattern recognition interaction	The VR-Together Pilot 3 platform MUST support multi modal pattern recognition mechanics for changing the storyline according to user's choices	No	N/A

FR.45.1	VRT	Pointing interaction	The VR-Together Pilot 3 platform MUST be able to recognize pointing gestures of end-users and change the storyline accordingly	No	N/A
FR.46.1	VRT	Speech interaction	The VR-Together Pilot 3 platform MUST be able to recognize the speech of end-users and change the storyline accordingly	No	N/A
FR.48.1	VRT	Interactive character	The system for Pilot 3 SHOULD integrate and use interactive character animation	No	N/A
FR.49.0	VRT	Networks	The VR-Together platform MUST support bandwidth configuration options for the end user	No	N/A
FR.50.0	VRT	Networks	The VR-Together platform MUST support delay constraint configuration options for the end user	No	N/A
FR.51.1	VRT	Self representation	The VR-Together platform MUST support self-representation projection configuration options for the end user.	Yes	The VR-Together platform MUST support self-representation projection configuration options for the end user **with volumetric capturing systems**
FR.54.0	VRT	VR Scenario	The VR-Together platform MUST allow one end user to see a dynamic projection of another end-user's body representation within the virtual space.	Yes	The VR-Together platform MUST allow one end user to see a dynamic projection of another end-user's body representation within the virtual space **when the other end user has a volumetric capturing systems**
FR.55.0	VRT	VR Scenario	The VR-Together platform MUST allow one end user to see the	Yes	The VR-Together platform MUST allow one end user to see the projection of another end-user's body representation

			projection of another end-user's body representation positioned at various distances within the virtual space.		positioned at various distances within the virtual space **when the other end user has a volumetric capturing systems**
FR.57.0	CA	RGB-D capture	The VR-Together hardware capturing component/system MUST capture RGB-D data from 4 RGB-D devices connected to 4 capturing nodes (RGB-D nodes)	Yes	The VR-Together hardware **full volumetric** capturing component/system MUST capture RGB-D data from 4 RGB-D devices connected to 4 capturing nodes (RGB-D nodes)
FR.58.0	CA	RGB-D Capture	The VR-Together hardware capturing component/system RGB-D devices SHOULD be automatically calibrated (extrinsic calibration).	No	N/A
FR.59.0	CA	RGB-D Capture	The RGB-D frames from the RGB-D nodes MUST be synchronized and grouped in a central node, resulting in a RGB-D group frame.	No	N/A
FR.60.0	CA	VR content formats	The VR-Together platform MUST process end-user's live coloured 3D point cloud to reconstruct a 3D time-varying mesh in real-time.	No	N/A
NF.66.1	CA	Latency	The input image captured by the hardware sensors of the capturing component MUST use a framerate of at least 25 fps.	No	N/A
NF.67.0	CA	Face capture	The VR-Together hardware capturing component/system	No	N/A

			MUST capture the end-user's face from at least two different sides.		
NF.72.0	EE	Compression	The VR-Together platform MUST be able to achieve a compression ratio of up to 1:10 in point cloud streams	No	N/A
NF.73.1		Latency	The VR-Together platform MUST achieve an end to end (capture to projection) latency that is lower than TBD.	No	N/A
NF.74.1	EE	Compression	The VR-Together platform SHOULD support point cloud compression of arbitrary topology (Topology TBD).	No	N/A
NF.75.0	VRT	Quality assesment	The VR-Together platform SHOULD be able to evaluate the expected quality of experience according to the objective metrics TBD	No	N/A
NF.77.0	EE	Compression	The VR-Together platform SHOULD be able to achieve a compression ratio of up to 1:30 for textured mesh (3D geometry and textures) content	No	N/A
NF.78.0	EE	Compression	The VR-Together platform MUST support compression for textured 3D time varying mesh content of arbitrary topology.	No	N/A
NF.92.1	OR	Configuration	The VR-Together platform orchestration module MUST be	No	N/A

			able to configure the native end-user play-out component		
NF.95.0	PL	VR content visual quality	The VR-Together platform MUST support playback of end-user's representation of at least 960x540 pixels	No	N/A
NF.96.1	PL	VR Content Visual Quality	The VR-Together platform MUST support playback of end users represention at a framerate of at least TBD fps.	No	N/A
NF.99.0	PL	WebVR	The VR-Together play-out component's web player MUST operate in a browser that supports WebVR and A-frame.	No	N/A
NF.104.1	PL	VR Content Format	The VR-Together play-out component's native player MUST support the reproduction of hybrid VR contents (combination of media formats considered in the project) in virtual space.	No	N/A
NF.107.0	PL	VR Content Visual Quality	The VR-Together play-out component's native player SHOULD be able to alter the lighting of specific objects within the virtual space, on the basis of custom shaders.	No	N/A
NF.109.1	PL	VR Content Visual Quality	The VR-Together play-out component's native player stereo	No	N/A

			effective display resolution MUST be up to 4K.		
NF.110.1	VRT	Latency	The VR-Together play-out component's native player self-representation projection MUST have latency under TBD.	Yes	The VR-Together play-out component's native player self-representation projection **for users with volumetric representation** MUST have latency under TBD.
NF.111.1	PL	Synchronization	The VR-Together play-out component's native player SHOULD support synchronization between different input formats with less than TBD of delay	No	N/A
NF.112.1	PL	Synchronization	The VR-Together play-out component's different players SHOULD support synchronization of frame accurate with a delay lower than TBD.	No	N/A
FR.115.0	CA	Quality Assesment	The VR-Together platform capturing component SHOULD record and store the recordings of the HMD for further furture analysis purposes.	No	N/A
FR.117.0	OR	Configuration	The VR-Together platform orchestration component MUST support remote operation.	No	N/A
FR.118.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 1, MUST manage sessions where 2 end-users participate in a virtual space.	No	N/A

FR.119.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, MUST manage sessions where at least 2 end-users participate in a virtual space.	No	N/A
FR.120.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 1, SHOULD support at least one session.	No	N/A
FR.121.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, SHOULD support at least two parallel sessions.	No	N/A
NF.122.0	CA	RGB-D Capture Framerate	The VR-Together hardware capturing component/system MUST achieve a capture rate of at least 25 fps.	No	N/A
NF.123.0	CA	Latency	The VR-Together platform MUST perform the People live 3d reconstruction with a delay lower than 80ms.	No	N/A
FR.124.0	CA	VR experience	The VR-Together hardware capturing component/system MUST store the captured end-user's face data. The information must be stored (on disk or in memory) and must be accessible in real-time by the face inpainting algorithm.	No	N/A

FR.125.0	CA	Benchmarking	The VR-Together platform MUST record the postion of the end user in the 3D scene at regular time intervals.	No	N/A
FR.126.0	CA	Benchmarking	The VR-Together platform MUST record the viewport video visualized by each end user with timestamp.	No	N/A
FR.127.0	CA	Benchmarking	The VR-Together platform MUST record the audio information (speech) from the end user with timestamp.	No	N/A
FR.128.0	OR	VR Scenario	The VR-Together Pilot 1 platform MUST allow for 2 end users to simultaneously be in the same virtual space.	No	N/A
FR.129.0	OR	VR Scenario	The VR-Together Pilot 2 platform MUST allow for 4 end users to simultaneously be in the same virtual space.	No	N/A
FR.130.0	OR	VR Scenario	The VR-Together Pilot 3 platform MUST allow for 10 end users to simultaneously be in the same virtual space.	No	N/A
FR.132.0	VRT	Compression	The VR-Together platform MUST support TVM compression configuration options for the end-user.	No	N/A

FR.134.0	PL	VR Content Visual Quality	The VR-Together play-out component's native player SHOULD be able to alter the lighting of specific objects within the virtual space, on the basis of custom shaders.	No	N/A
FR.135.0	PL	VR Experience	The VR-Together play-out component's native player MUST be able to reproduce content adapted to 3DoF or 3DoF+ movements.	No	N/A
FR.136.0	EE	Compression	The VR-Together platform MUST use typical browser supported audio encoding.	No	N/A
FR.137.0	EE	Compression	The VR-Together platform MUST use typical browser supported video encoding.	No	N/A
FR.138.0	EE	Compression	The VR-Together platform MUST use typical browser supported audio encapsulation.	No	N/A
FR.139.0	EE	Compression	The VR-Together platform MUST use typical browser supported video encapsulation.	No	N/A
FR.140.0	PL	VR Content	The VR-Together platform web player MUST support playback of 2D VR video content.	No	N/A
FR.141.0	PL	VR Content	The VR-Together platform web player MUST support playback of	No	N/A

			2D end-user representation projection.		
FR.142.0	PL	VR Content	The VR-Together play-out component platform SHOULD support spatial audio.	No	N/A
FR.143.0	PL	VR Content	The VR-Together play-out component MUST support input of separate VR content and end-user representations streams.	No	N/A
FR.144.0	PL	Network	The VR-Together play-out component's web player SHOULD support content bandwidth adaptation.	No	N/A
FR.145.0	CA	Synchronization	The VR-Together platform capturing component MUST timestamp media content in relation to a platform-wide common clock.	No	N/A
FR.146.0	PL	VR Content	The native player MUST support play-out of content for Point Clouds.	No	N/A
FR.147.0	PL	VR Content	The native player MUST support play-out of content for Static/Dynamic meshes.	No	N/A
FR.148.0	PL	VR Content	The native player MUST support play-out of content for mono/stereo 2d video.	No	N/A
FR.149.0	PL	VR Content	The native player MUST support play-out of content for 360 video.	No	N/A

NF.150.0	PL	VR Content Visual Quality	The VR-Together play-out component's native player mono effective display resolution MUST be up to 2K.	No	N/A
NF.151.0	OR	Configuration	The VR-Together platform orchestration module SHOULD be able to configure the web-client end-user play-out component.	No	N/A
FR.152.0	PL	Network	The Point Clouds (PC) used to represent the VR together end user content COULD be transmitted using an adaptive bitrate streaming technique.	No	N/A
NF.153.0	PL	Network	The adaptive bitrate streaming technique used SHOULD be the Dynamic Adaptive Streaming over HTTP (DASH)	No	N/A
NF.154.0	PL	Network	NF.154 The DASH adaptation set SHOULD include a low quality version of the PC user representation, for the DASH client to download it when the bandwidth conditions are bad	No	N/A
NF.155.0	PL	Network	NF.155 The DASH server COULD provide (as an option) a tile-based adaptation set, for the DASH client to download only the portion of the PC that falls in her/his field of view.	No	N/A

FR.156.0	VRT	VR content	End users SHOULD be able to see photorealistic Live content (mono or stereoscopic video) in the VR environment	No	N/A
FR.157.0	PoC	Non Volumetric Users	End-users SHOULD be able to access the virtual space of VR-Together as webcam users	N/A	N/A
FR.158.0	PoC	Non Volumetric Users	Webcam users MUST be able to navigate the virtual space of VR-Together with 6 degrees of freedom	N/A	N/A
FR.159.0	PoC	Non Volumetric Users	Webcam users MUST be able to receive listen and render any audiovisual content included in the virtual space of VR-Together	N/A	N/A
FR.160.0	PoC	Non Volumetric Users	End-users SHOULD be able to access the virtual space of VR-Together as spectator users	N/A	N/A
FR.161.0	PoC	Non Volumetric Users	Spectator users MUST be able to navigate the virtual space of VR-Together with 6 degrees of freedom	N/A	N/A
FR.162.0	PoC	Non Volumetric Users	Spectator users MUST be able to receive listen and render any audiovisual content included in the virtual space of VR-Together	N/A	N/A
FR.163.0	PoC	Non Volumetric Users	End-users SHOULD be able to access the virtual space of VR-Together with no representation	N/A	N/A

FR.164.0	PoC	Non Volumetric Users	No representation users MUST be able to navigate the virtual space of VR-Together with 6 degrees of freedom	N/A	N/A
FR.165.0	PoC	Non Volumetric Users	No representation users MUST be able to receive listen and render any audiovisual content included in the virtual space of VR-Together	N/A	N/A
FR.166.0	PoC	Non Volumetric Users	Webcam users MUST ll be represented as an avatar including a 2D video capable to render their representation captured through a webcam	N/A	N/A
FR.167.0	PoC	Non Volumetric Users	Spectator users MUST be represented as a floating avatar including their name	N/A	N/A
FR.168.0	PoC	Non Volumetric Users	No representation users MUST not be represented	N/A	N/A
FR.169.0	PoC	Non Volumetric Users	The VR-Together application MUST support a minimum of X webcam users	N/A	N/A
FR.170.0	PoC	Non Volumetric Users	The VR-Together application MUST support a minimum of X spectator users	N/A	N/A
FR.171.0	PoC	Non Volumetric Users	The VR-Together application MUST support a minimum of X no representation users	N/A	N/A

Table 5: Last Requirements Update including the new functionalities

5. Architecture

This section partially resumes the section 5 of the D2.2 deliverable as the whole Pilot 2 components and modules are kept and planned to be deployed in the Pilot 3 platform development. The Pilot 3 novelties are depicted additively in order to present major updates regarding the last platform architecture definition to be aligned with the Requirements work related with Pilot 3 new functionalities.

5.1. Software architecture

5.1.1. Pilot 1 & Pilot 2

This section describes the architecture definitions of the Pilot 1 and Pilot 2 milestones delivered in the document D2.1 and D2.2.

Since the Pilot 1 architecture definition, the VR-Together platform is described in a traditional production to consumption chain; audio-visual information flows from capturing to playout are portrayed together with the additional functional components. The modules and components form the current Software Platform of VR-Together.

Pilot 1 work resulted in one Pipeline involving 2 users connected by a transmission chain starting from the capture component sending streams through encoding and delivery. This first version of the pipeline architecture was especially referring to:

- A “**component**” as a conceptual entity related to a general task within the end-to-end communication system. We identify the following five components/general tasks:
    - Capturing
    - Encoding & Encapsulation
    - Delivery
    - Orchestration
    - Play-Out
- A “**module**” as a building block that performs a specific technical task. To perform the general task described by a component, multiple modules are needed. The modules within each component are listed and described hereafter.

The whole Pilot 1 pipeline architecture is visible in the following figure (large scale format visible in annexes):

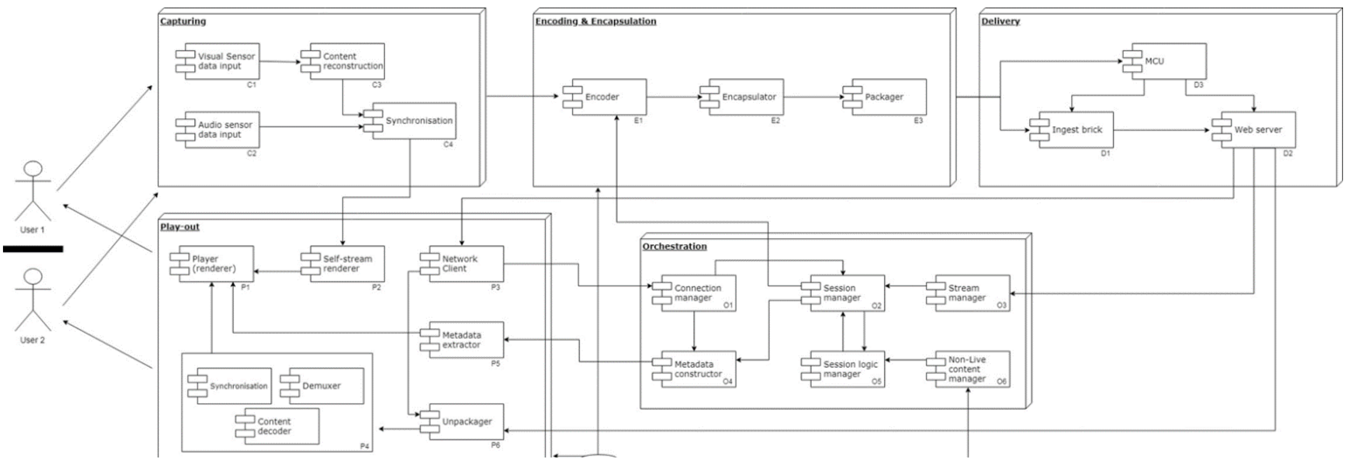


Figure 10: Pilot 1 software architecture definition

For Pilot 2, the requirements definition lead to involve multiple connected users (up to 4) as well as a live node delivering a live-news presenter stream. Iterated from the Pilot 1 definition, the Pilot 2 pipeline architecture introduced new entities:

- A "**user**" as a group of components related to the end-user client definition.
  A user entity instance is considered independent and replicable N times to grant platform scalability. We consider being located on the end-user client side the three components:
    - Capturing
    - Encoding and encapsulation
    - Playout
- A "**server**" as a generic entity group that designates centralized and remotely operational components that interact with each user entity. Although they can be hosted on different server location, we define the two components to be remotely accessible:
    - Delivery
    - Orchestration
- A "**live**" node as a remote entity dedicated to recording and streaming to connected users of the platform a presenter in live.

By combining all kind of entities into a unified platform network, the Pilot 2 architecture is constituted like so:

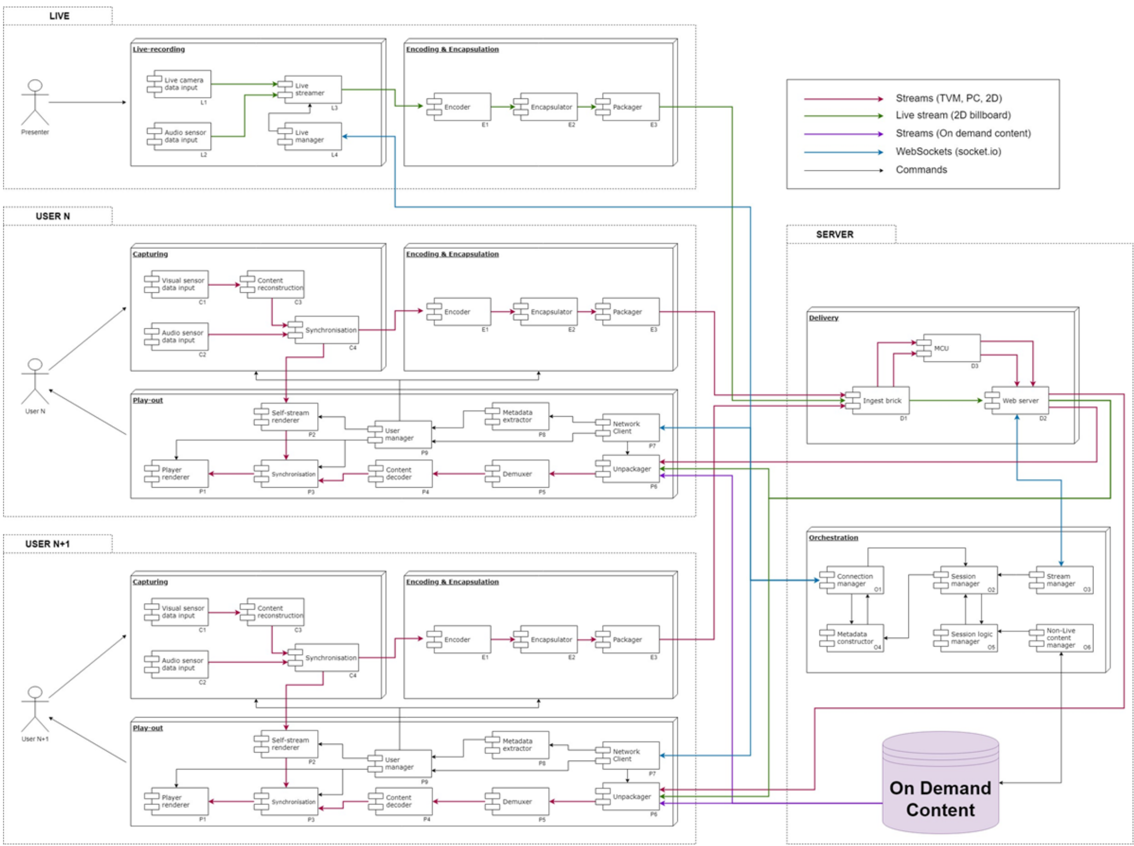


Figure 11: Pilot 2 Software architecture

5.1.1. Pilot 3

A major advance of the Pilot 3 definition is related to the ability to interact between all connected users with and within the virtual environment.

To do so, in this section, the interactivity system architecture is introduced and described to provide a comprehensive view of its operation as well as how the conceptual logics are planned in order to foresee how the system is going to be implemented.

Then these reflections are translated into the platform architecture to provide a definitive version that embeds all work done during Pilot 1 and Pilot 2 with the Pilot 3 advances.

5.1.1.1. Interactivity system architecture

Pilot 3 involves more interactions as each user is an active participant in the plot of the experience. The users' actions or inactions have a decisive role in the progress of the plot's playback and their behaviour and choices can influence where the story will go.

From a technical point of view, the interaction system needs to ensure that all users' actions can be completed in a same time and transmitted across all connected users. Synchronisation requirements needs to be considered in order to ensure that players data (hand positions and orientations, hand gestures, grasping, pointing) as well as interactive objects movements are correctly synchronised in a session.

The strategy plan to manage and transmit this wide of events consist in make one client as the main authority for anything needing decisions and use the Orchestration component to assign users' authority as well as act as a messaging/event service.

Consequently, new Pilot 3 technical requirements are introduced:

- Transmit the experience events, also considered as passive events to all connected users along the execution of the scenario
- Transmit the user events to all connected users in a session (actions, choices, objects handling)
- Transmit user gestures (hand positions and orientations, head position and orientation)
- Manage the scene interactions (lighting, objects)
- Define a client authority able to send the experience events transmission

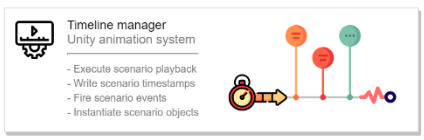
Then, these elements lead to identify flowing Architecture specifications:

- The Orchestrator attributes user roles in a session (authoritative vs standard users)
- The Orchestrator forwards events messages to all connected users in a session
- One authoritative user, i.e. "master user", sends the experience events
- Other users receive remote events

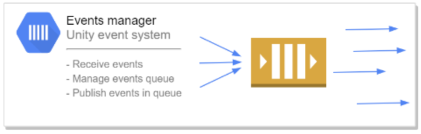
The impacts on the architectural definition of the platform are the introduction of additional modules allocated at the implementation and integration of the interactivity system.

These modules are listed as well as described hereafter and depicted in the following figures:

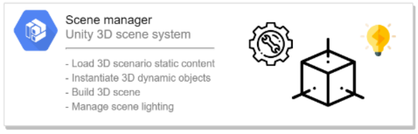
- [Playout] Timeline manager
  - Execute/Read scenario playback
  - Write scenario timestamp
  - Synch scenario playback
  - Fire/Catch scenario events
  - Instantiate scenario objects



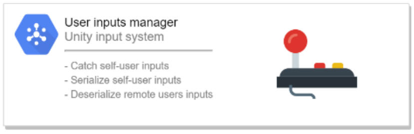
- [Playout] Events manager
  - Collect/Receive events
  - Manage events queue
  - Publish events in queue
  - Post events to Orchestrator



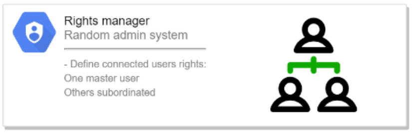
- [Playout] Scene manager
  - Load 3D scenario static content
  - Instantiate 3D dynamic objects
  - Build 3D scene
  - Manage scene lighting
  -



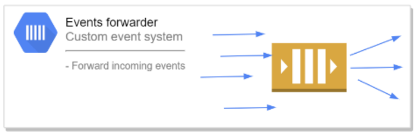
- [playout] Input Manager
  - Catch self-user input
  - Serialize self-user input
  - De-serialize remote users' inputs



- [Orchestration] Rights manager
  - Define connected users' right policy:
    - One master user
    - Others subordinated



- [Orchestration] Events forwarder
  - Distribute interactive events



From that, two types of user nodes can be constructed accordingly of technical requirements; one allocated for the authoritative user definition, the other for the subordinated users. Then, each node is connected with the Orchestration component where partial architecture overview (in the interactivity system scope) is depicted in the following figure:

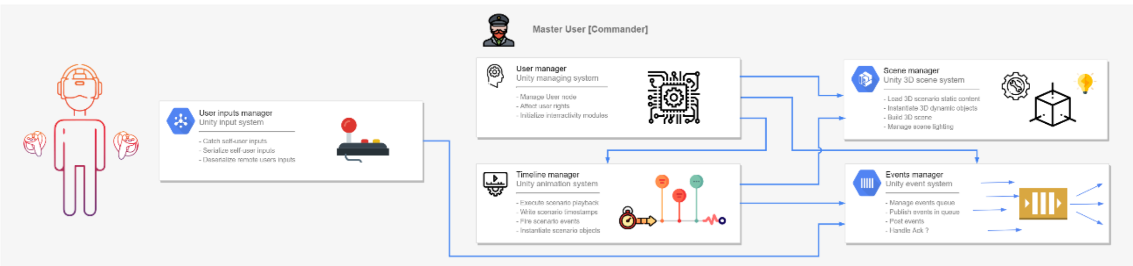


Figure 12: Authoritative user architecture definition

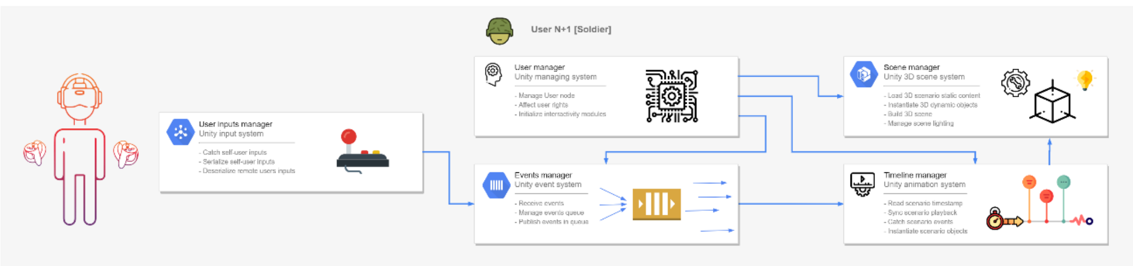


Figure 13: Subordinated user architecture definition

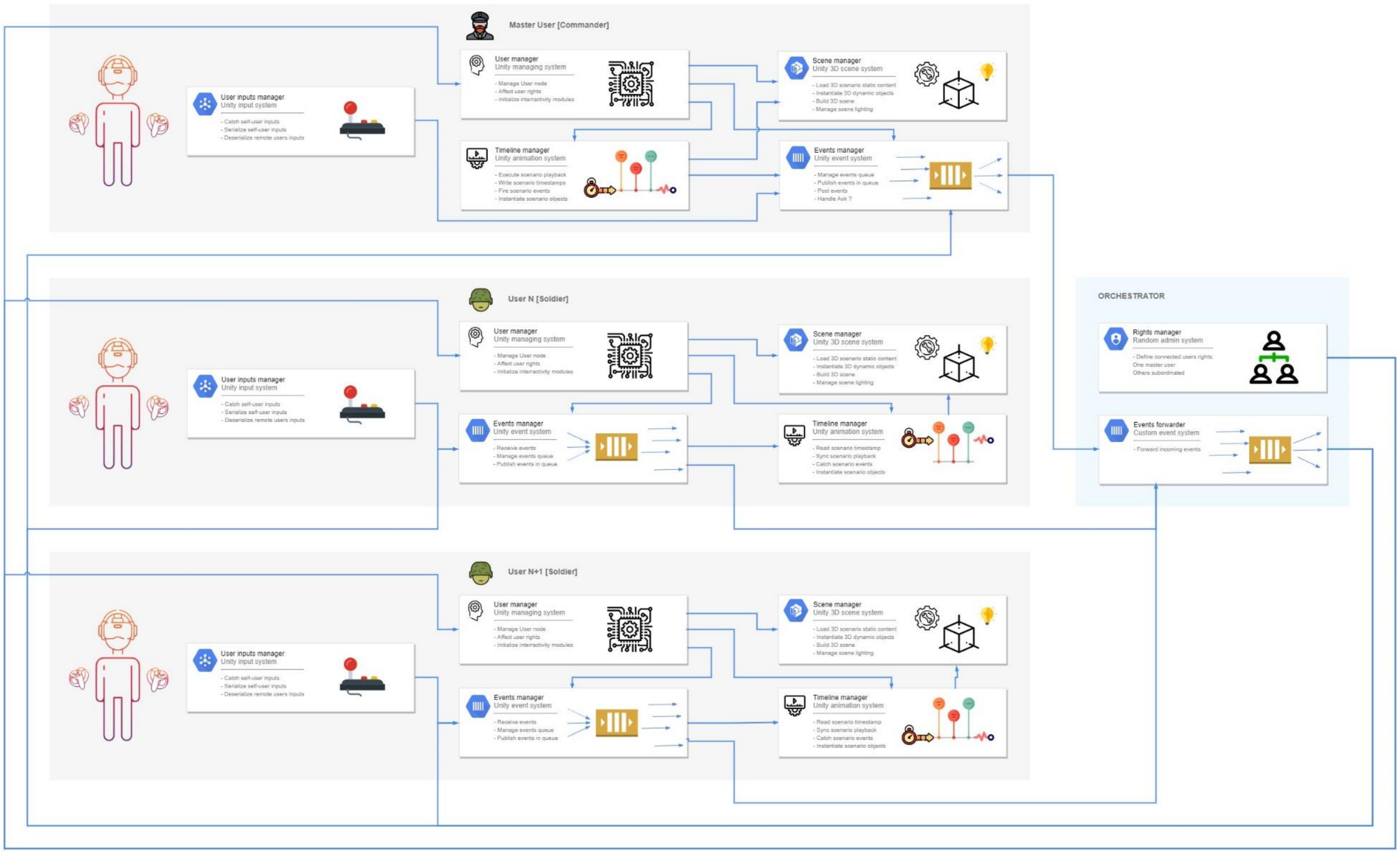


Figure 14: Pilot 3 architecture diagram

5.1.1.2. Platform architecture

The Pilot 3 platform definition is basically based on the Pilot 2 version where components and modules have been reorganised to integrate new modules allocated to the interactivity system presented in the previous section. These new modules are allocated to the interaction functionalities between users and virtual environment.

We can foresee 5 new modules which depend on the Playout and Orchestration components:

- O7 - Events listener
- O8 - Rights manager
- P10 - Timeline manager
- P11 - Scene manager
- P12 - Events manager
- P13 - Input manager

These modules are portrayed in the component architecture diagram in Figure 15 and take place into the whole component's definition in the following section 5.1.1.3.

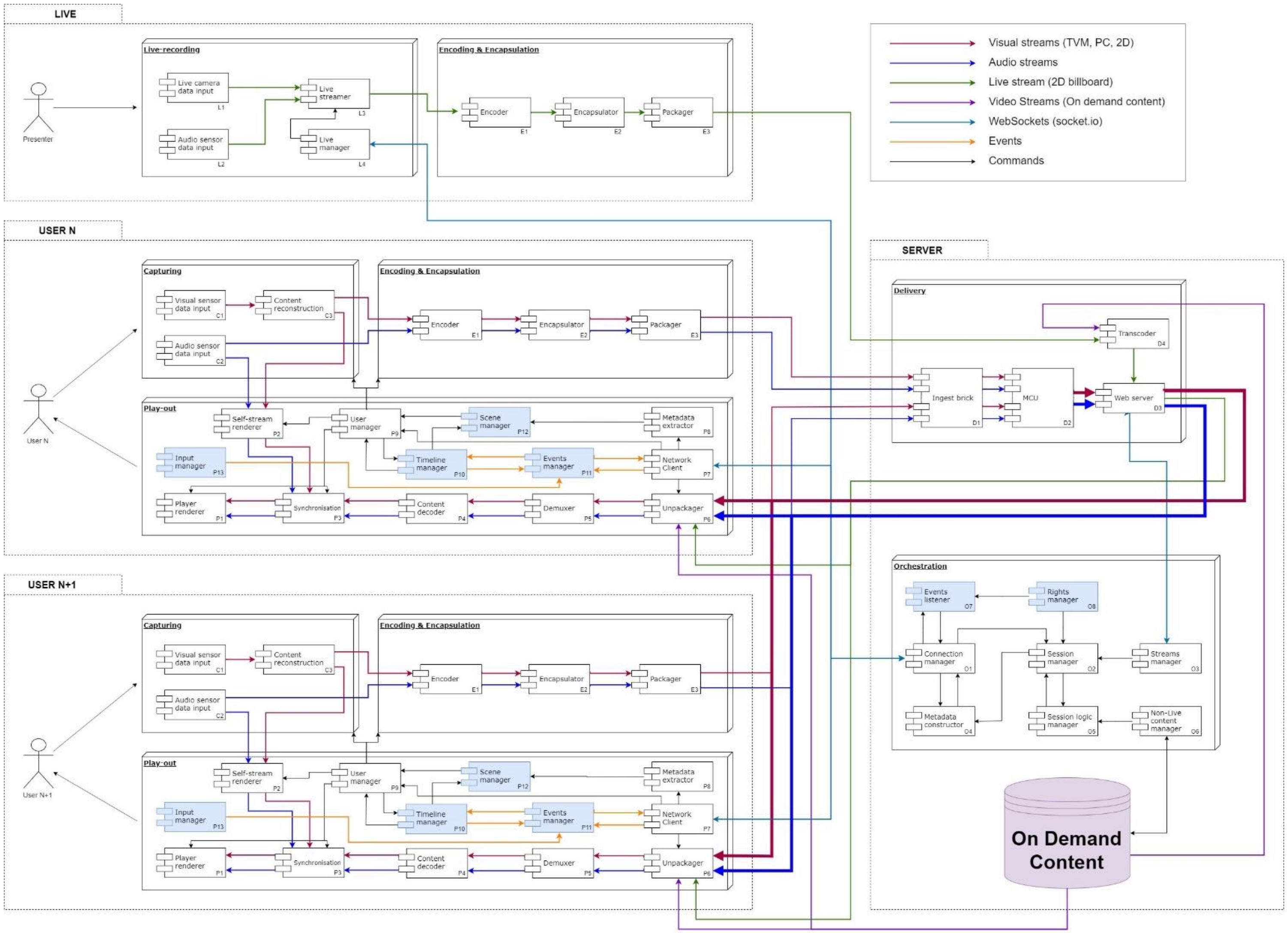


Figure 15: Pilot 3 component architecture diagram

5.1.1.3. Components definition

CAPTURING

The capturing component represents the first entry point of the VR-Together pipeline. It is in charge to create and provide the 3D representation of each end user. While the idea behind the capture technology has not changes from Pilot 1, in Pilot 2 and 3 the architecture is designed to accept more than 2 users, thus architecture definition depicted in Figure 15 is designed for an undetermined number of users in order to consider an extensible and scalable pipeline.

In the remaining part of this subsection, each sub-block of the capturing component is described.

- **C1 - Visual Sensor data input:** this module receives the data captured by a visual sensor (e.g., a RealSense camera or a Kinect) used in one participant's setup. The user is placed in a location where a hardware setup captures his/her motion and texture data. The current setup of VR-Together includes 4 capturing sensors (or just 1 in case of the lightweight web-based pipeline) that can be Microsoft Kinect or RealSense Cameras. In any case the input consists of 4 RGBD data streams together with the corresponding texture data. The streams are already time-stamped according to the internal clock of the sensor through which the data was captured.

Input: a user's motion data + texture
Output: raw RGB-D data + visual sensor timestamp

- **C2 - Audio sensor data input:** this module receives the audio sensor signal captured by the microphone used in one participant's setup. The participant's audio is captured in the configured bitrate, channel layout (the direction of the sound is inferred from the HMD direction) and time-stamped according to the audio sensor's internal clock.

Input: user's audio data
Output: user's audio frame + audio sensor relative timestamp

- **C3 - Content reconstruction**: this module receives the data captured from all visual sensors in a participant's setup and merges them into one single visual frame. The content captured from the visual sensors is processed and merged, performing tasks such as background removal, HMD removal or any other additional content reconstruction task that is needed, following the desired experience outcome. Furthermore, this module performs the synchronisation of the separate visual content streams. Each stream is synchronised according to its "creator sensor's" internal clock and all separate streams should be synchronised with each other when merging. The resulting visual data frame represents the RGBD and texture data of one single temporal instance. The resulting visual content stream follows a clock which is relative to the sensors' internal clocks. Note that this clock might drift from the Platform Clock.

Input: raw RGB-D data + sensor timestamp from all sensors in a participant's setup
Output: a visual frame (i.e. fused data, such as video, TVM or PC, created from data captured by all sensors of one user set-up) + visual frame hardware relative timestamp

LIVE-RECORDING

The live-recording component appeared in the Pilot 2 definition and remains part of the platform architecture for the Pilot 3. It allows the live representation of a presenter in the VR-Together experience (live news storyboard). This component takes the same structure as the capturing one except that it uses industrial audio-visual production techniques as the rendering of the live-presenter is based on an advanced panoramic billboard rendering.

- **L1 - Live Camera data input:** this module records the frames captured by a camera (e.g., a Cine stereo camera) installed in the live studio setup. The presenter is placed in front of a green background facing the camera. The incoming video stream is retrieved with a single time-stamp that depends on the clock of the recording device.

Input: presenter frame data
Output: RGB stereoscopic video raw stream data + record device timestamp

- **L2 - Audio sensor data input:** this module receives the audio sensor signal captured by a microphone used in the live studio setup. The presenter's audio is captured in the configured bitrate, channel layout (the direction of the sound is inferred from the HMD direction) and time-stamped according to the audio sensor's internal clock.

Input: presenter's audio data
Output: presenter's audio frame + audio sensor relative timestamp

- **L3 – Live streamer**: This module receives the raw stereoscopic video stream recorded from the panoramic camera in the live studio setup and operates a live stitching[5] operation on the visual frames. Then the module creates a unified RTMP stream constituted of stitched frames and audio frames by processing a synchronisation step based on the recording timestamp. It also provides a set of parameters to configure the output stream according the Live manager requests.

Input: raw panoramic stream + raw audio stream + record timestamp
Output: RTMP stream (including synchronised visual and audio data)

- **L4 - Live manager**: This module handles request exchanges with the connection manager of the Orchestration component. It can manage the live recording component states and provides relevant parameters to the Live streamer regarding stream specifications module according to the running Session specifications.

Input: JSON recording configuration
Output: void

ENCODING & ENCAPSULATION

This subsection describes how the media streams are prepared to be transmitted. In Pilot 1, two kind of content were involved: animated 3D Time Varying Meshes (TVM) and RGB-D video. In Pilot 2 the encoding platform was able to support additionally the Point Clouds (PC) data. For the Pilot 3, the plan is to make the encoding platform completely agnostic by supporting all media format involved in the VRT platform; 2D videos (RGB-D and live, VOD), TVM, PC and audio channel.

- **E1 - Encoder**: this module receives a visual or audio track related to one user (result of the capturing) and encodes it in order to reduce the bitrate needed to represent the visual/audio signal. The encoding configuration (including for example the target encoding bitrate, the frame rate, etc.) is dictated by the "Session manager" module ("Orchestration" component) that sets the platform configuration for the active session. The visual and audio streams are handled separately, each one by its corresponding encoder module. The result of this process is an encoded visual or audio stream.

[5] https://en.wikipedia.org/wiki/Image_stitching

Input: timestamped visual stream (i.e. Video, TVM or PC raw stream) / timestamped audio stream. Timestamps are set according to the Platform Clock.
Output: encoded visual stream (i.e. TVM or PC encoded stream) / encoded audio stream (Example: .ply file for an encoded PC and .aac file for encoded audio)

- **E2 - Encapsulator:** this module receives an encoded visual stream and an encoded audio stream, which are temporally synchronized (i.e., have timestamps that refer to the platform clock and are aligned), and multiplex them in a single stream. After being encoded the visual stream and audio streams are multiplexed and encapsulated to a media format (e.g., MP4, WebM, other), defined by the "Session manager" module, corresponding to the end-user's playout device, capabilities etc. The input in this process is the encoded and separated visual and audio stream.

Input: encoded visual stream (i.e. TVM or PC encoded stream) + encoded audio stream (Example: .ply file for an encoded PC and .aac file for encoded audio)
Output: audio-visual file (e.g. mp4, webM, etc.) including synchronized audio and visual tracks for one user

- **E3 - Packager:** this module receives single audio-visual (i.e., the encapsulated audio-visual content) or multiple audio-visual data corresponding to different users and packages it (them), so that the content can be transmitted by the "Delivery" component. The process can vary depending on the chosen content delivery configuration (e.g., DASH versus WebRTC).

Input: audio-visual file (s) (e.g. mp4, webM, etc.) (i.e., the encapsulated audio-visual content)
Output: packaged content (e.g. MPEG transport stream, SRT, MPEG-DASH, Microsoft Fragmented MP4 Ingest)

DELIVERY

The delivery component is the first block of the pipeline based on a server/cloud. It is in charge to receive process and transmit the various media stream received by each user. In this component, at the moment of the composition of this document, there are no remarkable changes between Pilot 1 and 2.

- **D1 - Ingest brick:** This module receives the audio-visual streams including the data of multiple users involved in the communication output of the "Encoding and encapsulating" component and adapt them to the format needed for storage in the "Web-server" module that transmits the data on the network.

Input: packaged audio-visual content
Output: audio-visual content ready for transmission (for example, DASH adaptation set)

- **D2 - Multi Control Unit (MCU):** Depending on the active session there might be 2 or more participants in one virtual environment. In the case where the participants are more than 2 the "Multi Control Unit" (MCU) module is activated. The MCU is responsible for combining the visual and audio inputs arriving from multiple sources; the MCU allows to reduce the network pressure on the capture system (since only one upload happens instead of number-of-users-minus-one when it is absent) at the cost of an intermediate server. The inputs are blended or re-organized into a common synchronised stream. The result of the MCU is then optionally encoded, encapsulated and packaged before reaching the "Delivery" component.

Input: multiple audio-visual content streams

Output: blended/mixed audio-visual stream including the data of multiple users involved in the communication (i.e., multiple TVMs or PCs)

- **D3 - Web server**: This module makes the content available for consumption and manages the endpoints at which the content is served.

Input: audio-visual content ready for transmission & signalling information (for example, DASH adaptation set, i.e. chunks and .mpd file)
Output: transport protocol messages & packets (for example, HTTP messages and .m4s DASH chunks)

- **D4 - Transcoder:** This module works directly with the Live node. Depending on the live presenter session status, the transcoder module can receive a live stream or an on-demand video stream (coming from a recorded video file), and process one of both to make it formatted as a DASH stream format. Thus, it allows to segment and timestamp the live video stream according to the platform clock, which will be the reference timestamp to consider on the playout side.

Input: Live RTMP stream / On-Demand RTMP stream
Output: DASH live stream

ORCHESTRATOR

The central component of the VR-Together platform is the Orchestrator component, which provides all clients with the information necessary to initiate a communication session of end users over the VR-Together platform. This includes the management of sessions, instantiation of scenarios, discovery of available rooms, gathering of pointers to content sources and other clients in a session as well as the capture sources.

The Orchestrator is responsible for signalling synchronization data between the different streams consumed by the clients and the MCU. Besides the synchronisation data, other session control data is signalled via the Orchestrator, like content changes, pause/play, scenarios updates, etc.

The Pilot 2 Orchestration reflexions have introduced five conceptual models:

- User: A user who wants to share an immersive social experience with other persons. Also as depicted in Section 5.1, user entity is composed of the three components that orchestrator needs to interact with: capturing, encoding and playout.
- Admin: An administrator can have a global control of the platform and force session control data through a corresponding interface, e.g. in order to facilitate user experiments and demos
- Scenario: This describes the virtual world composed with at least one room. The scenario includes the description of the whole 3D scene and the underlying logic (timeline, interactive event, etc.).
- Session: A session gathers users that want to share an immersive social experience together based on a scenario.
- Room: A room is a virtual space where users are located together which is part of the scenario.

More precisely, we can distinguish a scenario from these two ways:

- A scenario model. It defines a static scenario description; e.g. can't be changed by the VR-Together experience: Room descriptions, Room capabilities and constraints, 3D scene description, and pointers (URIs) to the content sources. This permanent dataset is stored on a dedicated database.

- A scenario instance. It is instantiated from a scenario model when a session is created and will carry on the logics and internal states of the scenario. It temporally stores and maintains dynamic and stateful information related to a VR-Together session. The clients in a session need to have a shared state (or view) of the virtual world. This includes: the current time in the world and assets (e.g. videos) in the world, the state of the world, as well as URI's to end-user content streams.

These entities are dependant within each other; we can consider these dependencies with the following UML diagram.

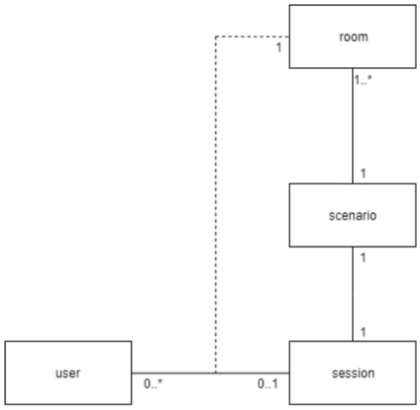


Figure 16: UML representation of the orchestrator entities dependencies

For the Pilot 3 development, the major novelty introduced is the ability to manage and relay the interactions that occurs between each user according to the scenario timeline. In order to provide a correct consistency across the multi-user experience regarding the interactions that will occur with the virtual environment, the retained strategy to implement consists in following a reference user that will execute the reference scenario timeline system as an administrator of the session.

It is important to note that the Orchestrator component controls but does not process media streams. For all control data, all clients (regardless of the type) have a common interface to the orchestrator. However, clients might have different interfaces to content (based on the content type and content server).

In this way a client may retrieve one or multiple media content streams from one or multiple Content Servers. The URLs to the streaming content are provided in the session's metadata as JSON data being sent as parameters through web-requests exchanges. In addition to media content, each client receives streams from other clients that are aggregated by the MCU for audio/visual communication and transmits streams for other users as well.

Thanks to the user-manager depicted below. Play-out component description, each client is responsible for its own capture and encoding component integrity both by matter of software integration design and hardware dependency.

As far as the modules functionality that is included in the Orchestrator component, we can see the:

- **O1 - Connection Manager**: It is the entry point of the orchestration component. It ensures connections within each node entity of the platform. The users' connections to the VR-Together platform are managed and maintained by this module which then transfers the relevant information to the "metadata constructor" and the "session manager".

Input: user connection information
Output: user connection pointers

- **O2 - Session manager**: This module handles and updates all running sessions in parallel according to users' requests. It operates all sessions commands (session creation, session connection, session update, etc.) and is aware of all the information regarding a session (e.g., how many users are connected to the session, on which room they are located, rooms attendance, which is the non-live content used, etc.). It finally brings corresponding configuration from the "session logic manager" and "stream manager".

  Input: Session information
  Output: Session information pointers

- **O3 - Stream manager:** This module is responsible for acquiring the stream information from the Delivery component and transferring it to the "metadata constructor" (through the Session Manager O2) in order to facilitate the stream selection process for the "Play-out" component.

Input: Stream information
Output: Stream information pointers

- **O4 - Metadata constructor**: This module of the Orchestrator constructs the metadata content description file, corresponding to a JSON content, which is used by the "Play-out" component of the VR-Together platform. The responsibility of the constructor is to build the metadata that is necessary for the player in order to project the video stream of the users, pointing to each one's video stream endpoint (originating from the Delivery component), follow the rule set as this is described in the game logic function while using the configuration provided by the "Session manager".

Input: Virtual experience configuration
Output: JSON description files

- **O5 - Session logic manager**: Within the virtual environment a set of rules are defined that form the desired "session logic" to be applied (height of video stream representation, starting position in the room, etc.).

Input: Platform configuration
Output: Configuration pointers

- **O6 - Non-live content manager**: The non-live content, also considered as on-demand-content, that is included in the VR-Together platform (room graphics, stereoscopic videos for billboards, 360 videos, etc.) is managed (host + delivery) by this module. This module also provides the necessary information (URLs) to the metadata constructor related to the non-live content accessibility.

Input: Session configuration
Output: Resource pointers + metadata configuration

- **O7 - Rights manager**: The mechanism of synchronising the interactions between connected users of a session and the virtual environment relies on this module. It is in charge of determining a "master" user that will fire the scenario events according to the current state of the timeline. Next, the other users are granted with a "slave" role, they are following the master user; they can only catch its events.

Input: Users pointers
Output: Users roles configuration

- **O8 - Events listener**: This module is in charge of transmitting the user events related to the interactions with the virtual environment. It is working with the "Rights manager" in order to receive the "super user" events and to relay them to the other connected users.

Input: Super User events
Output: Scenario events

PLAYOUT

The playout component is in charge of delivering to the end user the 3D representation of the virtual world where the experience designed for the platform is set. In Pilot 1 this component had to deal with the virtual scene including only one user because the experience was plan for 2 connected users. In Pilot 2 the experience can be performed for 4 users, which introduced an additional sub component; the User Manager. For Pilot 3 the experience involves up to 10 connected users to be part of the actors of the experience by letting them interact with the virtual environment and the scenario. For such feature, three additional modules appear; the timeline manager, the events manager and the scene manager depicted hereafter.

- **P1 - Player renderer:** The player is responsible for rendering the content following the metadata pointers in order and project the desired media content.

Input: Metadata descriptor value
Output: Content play-out

- **P2 - Self-stream renderer:** The user's self-created stream (generated from his "own" instance of the capturing component) is "consumed" in this function and passed on to the player for rendering.

Input: void
Output: void

- **P3 - Synchronization**: This module refers to the time-stamp alignment that needs to happen in the received content (decoded stream and self-stream) in order to correctly render the visual and audio streams according to the universal clock's timestamp.

Input: void
Output: void

- **P4 - Content Decoder:** The content arriving from the Delivery component is decoded and transferred to the player for projection. The decoders are different for visual and audio content but are just mentioned here as the same logical module.

Input: encoded and unpacked audio OR visual content
Output: decoded audio or visual content

- **P5 - Demuxer:** This module splits the audio and video of the content received from the web server.

Input: Unpackaged audio-visual content
Output: Unpackaged encoded audio content + unpackaged encoded visual content

- **P6 - Unpackager**: this module un-packages the audio-visual content that is received from the web server through the network client.

Input: void
Output: void

- **P7 - Network client**: This module is responsible for establishing the connection with the web server that provides the content to the "Play-out" component.

Input: void
Output: void

- **P8 - Metadata extractor:** This module parses the metadata file provided by the "Orchestrator" and extracts all the necessary information in order to facilitate rendering of the content.

Input: Metadata descriptor file
Output: Content and configuration values

- **P9 - User manager**: This module has the role of managing the whole user component on the client side. It is able to establish local exchanges between Capturing, Encoding and Play-out components. It is also in charge of managing the user states on the Platform by catching and managing orchestrator messages and requests to the Player.

Input: Orchestrator JSON messages
Output: void

- **P10 - Timeline manager**: The Timeline manager can have two separated roles; in first if the current user has been granted to be the authoritative user by the rights manager module (O8), the current state of advancement of the experience is followed by this module. From a predefined track system edited by the level designer of the 3D experience, the timeline manager is in charge of following the playing progress of the scenario to establish an experience timestamp that serves as a reference for each connected user sharing common interaction events. Otherwise it is in charge of realigning the scenario track progress according to the master user timestamp reference.

Input: Scenario track progress / User events pointers
Output: Timestamped pointers

- **P11 - Events manager**: In the same logic the Events manager can have two separated roles; either it is in charge of generating the events from the Timeline manager timestamps if the current user is a master user, either it is in charge of receiving the events coming from the master user and catches them to the Timeline manager and Scene manager modules.

Input: Timestamped pointers / User events JSON messages
Output: User events JSON messages / User events pointers

- **P12 - Scene manager**: This module structures the whole 3D environment apart from the user logics. According to the User manager states as well as the events provided by the Timeline manager and Events manager, the Scene manager is able to create instances of dynamic objects that need to appear in the virtual world when an interaction needs it.

Input: Metadata pointers
Output: 3D objects instances

- **P13 - Input manager**: All user inputs are tracked and recorded by this module. It provides the necessary interface to customize and select the type of input captured (i.e. those related to hand position and orientation, gestures triggers, etc.) as well as the required parameters to configure capture frequency according to the input type and interactivity purpose. Then it serializes the collected data in formatted metadata to be injected as events (high-frequency) in the Events manager (P11) that will take care of posting them to the orchestrator.

Input: User input data
Output: User input devices metadata

5.2. Hardware architecture

This section provides the hardware architecture of the VR-Together platform, as designed for Pilot 1 and 2 and as it has been updated for Pilot 3.

Pilot 1:

In the diagram shown in Figure 17, we can see the hardware components that form the VR-Together platform when Pilot 1 was released. The HW considered was: two capture rigs; combining 4 RGBD cameras (and a single one in case of the web pipeline) each and a server for capture integration and encoding. Two dedicated servers will take care of content delivery and content orchestration, respectively. Finally, two playout devices will allow end-user content consumption.

Pilot 2:

Figure 18 gives an updated version of the initial diagram compliant with the Pilot 2 scope. Major change relates to the 5 connected user nodes that constitute the whole platform. Each user node is shown as a user entity depicted according to 3 user node types; web-client node, TVM node, PC node. Additionally, a new node appears to grant the live news/presenter feature of the platform with the Live-recording node. It consists in a dedicated capture setup based on a green background recording of a presenter in live which is broadcaster thanks to a local streaming unit sending the raw stream to the network delivery infrastructure.

Pilot 3:

Finally, the last hardware architecture iteration is visible in Figure 19, intended for pilot 3. The model is essentially preserved in term of structuration with similar nodes (web, TVM, PC) definition but is distinguished with end-user representations that integrate virtual reality controllers as the key enablers of interactivity features. Moreover, the Live node appears greyed-out as the feature is not required for Pilot 3, but is preserved on the platform definition in order to grant platform retro compatibility with previous pilot version.

Platform architecture:

Across the 3 pilot definitions, it results that for each user node, the hardware setup is constituted of a capture rig (one or four RGBD cameras depending on the node type), an encoding component that processes captured stream data and send it to the delivery server, and the play-out setup that retrieves incoming stream data as well as on-demand content, provided by the content database, to finally project all of it to the user. Then the centralized orchestrator server operates web-requests within all the user nodes and delivery server to manage the whole platform.

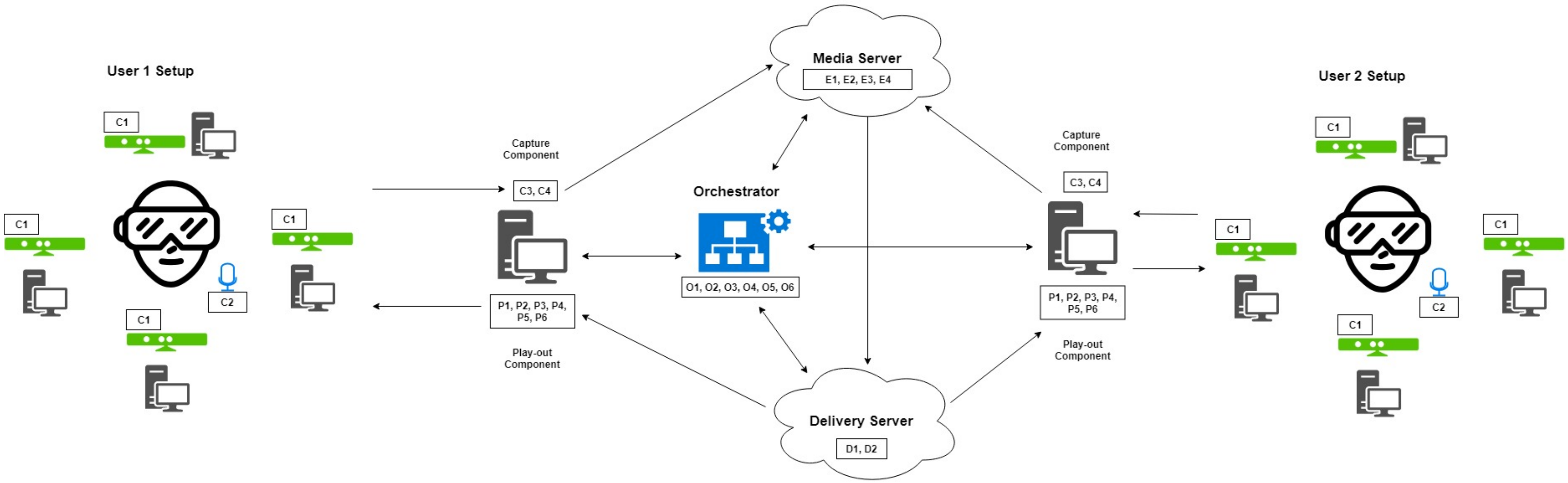


Figure 17: Hardware Architecture (Pilot 1)

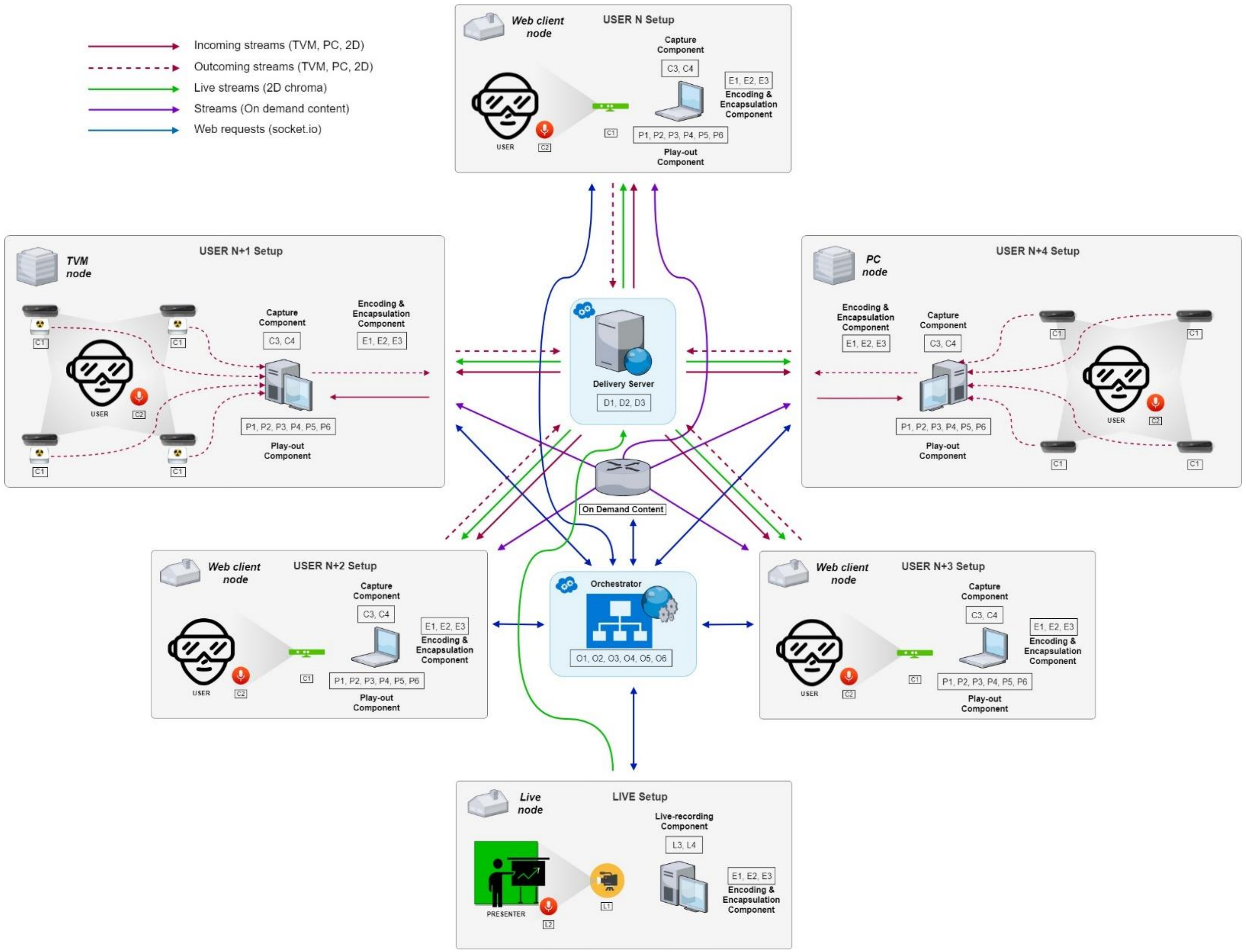


Figure 18: Hardware architecture (Pilot 2)

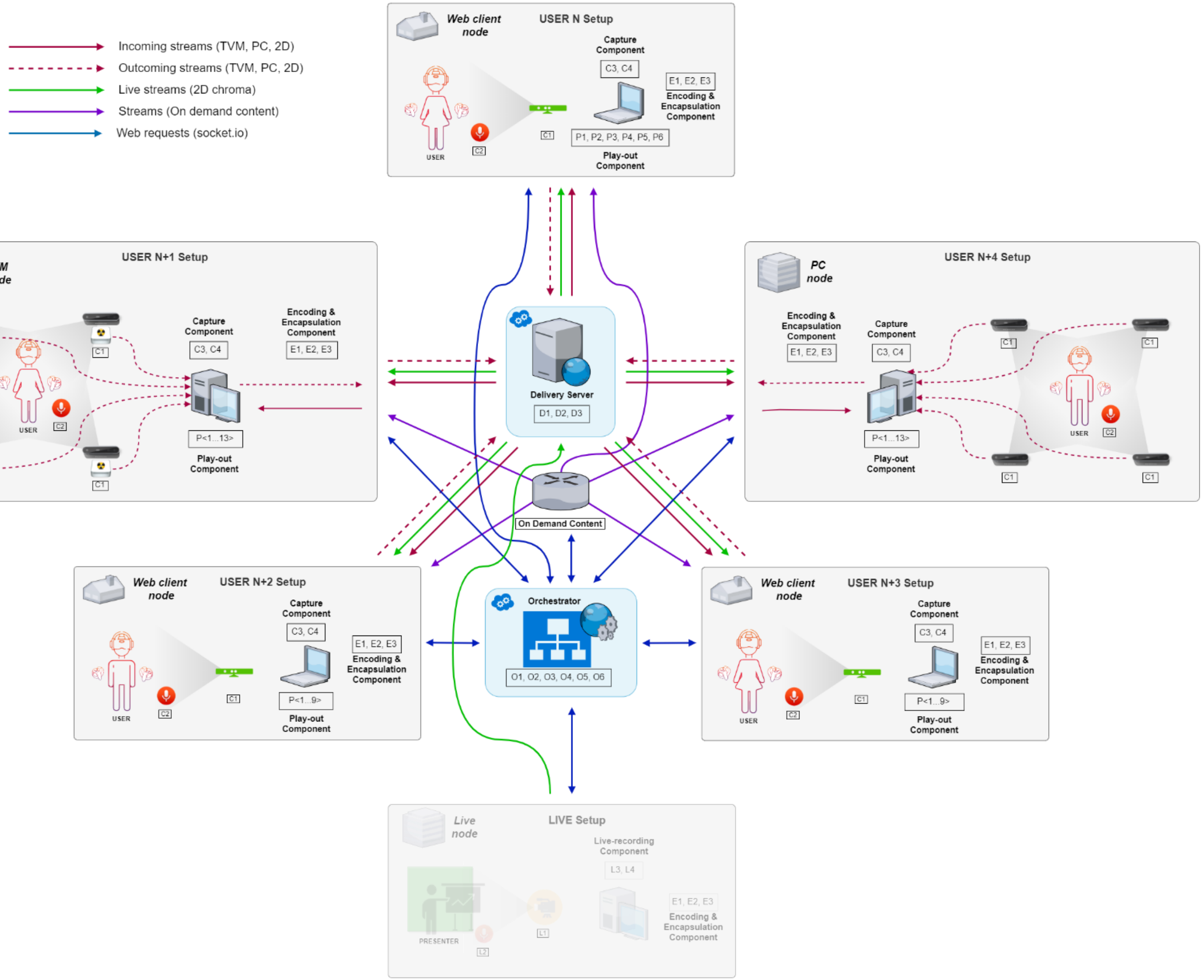


Figure 19: Hardware architecture (Pilot 3)

5.2.1. End-user Set-up

Each end user needs to be using a capturing setup, an encoding setup and a playout setup in order to be able to use the VR-Together software platform.

First, the capturing setup is composed of:

- Visual sensor capturing: the set-up includes either 1 visual-sensor (for a web-client node) or 4 visual sensors (for a native-client node) responsible for capturing the user's representation that will be encoded and projected in the virtual rooms of the VR-Together platform. Although for the Pilot 1 the retained sensors where the Microsoft Kinect for Xbox One, due to the current discontinuation of the product, the one added for Pilot 2 is the Intel RealSense D415. Nevertheless, the Kinect support remains maintained, particularly for the TVM and web-client pipeline to ensure compatibility with coming Kinect for Azure Microsoft project.
- Audio sensor capturing: for the capturing of the audio of the user the sensor to be used is the microphone that is embedded on the head mounted display that is used by the end user.
- Local visual sensor processing: For the TVM pipeline, each one of the visual sensors needs to be connected to a separate terminal that processes the raw input stream. In the current set-up of VR-Together the terminals used for this purpose are Intel NUC D54250.
- Local audio-visual stream processing: in a capturing set-up environment a central node/terminal is responsible for collecting the separate visual streams together with the audio stream and processing them accordingly. The hardware device used for this purpose is not specific but the most important minimum hardware requirements that it needs to comply with are:
    - 16GB of Ram
    - I7 or equivalent Central Processing Unit
    - Separate graphics processor with 8GB of memory

Then the encoding setup is the same one as used for the capturing, with the outcoming media stream of the capturing directly injected into the encoding component on the same machine.

At last, the playout setup is composed of:

- Rendering processing: the playout hardware machine is in charge of decoding the incoming streams and rendering the whole scene that aggregates every media (volumetric streams, 3D scene assets, videos …). For all the pipelines the rendering station is the same one as used for capturing except for the TVM pipeline which requires an additional station.
- Head mounted display: the playout devices to be used by the end users are commercially available products. In our set-up we are using the Oculus Rift Consumer Version 1. In Year 3, the use of non-tethered HMDs (Oculus Quest) will be tested for the lightweight pipeline.

5.2.2. Cloud server infrastructure

All the software components and modules that are not part of the capturing set-up will be hosted in cloud or dedicated servers that will be responsible for performing the required functionalities.

As mentioned in the Software Architecture description, the VR-Together platform will support 3 configurations for capture of the end users that will be later projected in the in the virtual environment of an active session. The table below describes the requirements of the cloud server nodes supporting the operation of each configuration.

Configuration	Type of server	HW requirements	SW requirements	Storage capacity	Operating System
PC	**Streaming Server / Orchestration Server**	VM instance: CPU: i7 4 cores @3.80 GHz RAM 16 GB	NodeJS Evanescent	128 GB	Windows / Linux
TVM	**Streaming Server**	VM instance: CPU: i7 4 cores @3.80 GHz RAM 16 GB	RabbitMQ Server 3.6.15	30 GB	Windows / Linux
TVM	**Storing Server**	VM instance: CPU: i5 4 cores @ 2.8+ GHz RAM 8 GB	-	2 TB	Windows
2D	**Other Server**	VM instance: CPU: i7 4 cores @ 3.80 GHz RAM 16 GB	Linux package manager, OS support for Docker	128 GB	Linux
ALL	**Live Streaming Server**	VM instance: CPU: 8 physical cores RAM 8 GB	Linux package manager, OS support for Docker	256 GB SSD	Linux

Table 6: Cloud Server Requirements

The values mentioned above are subject to changes depending on the requirements gathered during the VR-Together platform evaluations and experiments. That is because various aspects of the experience, such as latency or projection quality, might require higher processing capabilities and therefore render the above information out-dated. The currently shown values have been tested against the components as they are delivered in D3.6.

5.3. Architecture update for final platform delivery

From the last evolutions adopted by the Requirements Update made during the delivery of the final platform release, especially in a new context where social distancing is being crucial, the VRT platform architecture has been adapted to develop new modalities;

- Extend the scalability of the Platform (to connect a wider range of users)
- Hybridization of the capture modalities to combine simple and complex user representation topologies (to target more users)
- Enable legacy 2D webcam base capture
- Enable 2D Flat screen experience (in addition to VR immersion)
- Enable lighter experience to run on lower end devices

5.3.1. PoC Software Architecture impacts

5.3.1.1. Components Architecture update

To extend the capture modalities to allow to connect a wider range of connected users, the following components are impacted and slightly modified to take into account the new platform requirements;

CAPTURING

The capturing component represents the first entry point of the VR-Together pipeline. It is in charge to create and provide the representation (in 2D or 3D) of each end user. The final architecture definition depicted in Figure 15 is designed for an undetermined number of users in order to consider an extensible and scalable pipeline.

In the remaining part of this subsection, each sub-block of the capturing component is described.

- **C1 - Visual Sensor data input:** this module receives the data captured by a visual sensor (e.g., a Webcam, a depth camera like RealSense or Azure) used in one participant's setup. The user is placed in a location where a hardware setup captures his/her motion and texture data. The current setup of VR-Together can include various camera rigs depending of the user setup complexity;
  - One single Webcam camera which is recording in real time the User video stream.
  - One single depth-sensor like Intel RealSense or Microsoft Azure Kinect placed in front of the User and recording in real-time the color and depth video streams.
  - Three or four depth-sensors, that can be either Microsoft Azure Kinect cameras or Intel RealSense cameras which are connected together to be synchronised locally. They are placed around the User and are recording in real-time the color and depth unified videos streams.

  In any case the input consists of RGB[6] + optionally a D[7] data streams with the corresponding video texture data. The streams are already time-stamped according to the internal clock of the sensor through which the data was captured.

Input: a user's motion data + texture
Output: raw RGB or RGB-D data + visual sensor timestamp

[6] Red, Green, Blue
[7] Depth

- **C2 - Audio sensor data input:** this module receives the audio sensor signal captured by the microphone used in one participant's setup. The participant's audio is captured in the configured bitrate, channel layout (the direction of the sound is inferred from the HMD direction) and time-stamped according to the audio sensor's internal clock.

Input: user's audio data
Output: user's audio frame + audio sensor relative timestamp

- **C3 - Content reconstruction**: this module receives the data captured from the visual sensors in a participant's setup, and if multiple sensors are connected together, it merges them into one single visual frame. The content captured from the visual sensors is processed and merged, performing tasks such as background removal, HMD removal or any other additional content reconstruction task that is needed, following the desired experience outcome. Furthermore, this module performs the synchronisation of the separate visual content streams. Each stream is synchronised according to its "creator sensor's" internal clock and all separate streams should be synchronised with each other when merging. The resulting visual data frame represents the RGBD and texture data of one single temporal instance. The resulting visual content stream follows a clock which is relative to the sensors' internal clocks. Note that this clock might drift from the Platform Clock.

Input: raw RGB or RGB-D data + sensor timestamp from all sensors in a participant's setup
Output: a visual frame (i.e. fused data, such as video, TVM or PC, created from data captured by all sensors of one user set-up) + visual frame hardware relative timestamp

ENCODING & ENCAPSULATION

This subsection describes how the media streams are prepared to be transmitted. The encoding platform is completely agnostic supporting all media format involved in the VRT platform; 2D videos (RGB, RGB-D and live, VOD), TVM, PC and audio channel.

- **E1 - Encoder**: this module receives a visual or audio track related to one user (result of the capturing) and encodes it in order to reduce the bitrate needed to represent the visual/audio signal. The encoding configuration (including for example the target encoding bitrate, the frame rate, etc.) is dictated by the "Session manager" module ("Orchestration" component) that sets the platform configuration for the active session. The visual and audio streams are handled separately, each one by its corresponding encoder module. The result of this process is an encoded visual or audio stream.

Input: timestamped visual stream (i.e. Video, TVM or PC raw stream) / timestamped audio stream. Timestamps are set according to the Platform Clock.
Output: encoded visual stream (i.e. TVM or PC encoded stream) / encoded audio stream (Example: .ply file for an encoded PC and .aac file for encoded audio)

- **E2 - Encapsulator:** this module receives an encoded visual stream and an encoded audio stream, which are temporally synchronized (i.e., have timestamps that refer to the platform clock and are aligned), and multiplex them in a single stream. After being encoded the visual stream and audio streams are multiplexed and encapsulated to a media format (e.g., MP4, WebM, other), defined by the "Session manager" module, corresponding to the end-user's playout device, capabilities etc. The input in this process is the encoded and separated visual and audio stream.

Input: encoded visual stream (i.e. TVM or PC encoded stream) + encoded audio stream (Example: .pcc file for an encoded PC and .aac file for encoded audio)
Output: audio-visual file (e.g. mp4, webM, etc.) including synchronized audio and visual tracks for one user

- **E3 - Packager:** this module receives single audio-visual (i.e., the encapsulated audio-visual content) or multiple audio-visual data corresponding to different users and packages it (them), so that the content can be transmitted by the "Delivery" component. The process can vary depending on the chosen content delivery configuration (e.g., DASH versus WebRTC).

Input: audio-visual file (s) (e.g. mp4, webM, etc.) (i.e., the encapsulated audio-visual content)
Output: packaged content (e.g. MPEG transport stream, SRT, MPEG-DASH, Microsoft Fragmented MP4 Ingest)

ORCHESTRATOR

The central component of the VR-Together platform is the Orchestrator component, which provides all clients with the information necessary to initiate a communication session of end users over the VR-Together platform. This includes the management of sessions, instantiation of scenarios, discovery of available rooms, gathering of pointers to content sources and other clients in a session as well as the capture sources depending on the user configuration.

The Orchestrator is responsible for signalling synchronization data between the different streams consumed by the clients and the MCU. Besides the synchronisation data, other session control data is signalled via the Orchestrator, like content changes, pause/play, scenarios events, etc.

The Orchestration reflexions have introduced five conceptual models:

- **User:** A user who wants to share an immersive social experience with other persons. Also as depicted in Section 5.1, the user entity is composed of the three components that orchestrator needs to interact with: capturing, encoding and playout.
- **Admin:** An administrator can have a global control of the platform and force session control data through a corresponding interface, e.g. in order to facilitate user experiments and demos. It is also assimilated as the session creator.
- **Scenario:** This describes the virtual world composed with at least one room. The scenario includes the description of the whole 3D scene and the underlying logic (timeline, interactive event, etc.). Note: Specific scenarios descriptions (e.g lighter 3D environment) allows to address users equipped with low-end machines with lower 3D rendering capabilities.
- **Session:** A session gathers users that want to share an immersive social experience together based on a scenario. It allows to integrate all the logics of the Orchestrator on the client side (join, leave, refresh, infos, etc.)
- **Room:** A room is a virtual space where users are located together which is part of the scenario.

More precisely, we can distinguish a scenario from these two ways:

- A scenario model. It defines a static scenario description; e.g. can't be changed by the VR-Together experience: Room descriptions, Room capabilities and constraints, 3D scene description, and pointers (URIs) to the content sources. This permanent dataset is stored on a dedicated database.
- A scenario instance. It is instantiated from a scenario model when a session is created and will carry on the logics and internal states of the scenario. It temporally stores and maintains dynamic and stateful information related to a VR-Together session. The clients in a session need to have a shared state (or view) of the virtual world. This includes: the current time in the world and assets (e.g. videos) in the world, the state of the world, as well as URI's to end-user content streams.

These entities are dependant within each other; we can consider these dependencies with the following UML diagram.

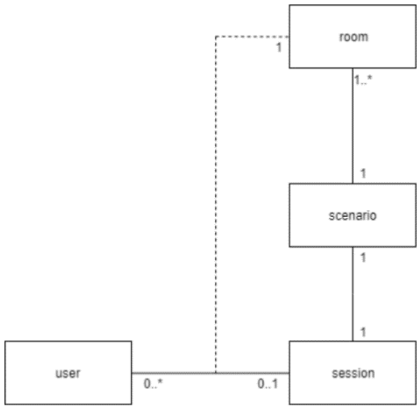


Figure 20: UML representation of the orchestrator entities dependencies

It is important to note that the Orchestrator component controls but does not process media streams. For all control data, all clients (regardless of the type) have a common interface to the orchestrator. However, clients might have different interfaces to content (based on the content type and content server).

In this way a client may retrieve one or multiple media content streams from one or multiple Content Servers. The URLs to the streaming content are provided in the session's metadata as JSON data being sent as parameters through web-requests exchanges. In addition to media content, each client receives streams from other clients that are aggregated by the MCU for audio/visual communication and transmits streams for other users as well.

Thanks to the user-manager depicted below in the Play-out component description, each client is responsible for its own capture and encoding component integrity both by matter of software integration design and hardware dependency.

As far as the modules functionality that is included in the Orchestrator component, we can see the:

- **O1 - Connection Manager**: It is the entry point of the orchestration component. It ensures connections within each node entity of the platform. The users' connections to the VR-Together platform are managed and maintained by this module which then transfers the relevant information to the "metadata constructor" and the "session manager".

Input: user connection information
Output: user connection pointers

- **O2 - Session manager**: This module handles and updates all running sessions in parallel according to users' requests. It operates all sessions commands (session creation, session connection, session update, etc.) and is aware of all the information regarding a session (e.g., how many users are connected to the session, on which room they are located, rooms attendance, which is the non-live content used, etc.). It finally brings corresponding configuration from the "session logic manager" and "stream manager".

  Input: Session information
  Output: Session information pointers

- **O3 - Stream manager:** This module is responsible for acquiring the stream information from the Delivery component and transferring it to the "metadata constructor" (through the Session Manager O2) in order to facilitate the stream selection process for the "Play-out" component.

Input: Stream information
Output: Stream information pointers

- **O4 - Metadata constructor**: This module of the Orchestrator constructs the metadata content description file, corresponding to a JSON content, which is used by the "Play-out" component of the VR-Together platform. The responsibility of the constructor is to build the metadata that is necessary for the player in order to project the video stream of the users, pointing to each one's video stream endpoint (originating from the Delivery component), follow the rule set as this is described in the game logic function while using the configuration provided by the "Session manager".

Input: Virtual experience configuration
Output: JSON description files

- **O5 - Session logic manager**: Within the virtual environment a set of rules are defined that form the desired "session logic" to be applied (height of video stream representation, starting position in the room, etc.).

Input: Platform configuration
Output: Configuration pointers

- **O6 - Non-live content manager**: The non-live content, also considered as on-demand-content, that is included in the VR-Together platform (room graphics, stereoscopic videos for billboards, 360 videos, etc.) is managed (host + delivery) by this module. This module also provides the necessary information (URLs) to the metadata constructor related to the non-live content accessibility.

Input: Session configuration
Output: Resource pointers + metadata configuration

- **O7 - Rights manager**: The mechanism of synchronising the interactions between connected users of a session and the virtual environment relies on this module. It is in charge of determining a "master" user that will fire the scenario events according to the current state of the timeline. Next, the other users are granted with a "slave" role, they are following the master user; they can only catch its events.

Input: Users pointers
Output: Users roles configuration

- **O8 - Events listener**: This module is in charge of transmitting the user events related to the interactions with the virtual environment. It is working with the "Rights manager" in order to receive the "super user" events and to relay them to the other connected users.

Input: Super User events
Output: Scenario events

PLAYOUT

The playout component is in charge of delivering to the end user the 3D representation of the virtual world where the experience designed for the platform is set. The VRT experience involves up to 10 connected users using a capturing system (volumetric or 2D) to be part of the actors of the experience by letting them interact with the virtual environment and the scenario. Then a larger amount of connected users is made available thanks to additional user representations (e.g Spectators, Ghosts, 3D Avatars) that don't rely on the capturing system and are part of the Player implementation. These additional connection mode are given thanks to the Orchestrator integration (e.g User manager module).

- **P1 - Player renderer:** The player is responsible for rendering the content following the metadata pointers in order and project the desired media content.

Input: Metadata descriptor value
Output: Content play-out

- **P2 - Self-stream renderer:** The user's self-created stream (generated from his "own" instance of the capturing component) is "consumed" in this function and passed on to the player for rendering.

Input: void
Output: void

- **P3 - Synchronization**: This module refers to the time-stamp alignment that needs to happen in the received content (decoded stream and self-stream) in order to correctly render the visual and audio streams according to the universal clock's timestamp.

Input: void
Output: void

- **P4 - Content Decoder:** The content arriving from the Delivery component is decoded and transferred to the player for projection. The decoders are different for visual and audio content but are just mentioned here as the same logical module.

Input: encoded and unpacked audio OR visual content
Output: decoded audio or visual content

- **P5 - Demuxer:** This module splits the audio and video of the content received from the web server.

Input: Unpackaged audio-visual content
Output: Unpackaged encoded audio content + unpackaged encoded visual content

- **P6 - Unpackager**: this module un-packages the audio-visual content that is received from the web server through the network client.

Input: void
Output: void

- **P7 - Network client**: This module is responsible for establishing the connection with the web server that provides the content to the "Play-out" component.

Input: void
Output: void

- **P8 - Metadata extractor:** This module parses the metadata file provided by the "Orchestrator" and extracts all the necessary information in order to facilitate rendering of the content.

Input: Metadata descriptor file
Output: Content and configuration values

- **P9 - User manager**: This module has the role of managing the whole user component on the client side. It is able to establish local exchanges between Capturing, Encoding and Play-out components. It is also in charge of managing the user states on the Platform by catching and managing orchestrator messages and requests to the Player.

Input: Orchestrator JSON messages
Output: void

- **P10 - Timeline manager**: The Timeline manager can have two separated roles; in first if the current user has been granted to be the authoritative user by the rights manager module (O8), the current state of advancement of the experience is followed by this module. From a predefined track system edited by the level designer of the 3D experience, the timeline manager is in charge of following the playing progress of the scenario to establish an experience timestamp that serves as a reference for each connected user sharing common interaction events. Otherwise it is in charge of realigning the scenario track progress according to the master user timestamp reference.

Input: Scenario track progress / User events pointers
Output: Timestamped pointers

- **P11 - Events manager**: In the same logic the Events manager can have two separated roles; either it is in charge of generating the events from the Timeline manager timestamps if the current user is a master user, either it is in charge of receiving the events coming from the master user and catches them to the Timeline manager and Scene manager modules.

Input: Timestamped pointers / User events JSON messages
Output: User events JSON messages / User events pointers

- **P12 - Scene manager**: This module structures the whole 3D environment apart from the user logics. According to the User manager states as well as the events provided by the Timeline manager and Events manager, the Scene manager is able to create instances of dynamic objects that need to appear in the virtual world when an interaction needs it.

Input: Metadata pointers
Output: 3D objects instances

- **P13 - Input manager**: All user inputs are tracked and recorded by this module. It provides the necessary interface to customize and select the type of input captured (i.e. those related to hand position and orientation, gestures triggers, etc.) as well as the required parameters to configure capture frequency according to the input type and interactivity purpose. Then it serializes the collected data in formatted metadata to be injected as events (high-frequency) in the Events manager (P11) that will take care of posting them to the orchestrator.

Input: User input data
Output: User input devices metadata

5.3.1.2. Final Platform Architecture

The Pilot 3 platform definition is basically based on the Pilot 2 version where components and modules have been reorganised to integrate new modules allocated to the interactivity system presented in the previous section. These new modules are allocated to the interaction functionalities between users and virtual environment.

We can foresee 5 new modules which depend on the Playout and Orchestration components:

- O7 - Events listener
- O8 - Rights manager
- P10 - Timeline manager
- P11 - Scene manager
- P12 - Events manager
- P13 - Input manager

These modules are portrayed in the component architecture diagram in Figure 15 and take place into the whole component's definition in the following section 5.1.1.3.

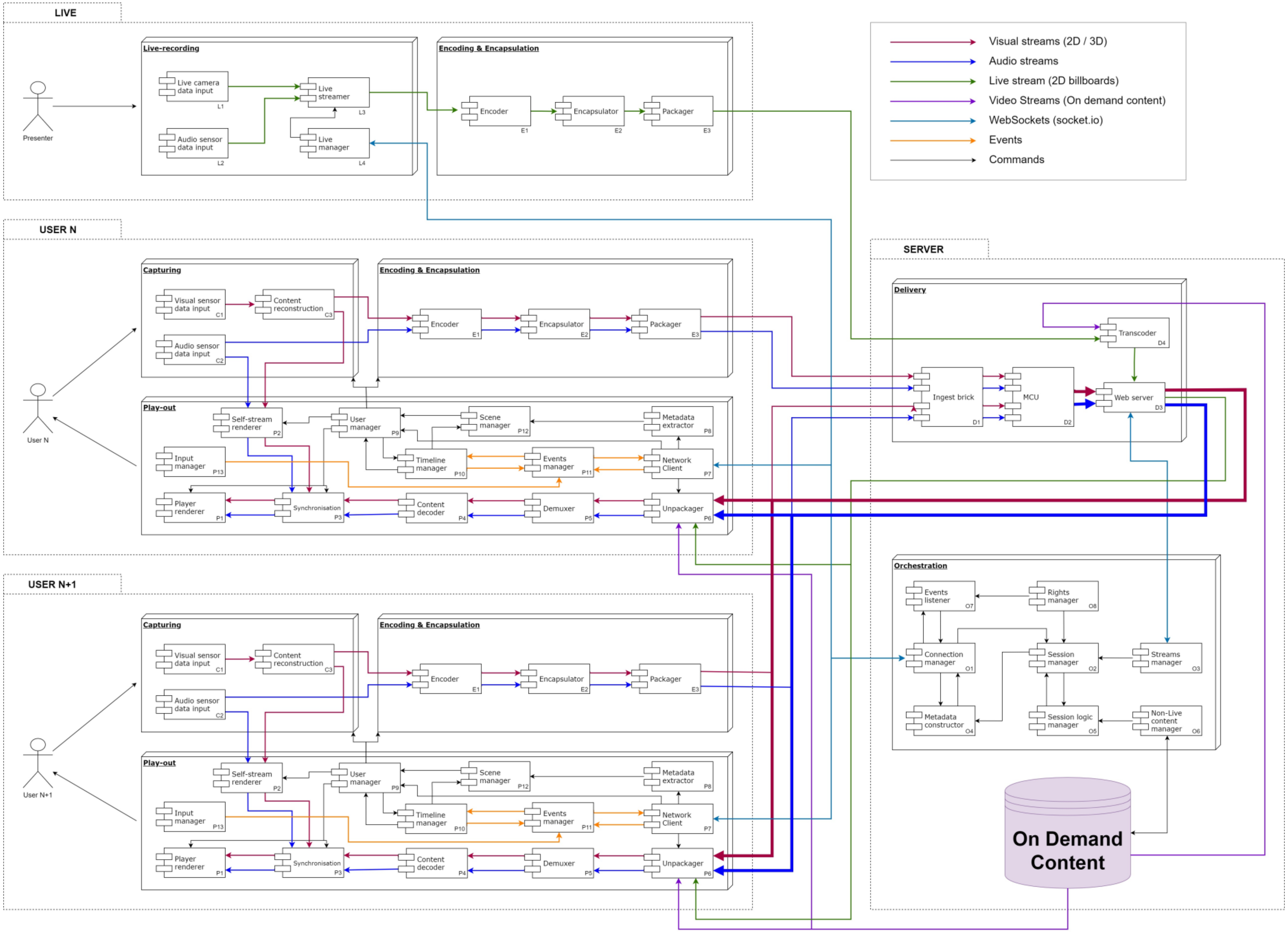


Figure 21: Pilot 3 component architecture diagram

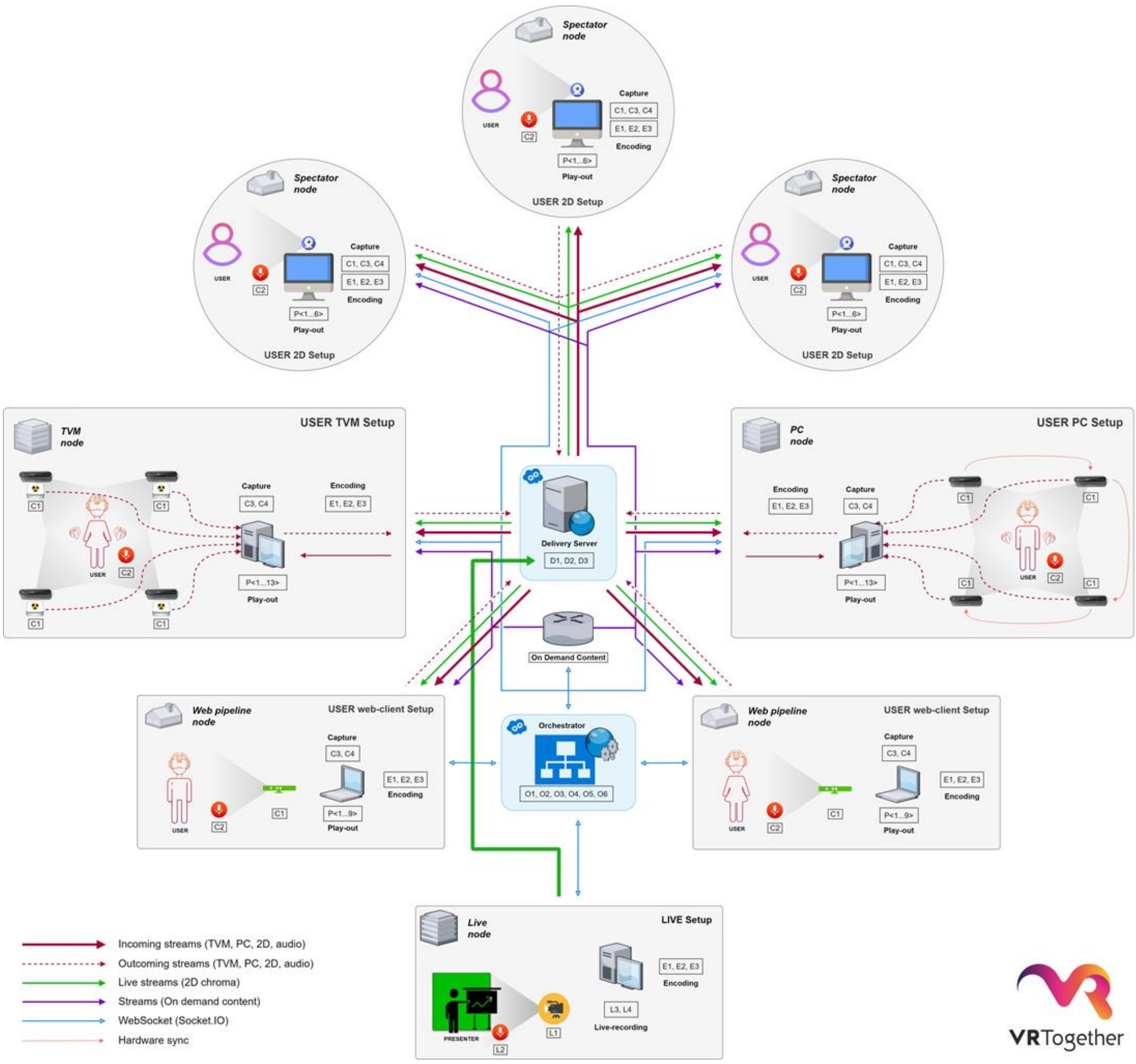
5.3.1.3. Final Hardware Architecture

This section provides the definitive hardware architecture of the VR-Together platform integrating all the Pilots advances as well as the last requirements updates.

It results that for each user capture node, the hardware setup is constituted of a capture rig (one or four RGB-(D) cameras depending on the node type), an encoding component that processes captured stream data and send it to the delivery server, and the play-out setup that retrieves incoming stream data as well as on-demand content, provided by the content database, to finally project all of it to the user. Then the centralized orchestrator server operates web-requests within all the user nodes and delivery server to manage the whole platform.

On top of that, a conceptually unlimited amount of simple user nodes (e.g non volumetric capture base nodes) is considered to get them connected to the VRT platform, as viewers of the experience. This setup is basically constituted of the Play-out setup which allows to render the whole experience to the viewers.

All of this connected nodes network is presented in the following Architecture diagram;



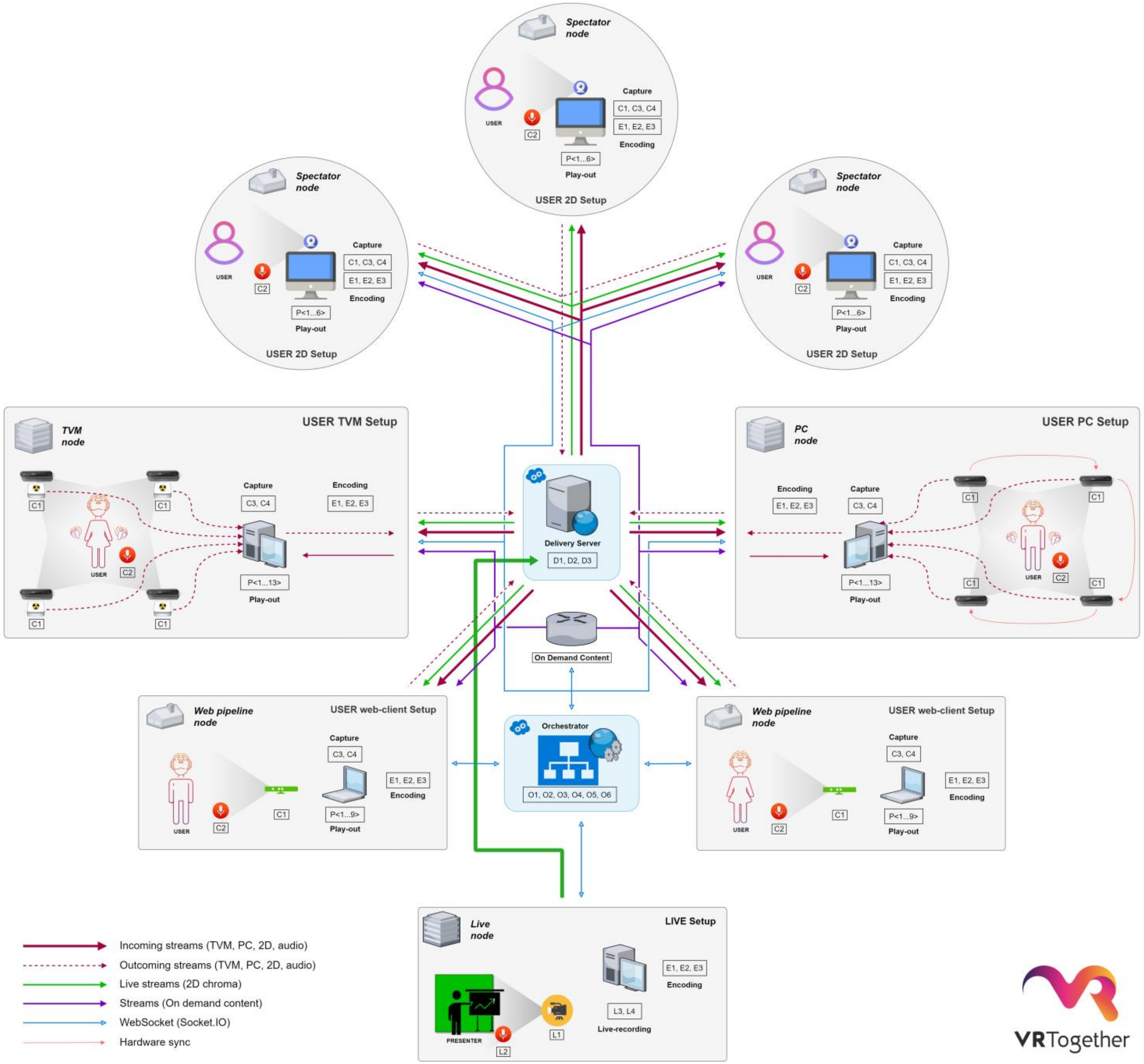


Figure 22: Hardware architecture (Pilot 3)

5.3.1.4. End-user Setups

Each end user needs to be using a capturing setup, an encoding setup and a playout setup in order to be able to use the VR-Together software platform.

First, the capturing setup is composed of:

- **Visual sensor capturing:** the set-up can include either from one to four visual sensors responsible for capturing the user's representation that will be encoded and projected in the virtual rooms of the VR-Together platform. The supported depth-sensors are the Intel RealSense D415© and the Microsoft Azure Kinect©. The deprecated Microsoft Xbox One Kinect© still remains supported. For a 2D capturing setup, a common Webcam device is supported.
- **Audio sensor capturing:** to capture the audio of the user, the sensor to be used is the microphone embedded into the configuration. It can be the one embedded into the head mounted display if the end user is equipped with or an external device (e.g headphones).
- **Capturing and processing units:** for the TVM pipeline, each one of the visual sensors needs to be connected to a separate local terminal that processes the raw input stream. In the current set-up of VR-Together the terminals used for this purpose are Intel NUC D54250.
  For every capture configuration, a central computer is responsible for collecting the separate visual streams together with the audio stream and processing them accordingly. The recommended hardware to use for this purpose is not specific but the most important minimum requirements that it needs to comply with are:
    - 16GB of RAM
    - Intel Core i7 or better equivalent Central Processing Unit
    - Separate graphics processor with at least 8GB of memory

Then the encoding setup is the same one as used for the capturing setup. It is encoding media streams coming from the capturing unit on the same machine.

Finally, the playout setup is composed of:

- **Rendering unit:** the playout hardware machine is in charge of decoding the incoming streams and rendering the whole scene that aggregates every media (volumetric streams, 2D streams, 3D scene assets, videos …).
  For the volumetric pipelines the rendering station is the same one as used for capturing. Note the TVM pipeline may require an additional station dedicated to the experience rendering depending on the computational capacity.
  For the simple user nodes (e.g viewers), as the capturing unit is not required, a single computer in charge of the rendering of the experience is required. A fairly reasonable computer with 3D rendering capabilities is nevertheless recommended;
    - 8GB of RAM
    - Intel Core i5 or better equivalent Central Processing Unit
    - Separate graphics processor with at least 4GB of memory
    - A VR enabled support for an HMD usage
- **Head mounted display (optional):** the playout devices to be used by the end users are commercially available products. The native pipeline involves Oculus HMDs (Rift or Quests using specific cables). The web-pipeline makes use of any HMDs compliant with the OpenXR [8]API (e.g SteamVR headsets, Valve Index, Oculus HMDs, etc.).

[8] https://www.khronos.org/openxr/

For the users who are not equipped with an XR device, a backup rendering mode is proposed by the Playout which allows to enjoy the experience with a legacy screen (e.g Desktop screen, laptop, TVs).

6. User Lab

This section describes the VR-Together user lab activities planned in the VR-Together project in order to feed the requirement definition process and to provide a set of objective and subjective results able to give an outcome about the accomplishments reached. This section describes, in detail, the experiments definition process, together with the physical infrastructures used in the experiments, the methodology followed and the planning of the whole process. The section starts with an introduction showing the general guidelines followed for the experiments process (Section 6.1). The following section (Section 6.2) describes the user labs involved in the experiments and, in the last one (Section 6.4), the experiments definition and planning is explained.

6.1. Introduction

The VR-Together experiments process follows a user-centric approach, in which selected user groups are required to provide feedback, in order to evaluate the platform and obtain new relevant requirements leading to further design and implementations. The user groups considered are:

- Stakeholders: To identify adequate business models and opportunities. The stakeholders are met in public events (fairs, conferences, congresses) and specific stakeholder workshops.
- Experts: To gather requirements about the accomplishments in terms of novelty and about the performance reached. They are consulted internally among the consortium companies and externally at targeted events (fairs, conferences, congresses).
- End Users: To gather a set of requirements able to validate the VR-Together functionalities, the user perception and interaction, and the quality of the representation. The end users are being consulted during the trials, through user lab experiments, via questionnaires and at open demos.

Both stakeholders and experts are part of the professional Advisory Board (https://vrtogether.eu/about-vr-together/advisory_board/). During the second year of the project, the consortium has constituted a new advisory board. This resulted from experiment i2CAT-2.1, a series of focus groups with a number of (local) VR industry professionals / stakeholders to whom were shown the VR-Together platform, with pilot 1 content. A number of selected members of the 25 represented companies (e.g., KLM, KPN, HP, ATOS…) form now the advisory board of the project. This is the current advisory board:

- Xavi Guardia (Soft for You)
- Jordi Colom (Betevé)
- Amadeu Gassó (TV3)
- Sergio Gardella (Bloompix Studios)
- Xavi Conesa (Visyon)
- Francesco D'Andria (ATOS)
- Jae Maloney (KLM)
- Gijs den Butter (SenseGlove)
- George Papagiannakis (ORamaVR)
- Isabelle Lebreton Duval (ARKA Studio)
- Jerome Royan (b<>com)

During the second year the VR-Together consortium has built two permanent collaborative distributed user labs with the equipment needed to run demos but also to test the new releases, analysing development and integration (as detailed in section 6.2). Moreover, the consortium has planned a number of experiments for the third year to pave the way for pilot 3.

6.2. User Lab Nodes

During the second year, there has been a significant effort on the integration and deployment of two connected user labs, one for the web pipeline and two for the native pipeline. Such labs are used for experimentation and showcasing.

Regarding the web pipeline, three nodes have been deployed and connected remotely across The Hague, Paris and Rennes locations (see Figure 23) and have been the subject of multiple experiments sessions which generated useful network metrics in order to assess the socket.io protocol efficiency regarding the VR-Together platform requirements.

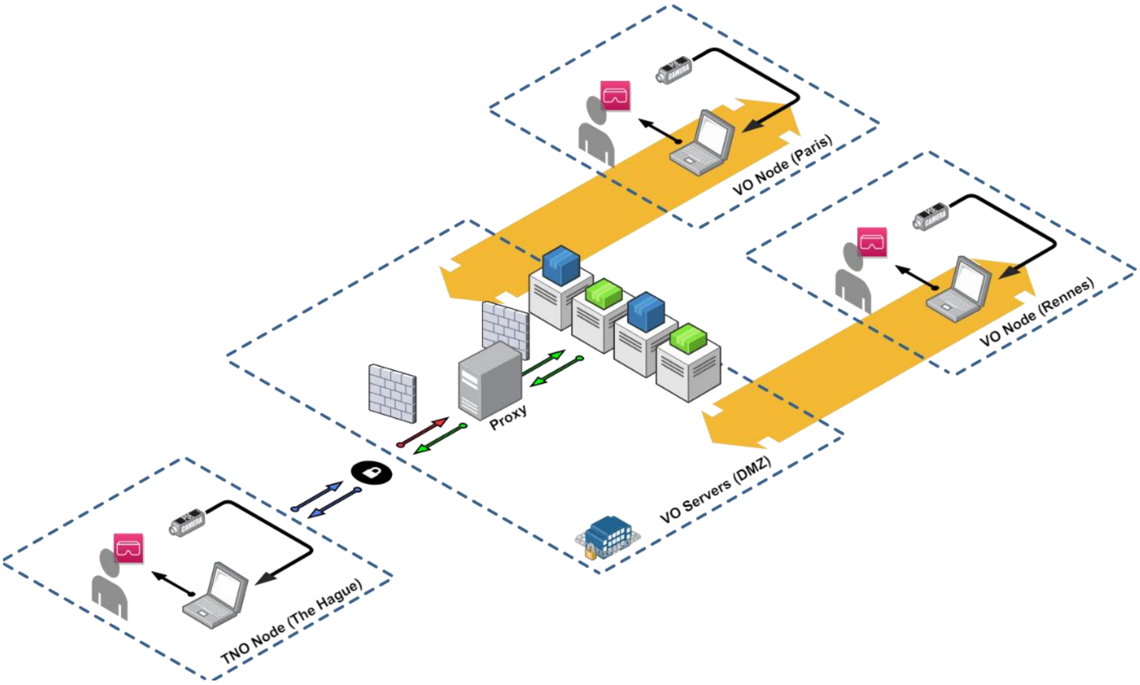


Figure 23: VR-Together web-pipeline nodes network architecture

With respect to the TVM native pipeline, three nodes have been deployed, one in Thessaloniki (CERTH) and two in Barcelona (i2CAT). These three nodes produce a TVM instance each and publish it to a server using RabbitMQ protocol. Each VRT player receives these TVMs from the servers and renders them in real-time. Regarding the networking part, the self-representation of each user is acquired from the local network, while all the other instances are acquired remotely. The entire TVM native pipeline is a mature product of multiple performance experiments, each time monitored with the appropriate metrics leading to continuous improvements through the different versions. Four nodes produce in real-time point cloud instances, using both DASH and socket.io for delivery. Two of the nodes are in Amsterdam (CWI) and the other two in Barcelona (i2CAT).

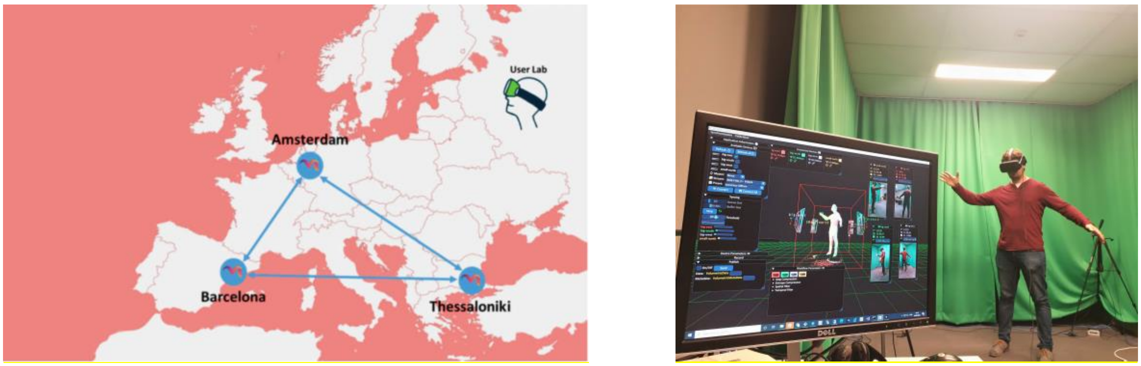


Figure 24: VR-Together native-pipeline nodes

In addition to the connected labs, several partners of the project created dedicated user labs with a partial infrastructure of the full VR-Together platform (see Annex IV). These labs are being used for targeted experiments that will inform about different aspects of the project: QoE, improved reconstruction, comparison of different media types, and production of media assets.

6.3. Experiments

The partners of VR-Together carry out experiments to inform about the different aspects of the project: technology, pilots, and evaluations. These experiments run either in the connected user labs or in the different premises of the partners. In the project we foresee three main categories of experiments, with distinct objectives:

- Assessment of technology, such as HMD removal or content distribution. They serve to path the way for the pilot and have a direct influence on the user hubs under development;
- Subjective quality of experience, mainly based on perception of the medium under different constraints (different compression mechanisms or bandwidth);
- Psychological dimension of the VR-Together experiences, evaluating aspects such as the feeling of being there, as well as the feeling of being together.

All VR-Together experiments are GDPR compliant, as we follow informed consent procedures, protect the privacy of personal data and, to the extent that it is possible, make research data publicly accessible to facilitate further experimentation.

6.4. List of Experiments

The latest updates introduced in this document correspond to the final description of the Experiments provided in the VR-Together project.

The list of experiments planned for Pilot 1 have been published in the previous versions of this document (D2.1 and D2.2), so in this deliverable we will only list and describe the experiments planned for year 3.

The experiments planned for Year 3 will use the same labs and infrastructures designed and organized for Year 1 and 2.

In addition, a detailed description of each experiments, of the process followed to reproduce and validate each of them and of corresponding requirements will be provided also in the deliverable D4.6.

Section 6.4.2 depicts the list of experiments in initially planned for year 3, and the reader is referred to the deliverable D4.4 for a full description of the protocol, methodology, and results of all these experiments. Last, Section 6.4.3 depicts the final list of experiments, actually carried on within the project frame. This final list of experiments is further described in the deliverable D4.6, with the corresponding details related to protocols, methodology and results.

6.4.1. Requirements gathering and feedback from professionals

In the second year of the project, i2CAT coordinated a series of focus groups with (local) VR industry professionals, based on the pilot 1 experience (more details in D4.4). The consortium has the intention of following the same approach during the third year of the project using the content of the second pilot. These focus groups have proven to be effective for gathering feedback, validating requirements, and creating a network of professionals around the project. Moreover, the project will participate, as in previous years, in the next editions of IBC and VRDays, the two more relevant events in Europe about media and VR technology, demonstrating the results of the project. Finally, the consortium will try showcase the three pilots, as complete and coherent story, in a festival before the end of the project.

6.4.2. Initial list of experiments with end users

EXP-VO-3.1 Connect user lab (web)

Connected Lab performance monitoring. the goal is to collect additional statistics related to the performance results of the web-client pipeline.

CERTH-3.1 Connected user Lab (native)

Connected Lab performance monitoring. the goal is to collect additional statistics related to the performance results of the native pipeline.

I2CAT-3.1 Dataset of social VR activities

Creation of a high-quality dataset of point clouds with common social VR activities. This dataset will be submitted to MPEG

CERTH-3.2 Visual comparison between TVM versions

Objective and subjective evaluation of different versions of self-representations for TVMs. Each version will be generated from a different sensor (Intel RealSense and Kinect) and the content

CWI-3.1 Physiological signals for evaluating QoE

Collection of physiological signals (heart rate and galvanic skin response) for the development of objective metrics for predicting QoE of participants.

CWI-3.2 Adaptive Streaming of Point Clouds

Subjective evaluation of the quality of experience achieved by using user-based adaptive streaming of tiled point clouds against standard network adaptive streaming

CWI-3.3 Impact of End-to-End Delay on QoE

Subjective evaluation to better understand the impact of latency on the Quality of experience when two users communicate in a shared space

I2CAT-3.2 Impact of number of participants on experience

Subjective evaluation to better understand the impact of the number of participants on the experience created in pilot 2

I2CAT-3.3 Impact of number type of content on experience

Subjective evaluation to better understand the impact of the type of content (live, recorded) on the experience created in pilot 2

I2CAT-3.4 MCU performance for delivery of volumetric video

Performance evaluation of the MCU in terms of resource consumption, delay, frame rates, and scalability

CERTH-3.3 Reconstruction rate between TVM versions

Evaluation of the reconstruction rate, when comparing the two existing capturing systems: Kinect Azure and Intel RealSense

TNO-3.1 Comparing mobile and non-tethered HMD devices

This experiment will compare tethered HMDs (Oculus Rift) with mobile based VR (Google Daydream, Samsung Gear) HMD and non-tethered devices (Oculus Quest). The comparison will be based on objective performance metrics and Subjective performance metrics.

6.4.3. Final list of experiments with end users

During the third year of the project, the consortium had to be dynamic and adapt to the external circumstances, which lead as well to a re-prioritisation of the tasks and objectives. This is the final list, which will be detailed (protocol, results, analysis) in D4.6.

VRT-3.1 - Benchmarking SocialVR Systems

Benchmark exiting socialVR systems in terms of performance, functionality, and user experience

VO-3.1 - Connected User Labs (web)

Assess Web Pipeline network and rendering performances

CERTH-3.1 - Connected User Labs (native)

Assess Native Pipeline network and rendering performances

CERTH-3.2 - Visual comparison of TVM versions

Evaluation of visual improvement between TVM v2 (with RS) and TVM v3 (with K4A)

CWI-3.1: Focus Groups with Professionals

Showcase the results of the project to the local advisory board

CWI-3.2 - Point Clouds Tiling

Experiment to test subjective quality achieved by using DASH streaming of point clouds with tiling vs without tiling strategy

TNO-3.1 - Evaluating tethered and non-tethered HMD's

To evaluate different types of HMDs, measuring qualitative and quantitative measures

i2CAT-3.1: Evaluation of the MCU

To evaluate the effects of a Multipoint Control Unit (MCU) for Virtual Reality environment, focusing on the holoportation use case

i2CAT-3.2: Subtitles 3D Content

This experiment will assess the impact on the user experience of different presentation modes for subtitles in 3D VR environments, by using the pilot 1 content as stimuli.

i2CAT-3.3: Holoconferencing use case

This experiment will investigate the acceptability and impact on the user experience of the use of the VR-Together technology for holoconferencing (or holomeeting) scenarios, with groups of 2 and 4 users, in a newly modelled 3D meeting room, and using single-camera Point Cloud representations with Azure Kinect sensors.

7. CONCLUSIONS

This document shows an extended description of the current status of the following aspects of the VR-Together development:

- Pilot 3 Production and Technology
- Current Status of the Requirements
- Current Architecture
- User Labs and experiments Status for year 3

More in detail, regarding Pilot 3 production and technology, a short description of Pilot 1 and 2 has been included in Section 3, in order to provide to the reader, the full picture and, then, explaining the updates related to Pilot 3 more in depth.

Regarding the Requirements, the work have followed the method extensively described in D2.2. For this reason, the reader can find in Section 4, the updates from the status in the previous version of the deliverable.

The VR-Together architecture has grown together with the technological development, defining one version for each pilot of the project. Section 5 shows the Architecture defined for Pilot 3 and its status at the moment of the delivery of this document.

Finally, Section 6 describes the users labs activities planned for the last year and the corresponding experiments.

8. Annex I: Project Requirements

In this section we aim to describing the high-level requirements of VR-Together platform. The project can be considered as divided in 3 iterations, each one addressing one technical scenario that will be validated by user groups through 3 pilots. In terms of pilot content, the ones initially foreseen to feed public demos and user evaluations was: an intimate concert, a live news format and a fictional story plot. In terms of technical scenario of each pilot, they were classified as offline, live and interactive respectively. This division allows the project to reach intermediate objectives both at creative and technical level, facilitating the consortium to deal with the challenge of delivering a satisfying social VR experience.

The requirement definition has been updated during the project. This section describes the initial status of the requirements in order to provide to the reader, since the beginning, an understanding of the VR-Together platform purpose. A more detailed definition of the requirements gathering process and of the updated status of them can be found in Section 9. The Requirements provided correspond to the status at the time of the delivery of this document and, as a consequence, the most updated version.

This section is structured as follows: first, we introduce the initial set of general requirements (those requirements that have been valid since the beginning of the project and that all versions of the VR-Together system should meet). Second, we show the different ideas, that were included in the project proposal, able to set the differences between pilots, providing a list of the specific requirements of each pilot. Finally, we compile the initial scenario to be addressed in VR-Together.

8.1. General requirements

VR-Together aims at exploring how the combination of various data streams (content, human representations, data) will result in a highly personalized experience that is delivered in an adaptive manner, enabling individuals in different locations participate together in the same experience. The objective is to deliver close to market prototypes and implement an integrated platform to achieve the main project objective: delivering photorealistic immersive virtual reality content which can be experienced together with friends and demonstrate its use for domestic VR consumption.

VR-Together, as mentioned above, is structured in 3 iterations. After each iteration, the project will deliver a system version that will meet the indicated requirements. Both system and requirements will be validated, and the consortium will evaluate if and to what extent the work done meet each of the requirement. The following table gathers the initial list of general requirements considered by the consortium.

CODE	NUM	TITLE	DESCRIPTION
GEN	1	Co-presence	End users should be able to be virtually present in the same virtual space and engage in real-time face-to-face social activities. Co-presence should lead to other-awareness, social behaviour, responsiveness to one another's actions and self-awareness
GEN	2	Distributed experience	End users should be able to access a shared virtual space from different physical locations (equipped with the corresponding capture and visualization systems)

GEN	3	Number of users per physical space	At least one end user should be able to access a shared virtual environment from a specific physical location (equipped with the corresponding capture and visualization systems)
GEN	4	Natural communication	End users should be able to communicate with each other in a natural, fluid, way. This requires real-time interaction (i.e. transmitting/receiving the other user's graphical representation and voice with imperceptible delay)
GEN	5	End user representation	End users inside a virtual space should be able to see other end users body representation
GEN	6	Self-representation	End users inside a virtual space should be able to see their own body representation
GEN	7	Place illusion	End users inside a virtual space should have the feeling of being in the physical space depicted in the VR content
GEN	8	VR content	End users inside a virtual space should be able to see VR content
GEN	9	VR content formats	End users should be able to see different examples of VR content formats
GEN	10	VR content image quality	End users should be able to see photorealistic VR contents
GEN	11	Synchronization	End users in distributed locations sharing a virtual space should be able to see the same VR content at the same time
GEN	12	End-user image quality	End users should see other users in photorealistic quality
GEN	13	End-user blend	End users should see other users seamlessly blended in the VR content
GEN	14	Perception of VR quality	VR-together should improve the subjective quality of previous Social VR experiences
GEN	15	Comfortability	End users should be comfortable in using the system for at least the duration of the pilot experience
GEN	16	Body language	End users should be able to understand each other’s body language expressions.
GEN	17	Immersive VR audio	The VR audio content should be immersive. That is, when the end user turns the head, audio should change as it does naturally
GEN	18	Audio/Video Synchronization	The VR audio and video content must be synchronized, as in any content experience

GEN	19	End-user audio	The end-user audio for communication should be directional. That is, end-user audio should appear to come from its originating point.
GEN	20	End-user devices	End users should access the experience using commercially available HMDs and capture systems
GEN	21	Data logging	The system has to record end user activity data
GEN	22	Blend of media formats	End users, scene of action and characters should be represented using different media formats. The resulting VR image should be a blend of different formats.
GEN	23	Networks	The VR content and end-user representations need to be delivered over commercial communication and media delivery networks.
GEN	24	Adaptive media delivery	Media streams should provide adaptive quality to network, device and interface capabilities
GEN	25	Web interface	End users should be able to access the experience using a web application.
GEN	26	Native interface	End users should be able to access an experience using a native application

Table 7: General Requirements

8.2. Requirements for Pilot 1

In this subsection we review the initial assumptions to be considered in Pilot 1, as initially planned in the project proposal. As described in the proposal, section 1.3.4.2:

"Pilot 1. Intimate Concert. The goal of the offline pilot is to demonstrate that the innovative media format of VR-Together (orchestrating point clouds, 3DMesh based models and multiple video sources) can produce a more intimate and binding activity than more traditional content production pipelines, based on omnidirectional content. We will compare different capture and production techniques (video, point cloud capture, high-end motion capture) as well as combinations of them to determine quantitative balances among the different formats available (video, point clouds, time-varying meshes, dynamic meshes, motion data). The main variables considered to compare the different means available to deliver such an experience will be:

· *Production costs, integrating shooting, editing, compositing, post-production, etc.*

· *Bandwidth and computational resources required at the different nodes (capture, encoding, delivery, rendering)*

· *Impact on the subjective social experience among end-users.*

Typology of contents addressed: *An intimate music concert seems an ideal starting point to demonstrate VR-Together's innovative media format. It is a good opportunity to show how the VR-Together works for offline produced content. The goal is to demonstrate that the orchestrated delivery of the VR-Together media format, combining several video sources, point cloud and 3D mesh representations will improve closeness with the musicians and with at least 2 distant end users. Particular care will be taken to integrate facial expression within the production pipeline, i.e. how we will capture the photorealistic 3D actors in costume. For example, uses 108 cameras to capture the actors' performance, costumes, facial expressions and the result is a stream-able 3D model with appropriate facial expressions. This also applies to lighter methods, which are more affordable and portable. For example, uses 4 Kinect sensors and a short automatic calibration process. Industrial methods capturing actor facial MoCap performance using marker-less methods and pre-rigged models will also be considered. Different combination of methodologies and technologies will be studied to deliver the best possible balance between visual quality and cost efficiency in content production."*

As described in the proposal, in T4.1, the task that addresses the prototyping and production of demo content:

"Offline CoVR: The content format that we have pre-selected is an intimate concert, which seems relevant to validate the unique feeling of closeness between the audience and the content that the VR-Together platform will deliver. We will also seek to detect implicit social interaction cues that may improve the connection between the audience and the users, such as real-time retargeting of gaze or pointing gestures in the characters being rendered, in order to further integrate the content consumer's presence."

As described in the proposal, in T4.2, the task that addresses the deployment of demos and pilots, with a more practical (technical deployment) approach:

"Offline CoVR: In this first example of content production and delivery, we will focus on validating the staging and capture process to deliver the feeling of co-presence in a shared photo-realistic immersive virtual reality environment. We will study which computer graphics techniques can appropriately blend the representations of end users, created with real-time constraints, home lightning, affordable cameras and sensors for capture, with the offline produced content. Where possible, we will seek to apply re-illumination techniques to blend end-user representations within the pre-recorded content. "

The following table gathers the subset of high-level requirements for pilot 1.

CODE	NUM	TITLE	DESCRIPTION
P1	1	Facial expressions	Some detail to see facial expressions should be available in the end-user and character representations
P1	2	Offline content	The VR content to be displayed must be stored in the end user device
P1	3	Illumination	Illumination should be consistent in the whole experience
P1	4	Gaze	Rendered characters should be able to retarget their gaze according user's viewpoint
P1	5	Pointing gestures	Rendered characters should be able to retarget pointing gestures
P1	6	Rendered Characters	The scene should contain rendered characters
P1	7	Characters' representation	The end user should perceive the 3D appearance of the characters (some parallax, depth)
P1	8	Basic end user movement	Users can rotate their head and have certain level of translation capacity while seated (3DoF+)

Table 8: Pilot 1 requirements

8.3. Requirements for Pilot 2

In this subsection we review the initial assumptions to be considered in Pilot 2 as initially planned in the project proposal. As described in the proposal, section 1.3.4.2:

"Pilot 2. Live news. We will demonstrate the live production of multi-source immersive content. We will study the conditions which maximize the connection between the audience and the news. Numerous benefits for cost-effective production efficiency will be derived from introducing live processing constraints. Quantitative measures comparing the benefits and costs of introducing offline processing steps will be sought. To realize this scenario, we foresee the creation and demonstration of a hybrid live production that combines omnidirectional cameras and depth sensors and off-the-shelf capture devices targeting consumers (webcam, Kinect) in order to allow several users to feel like being together inside an immersive virtual environment and to increase the feeling of connection with the environment thanks to embodied social interaction. In this scenario, inter-stream synchronisation is critical: this is not a live VR conference, but a production broadcast. Technically speaking, we need clock sync between equipment at both production environments and insert / correlate timestamps in the recordings. This kind of activity is aligned with current standardization activities in MPEG MORE, to which part of the VR-Together consortium contributes actively.

<u>Typology of contents addressed:</u> We will demonstrate a novel content format of immersive news consumption, where people can feel like being together where the news actually occurred. For this we will combine more closely the content production expertise (camera placement, social setting between presenters and the audience, how transitions to other settings (for example, a journalist on the field) can be established and delivered comfortably to the audience, etc. The introduction of live delivery for the case of live news will require a production design adapted to the needs and constraints of News Production (Main set with news presenter, live connection with journalist on the field, etc.), but which still allows for a quality of content as close as possible as an offline production."

As described in the proposal, in T4.1, the task that addresses the prototyping and production of demo content:

"Live CoVR: The content format that we have pre-selected is a broadcasted news, which seems relevant to validate the feeling of immediacy that such techniques can deliver. We will however, study other options if real content production opportunities (events, real concerts, etc) appear, and they seem more appropriate for the validation purpose at hand. "

As described in the proposal, in T4.2, the task that addresses the deployment of demos and pilots, with a more practical (technical deployment) approach:

"Live CoVR: In this second example of content production and delivery, we will focus on validating the real-time processing tooling implemented to deliver, at best as possible, the feeling of co-presence in a shared photo-realistic live immersive virtual reality environment. Building upon the insight of first pilot, we will simply aim at assessing to what extent we can preserve the feeling of closeness and empathic connection between the audience and the content, when real-time constraints are imposed. Imposing real-time processing, with no possible offline manual adjustment and manipulation of the content captured severely limits the range of technical possible options. "

The following table gathers the subset of high-level requirements for pilot 2.

CODE	NUM	TITLE	DESCRIPTION
P2	1	Number of users	The system must accept between 2 and 10 end users (in different rooms/locations)
P2	2	Facial expressions	Sufficient detail to see facial expressions should be available in the end-user and character representations
P2	3	Multi-source	The system must be able to produce multi-source immersive content.
P2	4	Live	The system must be able to deliver a photorealistic live immersive VR environment.

Table 9: Pilot 2 requirements

8.4. Requirements for Pilot 3

In this subsection we review the initial assumptions to be considered in Pilot 3 as initially planned in the project proposal. As described in the proposal, section 1.3.4.2:

"Pilot 3. Interactive Fiction. We will seek to demonstrate how the VR-Together platform, in a custom-designed content production process, can allow for a novel form of content where users meet, and blend within the interactive immersive experience. Thus, consumers can watch passively. However, they are also able to, essentially, become a character within the story plot being rendered. They can have this experience through a more active engagement in the experience, i.e., by moving and talking like one of the characters in the plot, and with these actions change significant aspects of the plot being rendered. This will require the combined delivery of broadcast video, mesh or point-cloud content, together with end-user capture in the form of video, point cloud or interpolated 3dmesh, as well as with event-based synchronization similar to how MMO video-games are synchronized. Regarding the integration of advanced multi-modal pattern recognition, the effort will not be on creating sophisticated multimodal pattern recognition of social actions, which would work for any plot, but rather to demonstrate how readily available pattern recognition tools (speech recognition, existing gesture recognition algorithms) can be used and integrated to convincingly deliver one specific plot. For this matter, the previous work done within the VR-Together project, regarding spontaneous social interaction in SIVE will become essential to guide this process. Regarding the processing of interactive plots in SIVE, we will use tools readily available from previous research initiatives by the partners within the consortium. The main challenge to maintain place illusion and plausibility is to render credible interactivity within the experience. We will address how to integrate the user input with the events being depicted within the immersive virtual environment. The goal will be to show to what extent and how a fiction scenario can be rendered in VR, while still allowing the users immersed in the scene to intervene actively in the scene being broadcasted within the shared virtual reality experience (and thus, preserving the feeling of being there together).

<u>Typology of contents addressed:</u> We will address interactive content rendered in the form of interactive fiction. This will be demonstrated as a story-like plot rendered within the immersive experience. The user will be able to actively intervene and change some aspects of the experience by performing some of the actions (i.e., talking, pointing or performing simple physical actions) that correspond to the character he/she wants to become within the plot."

As described in the proposal, in T4.1, the task that addresses the prototyping and production of demo content:

"Interactive CoVR. The content format that we have pre-selected is a fiction production, which will allow for additional control in the production process and will develop a scenario that will be close to a movie script. We will use the insight of subtask T4.3.1 co-presence and social interaction evaluation, in order for the experience of the content to integrate harmonically with possible social interaction occurring, not only among the end users, but also with the content being rendered. The global aim will be to achieve a qualitatively different level of co-presence, social interaction and place illusion in an aesthetically coherent virtual reality experience."

As described in the proposal, in T4.2, the task that addresses the deployment of demos and pilots, with a more practical (technical deployment) approach:

*"Interactive CoVR**.** In this third example of content production and delivery, we will focus on validating the production of explicitly interactive content to maintain, preserve and if possible, reinforce the feeling of co-presence in a shared photo-realistic immersive virtual reality environment. We will seek to detect an expanded range of social and bodily-centred interaction cues (head movements, body movements, peri-personal space, and spoken keywords) to further*

allow the integration of the end users' actions within the narrative. We will integrate existing innovative interactive storytelling engines available within the VR-Together consortium, along with re-illumination, rendering, and interactive character animation techniques. "

The following table gathers the subset of high-level requirements for pilot 3.

CODE	NUM	TITLE	DESCRIPTION
P3	1	Facial expressions	Photorealistic detail to see facial expressions should be available in the end-user and character representations
P3	2	Passive watch	End users can watch the content in a passive way
P3	3	Active watch	End users can become a character within the story plot being rendered
P3	4	Movement	End users can move (translation). 6DoF
P3	5	Derived actions	End user actions change significant aspects of the plot being rendered
P3	6	Pattern recognition	The system must demonstrate how multi modal pattern recognition tools can be used and integrated into the plot.
P3	7	Pointing	End users can trigger story actions with pointing gestures
P3	8	Talk	End users can trigger story actions by talking
P3	9	Physical actions (triggering gestures?)	End users can trigger story actions by performing simple physical actions
P3	10	Interactive storytelling	The system will integrate existing interactive storytelling engines
P3	11	Interactive character	The system will integrate interactive character animation techniques

Table 10: Pilot 3 requirements

8.5. Experimental requirements

The evaluation of the VR-Together platform is organised in two different parts. The first part is concerned with validating the different parameters that need to be preserved or improved. This includes aspects such as delays, resolution, etc. These experiments do not imply specific requirements on the overall platform.

The second part is concerned with validating the feeling of being there, in the virtual environment, and of togetherness, i.e., determining under which technical conditions it can be maximized. This presupposes experiments which involve specific requirements on the end-to-end architecture, which we list below.

CODE	NUM	TITLE	DESCRIPTION
EP1	1	Place illusion under bandwidth and delay constraints	one single end user, through gamepad or wand, can change between different bandwidth and delay constraints, and choose which experience is better, worse, or equal
EP1	2	Place illusion changing content and self-representation formats	one single end user can change his self-representation (static virtual body, dynamic virtual body, 3d-reconstructed mesh) and the media format (omnidirectional video, 3d geometry + stereo billboards, 3d geometry + 3d virtual characters)
EP1	3	Render the other's virtual body is animated or static	To reproduce the Joint Action Effect On Memory (Wagner et al 2017, Eskenazi et al. 2013), the experimenter needs to be able to show the participant to see the other's virtual body either static or dynamic.
EP1	4	Render the other's virtual body at different distances	To reproduce the Joint Action Effect On Memory (Wagner et al 2017, Eskenazi et al. 2013), the experimenter needs to be able to show the participant to see the other's virtual body at different distances.
EP1	5	capture motion data and speech	To find behavioural measures related with togetherness, we need to be able to record the entire multi-modal data, with good time precision.

Table 11: Experimental requirements

9. ANNEX II: D2.2 REQUIREMENTS MATRIX AND PROCESS

This section provides an overview of the goals of the VR-Together project, by the point of view of services provided, technological advance and performance reached; such goals have been written down as requirements to be met by the platform. In the document D2.1 the requirements have been written as *User Scenarios* and *Use Cases* and then recompiled as Functional and Non Functional Requirements; all the information related to the requirements status at the time of the document D2.1 release has been included at the end of this document in the Annex III (0). The current status of the requirements is an updated and refined version of the one mentioned above, according to the outcome of several iterations of refinement and rewriting, in order to provide an accurate description of the status of the VR-Together platform at the time of the release of this document.

9.1. General Requirements Specification

VR-Together is a software platform for an end-to-end communication pipeline between end users in virtual reality. The software is described by a number of requirements that define its functionalities and characteristics. In the following part we lay out the requirements gathering methodology, describing the attributes of the requirements and their meaning. Next, we present the User profiles, depicting the types of users that can use/participate on the platform and also lining up their characteristics. Last, we mention the additional "environment" requirements such as assumptions or Interface requirements.

9.1.1. Requirements gathering methodology

The following part describes the requirements gathering methodology, the attributes of the requirements, how they are prioritized and the distinction regarding the software architecture component that they are referring to.

9.1.1.1. Requirements Gathering Techniques

In the VR-Together project we employ a number of techniques, for the requirements gathering that will define the project's end-results and features of the pilots to be developed. All possible requirements are gathered and classified accordingly by examining the attributes presented in the following sections of this chapter. It is important to mention that all the gathered requirements are judged upon the compliance with the core objectives of VR-Together, as they are described in the Grant Agreement (Section 1.1.2 Part B).

The requirements gathering techniques that we use in VR-Together are:

- Document Analysis: We identify and extract the requirements from documents generated in the VR-Together project, such as deliverables, reports, etc. As a first example of this we identified and extracted the requirements included in the Grant Agreement document. Requirements as "An end user MUST be able to see his own representation in the virtual space of VR-Together" or "An end user MUST be able to see the visual representation of another user in the virtual space of VR Together" are perfect examples of the generic platform requirements defined considering the Grant Agreement.
- Focus Groups: Groups of end users will be asked to perform a specific task on the VR-Together platform. After each focus group gathering, participants are asked to give their feedback on a number of different aspects of the software platform itself as well as the

overall experience. The collected feedback is analysed in an effort to determine additional requirements as well as refine and validate the existing ones. The requirements generated or validated by this part of the process are more focused on the way the end users perceive the experience; examples are requirements like “End users SHOULD feel comfort in being immersed in the virtual space of VR-Together” or “An end user MUST have a network latency allowing for seamless and natural communication and interaction with other users in the virtual space of VR-Together”.

- Interviews: By conducting interviews with end users and other important stakeholders we identify the expectations that VR-Together should meet. The expectations that align with the objectives stated in the Grant Agreement are translated to requirements and captured in the requirements matrix. An indicative example of the requirements that will be gathered with this process are the same as for the Focus Groups but also more specific requirements addressing the objective performance as “The VR-Together hardware capturing component/system MUST achieve a capture rate of at least 25 fps”.
- Surveys/Questionnaires: Carefully designed surveys help in acquiring a large amount of user feedback in a short time as well as in a structured an easily comparable way. The design of the surveys includes questions where options are in the level of agreement/disagreement or rating of an argument. An example questionnaire can be found in Annex I (8).
- Other techniques: Depending on the occasion a number of additional processes could be used in order to generate requirements that would help in developing a higher quality end platform. These additional techniques could be brainstorming sessions, requirements gathering workshops, short interviews and discussions during exhibitions and conferences etc.

The task of requirements gathering is not a finite task with a specifically determined ending point. Thus, by practising the techniques mentioned above we will create new or amend requirements that will be included in future versions of the current document.

9.1.1.2. Types of requirements

VR-Together aims at gathering the software platform requirements from the view point of the end user. For this we are focusing on two types of requirements:

- Functional requirements (FR): Define what the system must accomplish or must be able to do.
- Non-functional requirements[9] (NFR): The required overall attributes of the system, including portability, reliability, efficiency, human engineering, testability, understanding, and modifiability.

9.1.1.3. Prioritization of requirements

The description of a requirement must contain one of the following terms to define the prioritisation of the requirement: “must”, “should”, “could” or “won’t”. The definition of these terms has been adopted from the MoSCoW prioritisation. Negative requirements such as “should not” and “shall not” are omitted, as they are not common in software development.

MoSCoW [1] defines the terms as follows:

[9] A. Davis (1993). Software Requirements: Objects, Functions and States. Prentice Hall.

MUST	Requirements labelled as ***MUST*** have to be included in the solution to be a success. Think of MUST as a requirement that without it the result is considered a failure.
SHOULD	***SHOULD*** requirements are as important as *MUST*, although *SHOULD* requirements are often not as critical or have workarounds, allowing another way of satisfying the requirement. They are important and of high value to the user but even without them the system could still be considered a success.
COULD	Requirements labelled as ***COULD*** are less critical and often seen as 'nice to have'.
WON'T	***WON'T*** requirements are either least-critical or not appropriate at that time.

9.1.1.4. System component of requirements

The user requirements are based on the user scenario compilation, separated depending on the component of the VR-Together platform that they are related to.

The system reference for the requirements are categorised following the components as they are seen in the Architecture diagram included in D2.1 (Section 4.1):

- Capturing (CA)
- Encoding & Encapsulation (EE)
- Delivery (DE)
- Orchestration (OR)
- Play-out (PL)
- VRT (VR Together General)

If a requirement refers to a combination of different components within the platform, then the assigned value in the requirements matrix is: VR-Together (VRT)

The requirements matrix can be found in Section 11 of this document.

9.1.2. VR-Together User Profiles

There are three types of users that interact with the system:

- End users of the native or web player (content consumer),
- Users that can set up, control, monitor and modify the course of the content consumption and social interaction actions (Researcher)
- Administrators

Each of these three types of users has different scope regarding the use of the VR-Together platform and for that reason they have a set of associated requirements as well as available functionalities. Below we give a description for the profile of each one of the different types.

9.1.2.1. End user

The end user of the VR-Together platform is the content consumer of the VR-Together platform. It can be a person of any age, gender and condition, without acoustic or visual impairment and without any previous known problem while accessing contents using Head-Mounted Displays. End users can use the web or native players to access the VR-Together contents, consume them, interact with other users participating in the experience, or interact with the content itself (in future versions of the VR-Together platform).

9.1.2.2. Administrator

The administrator of the VR-Together platform is able to create and set up the VR-Together experiences. Typically, the administrator will be able to set different parameters like the content sources, the available media representation formats used in a specific experience session or room, the format used to represent end users in a specific session, spawn points where end users are located inside a virtual environment, etc. The administrator will configure most of the previous parameters through a relevant graphical interface.

9.1.2.3. Researcher

The researcher in the VR-Together platform is typically a person who will be able to modify parameters of the experience for comparative research and monitor data collection processes. The researcher will also be able to configure specific instances of the players in lab environments.

9.1.3. Reference documentation

The VR Together experience makes use of the following standards:

- Production audio and video will use standards from MPEG to encode and package the content. [MPEG-4 ISO/IEC 14496, MPEG-H ISO/IEC 23008]
- The delivery of production content will use HTTP. [HTTP 1.1 RFC 2616]
- Audio, video, and depth information might be transported using WebRTC [WebRTC RFC 7478]
- Audio, video and 3D point clouds MPEG-DASH [MPEG DASH ISO/IEC 23009]
- 3D meshes will be used with TCP [TCP RFC 793] or a message broker [https://www.rabbitmq.com/]
- WebVR [Draft: https://w3c.github.io/webvr/], Webaudio [https://www.w3.org/TR/webaudio/], WebGL [https://www.khronos.org/webgl/]

9.1.4. Assumptions and dependencies

All the components the VR-Together platform is based on or is dependent from are properly described in D3.1 and D3.2.

9.1.5. Interface Requirements

Web player interface. Content consumption and social interaction will be accessed through a web application. The objective is to explore social VR cases that are easy to deploy, aiming at a social experience without demanding requirements in terms of equipment.

Native player interface. Content consumption and social interaction will be accessed through a native application based on Unity3D. The objective is to explore social VR cases with specific hardware deployments and higher rendering capabilities.

Admin interface for room configuration. The VR-Together system will be configurable by means of rooms. Here an administrator can define the number of end users, user spawn points, content sources and other parameters related to the conditions in which the content consumption and social interaction happen. Additionally, the administrator can perform actions such as the initial calibration of capture system of modification of the capture parameters.

Researcher interface for experiment execution. To select a room and start the experience for different players as well as to modify room or player parameters according to the characteristics and configuration of the experiment to be executed.

9.1.6. Conclusions

This current section represents the list of generic rules that have been followed, and are being followed, for the definition of the final requirements of the VR-Together project. The current requirements status represents the application of these rules together with an iterative process of peer reviewing performed by the partners. The peer reviewing process follows an iterative application of the feedback of the experts of each field and it is fully tracked in the corresponding document *VR-Together Requirements Matrix*[10]. The next section (9.2) describes how each iteration is followed and how they generate an outcome that is the starting point for the following iteration.

9.2. Requirements update iterations

As explained in the introduction of this chapter, the requirement status at the time of the writing of this document corresponds to the outcome of several iterations of work done to improve the requirements initially stated and to adapt them better to the actual development of the platform.

In this section, the reader can find the table corresponding to the requirements status corresponding exactly to the time of the release of this document. The updated requirements have been discussed and processed among the partners of the projects following a well-defined process.

First of all, for each requirement, an owner has been identified depending on the area of expertise and responsibility; for example, regarding the production part, Entropy was responsible for most of the requirements. Each owner, or group of owners, is considered responsible also for the decision taken for each requirement, such as update, deprecation etc.

In order to reach an agreement a first iteration has been processed analysing all the requirements and checking if, for each of them, certain parameters were respected. The parameters considered were about the wording of the requirements, how clear they were, if they needed to be split into more requirements, if they were out of scope or not and if they were specific requirements for one pilot or for more. This first iteration ended with a proposed action for each requirement (i.e. deprecation, refinement, etc.).

Afterwards, the second step has been to receive a feedback, about the proposed actions, directly from the partners responsible for each requirement. The process has followed a traditional Requirement peer review process where every partner involved in the decision was in charge to perform one of the following possible actions:

- Confirming the requirement
- Suggest rephrasing the requirement
- If needed, and if a rephrasing was requested, suggest the new requirement
- Suggest if the requirement needed to be split into 2 or more requirements
- If needed, and a splitting was requested, suggest the new requirements
- Suggest deprecation

[10] VR-Together Requirements Matrix https://docs.google.com/spreadsheets/d/12IF74tYmMmCin5_hAgm0PL4ajia6li-WQ3fS-NWS5q0/edit?ts=5c87c8f2#gid=196101635

- Confirm any of the above actions

All the discussion carried over to reach a decision has been fully tracked in the document *VR-Together Requirements Matrix*[11], where it is possible to see, in each tab, the status at the different stages of the requirements process. In particular, in the tab "D2.1 Analysis" we have followed a peer reviewing method tracking the discussion needed to reach an agreement about each single requirement. Those requirements deprecated have been struck-through in order to leave them in the history of the project and to avoid reusing the deleted ID's. For those requirements that have been updated, the peer review reaches an agreement with the definition of the new requirement and of an ID equal to the previous one plus a 1 in the suffix (i.e. FR.42.0 becomes FR.42.1).

In order to show an example of how the discussion have been tracked, the requirement of Table 12 has been provided. It is possible to notice that the discussion has been tracked writing a tag for every comment, indicating the date and the initials of the author of the comment or of the action (i.e. GC 19.03.2019). In addition, in this specific case, the requirement has been deprecated and three new requirements have been created. The old ID of the deprecated requirement has been eliminated and it will never be reused, and the three new requirements have three new IDs.

ID	Description	Suggestions
~~NF.71.0~~	~~The VR-Together platform SHOULD allow a low quality point cloud to be decoded from a partial bitstream~~	[GC 19.03.2019 Agreed. NF.71 deprecated. New req: FR.152 The Point Clouds (PC) used to represent the VR together end user content COULD be transmitted using an adaptive bitrate streaming technique. NF.153 The adaptive bitrate streaming technique used should be the Dynamic Adaptive Streaming over HTTP (DASH) NF.154 The DASH adaptation set SHOULD include a low quality version of the PC user representation, for the DASH client to download it when the bandwidth conditions are bad NF.155 The DASH server COULD provide (as an option) a tile-based adaptation set, for the DASH client to download only the portion of the PC that falls in her/his field of view. [FDS 18.03.2019] New proposed reqs: 1. The DASH adaptation set must include a low quality version of the PC user representation, for the DASH client to download it when the bandwidth conditions are bad 2. The DASH server must provide (as an option) a tile-based adaptation set, for the DASH client to download only the portion of the PC that falls in her/his field of view. [GC 14.03.2019] I suggest deprecating and add, if needed, new requirements according to the comments of FDS. CWI please agree and provide new reqs. [FDS 10.03.2019] I guess this applies only to the PC pipeline, thus to the Unity-based solution only (but to be checked with CERTH in case TVM encoder allows scalability). The CWI PC encoder does not have any scalable functionality: so, if this requirement refers to scalable video decoding (as it seems) I do not think this can be achieved within the project duration. On the other hand: a low quality PC version can be downloaded by the DASH client when the adaptation set includes a low quality

[11] VR-Together Requirements Matrix https://docs.google.com/spreadsheets/d/12IF74tYmMmCin5_hAgm0PL4ajia6li-WQ3fS-NWS5q0/edit?ts=5c87c8f2#gid=196101635

		version of the PC; a portion of the PC can be downloaded by the DASH client when tiling is used to encode the PC. In both these cases, there is no scalability (i.e., the low quality point cloud is NOT decoded from a partial bit stream). [GC 26.02.2019] In addition, I suggest to change to a FR [MM 24.02.19] See comments on the left (refinements)

Table 12: Example of the discussion followed to process a Requirement

When all the parties involved have come up with an agreement about each requirement, the process have been considered closed and the new requirements were included in the new matrix that can be found in the requirement document, in the tab "D2.2 Proposal". Most of the requirements IDs have been maintained as in the previous version (document D2.1 and Annex III (0)). Those requirements that where not considered as perfectly describing the feature mentioned, have been rephrased and updated. Some requirements have been deprecated because redundant or out of scope; in this case the ID have been removed and will never be reused for another requirement in order to provide a perfect traceability of the project evolution. The new requirements included have been defined with new and unique IDs.

The following iteration has focused on the clustering of the requirements. The fields "Component" and "Title" have been processed with a twofold goal: first, all the partners have been requested to provide a feedback about the correctness of the components the requirements were assigned to, second the titles have been corrected and redefined so that they can represent a second clustering, more specific when compared to the components. In this way, all the tasks related to the requirements analysis will be facilitated by the new classification and the partners will be able to select only the requirements related to specific functions and components. An example of the new clustering process can be found in Table 13. The old Titles and Components, when updated, have been deprecated by a strike-through so that a backward tracking of the previous stage is always feasible. The new clustering can be found in the requirements document in the tab "D2.2 New Clustering" and, in this document, in Table 13.

ID	Component	Title	Component Correct?	Title Correct?	Title and Component Comment
FR.28.0	VRT	~~Facial expressions end users HMD removal~~ VR experience	Yes	Yes	[FDS 29.04.2019] Title could be "HMD removal" maybe? or if you want to cluster under same title multiple items than you could use "Virtual Experience", "User representation" or "Virtual content image quality". Component should be CA imo.
FR.30.1	~~VRT~~ PL	~~Offline content~~ VR Content	Yes	Yes	[GC 24.04.2019] Suggest changing component to PL. Suggest changing the Title to "VR Content" as the req mentions exactly that is not only offline

Table 13: Example of the new clustering process

At the moment of the release of this document, the following stage has already started. The current work is focusing on defining the performance values to be reached by the VR-Together technology. For some requirement it was already possible to write a new definition (i.e. NF.72.0

in Table 14) so that there is already a new proposal for many of them. The new proposal is under peer reviewing process similar to the previous iterations.

For other requirements, an experiment is needed before defining the final performance thresholds. So, currently, those requirements have been linked to one, or more, of the experiments planned for the second year of the project. In addition to the process mentioned above, apart from the previous list of responsible for each requirement, that was mentioning only the names of the companies, one, or more, partners have been nominated as responsible and they will be in charge of defining the values based on the outcomes of the experiments. An example of a requirement following this process is NF.75.0 (Table 14).

ID	Description	Update Suggestion	Experiment Linked	Responsible
NF.72.0	The VR-Together platform MUST be able to achieve a compression ratio of up to 1:10 in point cloud streams	The VR-Together platform MUST implement a volumetric video encoding system able to transmit a bit-stream that meets the target scenario bandwidth availability		Jack/Spiros
NF.75.0	The VR-Together platform SHOULD be able to evaluate the expected quality of experience according to the objective metrics TBD	To be written based on the outcome of the experiment	CWI2.1/CWI2.2	Jie

Table 14: Example of the current requirements update end linking to the experiments

In this same iteration, the process to define the final values needed to fully address the requirements is being performed. The partners have analysed the requirements one by one and have assigned, to each of them, one of the experiments planned for the second year of the project. The outcome of this iterations is tracked in the tab "Experiments/Requirements Links" of the *VR-Together Requirements Matrix*[12] document and an example of the linking can be found in Table 15.

Experiment Id	Linked Non-Functional Requirements	Linked Functional Requirements
i2CAT-2.1	NF.104.1, NF.107.0, NF.109.1, NF.150.0, NF.66.1, NF.96.1, NF.122.0	FR.1.0,FR.2.0,FR.3.0,FR.5.0,FR.15.1, FR.16.0, FR.17.0, FR.27.1, FR.28.0, FR.31.0, FR.37.0

Table 15: Experiments/Requirements Linking example

[12] VR-Together Requirements Matrix https://docs.google.com/spreadsheets/d/12IF74tYmMmCin5_hAgm0PL4ajia6Ii-WQ3fS-NWS5q0/edit?ts=5c87c8f2#gid=196101635

ID	Component	Title	Description	Priority	Responsible
FR.1.0	PL	Self-representation	An end user MUST be able to see his own representation in the virtual space of VR-Together	MUST	CERTH, CWI, TNO
FR.2.0	PL	Users audio representation	An end user MUST be able to hear the sounds made by another user in the virtual space of VR Together	MUST	i2CAT/ENTROPY, CERTH, CWI, TNO, MSE
FR.3.0	PL	Users representation	An end user MUST be able to see the visual representation of another user in the virtual space of VR Together	MUST	CERTH, CWI, TNO
FR.4.0	CA	Audio Capturing setup	A location where the VR-Together platform's capturing setup is deployed MUST capture the audio generated by the user	MUST	i2CAT/ENTROPY, CERTH, CWI, TNO
FR.5.0	CA	Visual Capturing setup	A location where VR-Together platform's capturing setup is deployed MUST capture the visual representation of the user	MUST	CERTH, CWI, TNO
FR.8.0	DE	Latency	An end user MUST have a network latency allowing for seamless and natural communication and interaction with other users in the virtual space of VR-Together	MUST	CERTH, CWI, TNO, MSE
FR.9.1	OR	VR Scenario	An end-user client MUST be able to create a reconstruction of the virtual space of VR-Together.	MUST	ENTROPY, i2CAT, TNO
FR.11.0	VRT	VR content formats	End users SHOULD be able to see different examples of VR content formats	SHOULD	ENTROPY, i2CAT, ARTANIM, TNO
FR.12.1	VRT	VR content visual quality	End users MUST be able to see photorealistic VR contents	MUST	ENTROPY, i2CAT, TNO, CWI, CERTH
FR.13.0	VRT	Synchronization	End users in distributed locations sharing a virtual space MUST be able to see the same VR content at the same time	MUST	TNO, i2CAT
FR.15.1	PL	VR content visual quality	End users SHOULD see other users seamlessly blended in the virtual space of VR-Together. The Seamlessness evaluation will be performed by TBD.	SHOULD	ENTROPY, i2CAT

FR.16.0	VRT	VR Experience	End users SHOULD feel comfort in being immersed in the virtual space of VR-Together, at least for the duration of the pilot experience	SHOULD	-
FR.17.0	VRT	VR Experience	An end user SHOULD have an experience that visually and acoustically allows to perceive and understand the other participants' body language expressions.	SHOULD	TNO, CWI, CERTH, i2CAT
FR.18.0	PL	VR Content	The VR audio content MUST be directional giving the perception of point sources within the virtual space of VR-Together.	MUST	TNO, i2CAT
FR.20.0	VRT	End-user devices	End users MUST be able to access the VR-Together platform by using commercially available HMDs and capture systems	MUST	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI
FR.22.0	PL	VR content visual quality	End users, scene of action and characters SHOULD be able to be projected in the virtual space of VR-Together using different media formats. The resulting VR image should be a blend of different formats.	SHOULD	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI
FR.23.1	DE	Networks	The data transmission within VR-Together MUST be using commercial communication (e.g. MPEG-DASH) and media delivery networks (e.g. CDNs)	MUST	MSE
FR.24.0	EE	Networks	Media streams SHOULD provide adaptive quality to network, device and interface capabilities	SHOULD	MSE, CWI, CERTH, TNO?
FR.25.0	VRT	Web interface	End users MUST be able to access the VR-Together platform using a web application.	MUST	TNO
FR.26.0	VRT	Native interface	End users MUST be able to access VR-Together platform using a native application	MUST	i2CAT, ENTROPY
FR.27.1	VRT	VR Content Visual Quality	The virtual character representation MUST be detailed enough to allow for the recognition of facial expressions.	MUST	ENTROPY, ARTANIM, TNO, i2CAT

FR.28.0	VRT	VR experience	The level of detail of end-user representation in the virtual space of VR-Together MUST allow the recognition of facial expressions.	MUST	CERTH, CWI, TNO
FR.30.1	PL	VR Content	The VR-Together platform MUST be able to display the VR content which, depending on the configuration, can be either i) local or ii) stored in a network server	MUST	ENTROPY i2CAT, TNO
FR.31.0	VRT	VR Content	Illumination MUST be consistent in the whole experience	MUST	ENTROPY
FR.32.1	VRT	VR Content	The representations of the rendered characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget their gaze according to the end-user's viewpoint	MUST	ARTANIM, ENTROPY, i2CAT, TNO
FR.33.1	VRT	VR Content	The representations of the rendered pre-rigged characters, inside the virtual space of VR-Together Pilot 3, MUST be able to retarget pointing gestures	MUST	ARTANIM, ENTROPY, i2CAT, TNO
FR.35.0	PL	VR content visual quality	The representations of the rendered characters inside the virtual space of VR-Together MUST have parallax and depth to allow for a 3D representation.	MUST	ARTANIM, ENTROPY, i2CAT,TNO
FR.36.0	PL	VR content visual quality	The end user inside the virtual space of VR-Together MUST be able to perceive the 3D appearance of the characters (parallax, depth)	MUST	ARTANIM, ENTROPY, i2CAT, TNO
FR.37.0	VRT	VR Experience	The end user inside the virtual space of VR-Together MUST be able to rotate their head and have certain level of translation capacity while seated (3DoF+)	MUST	i2CAT, ENTROPY
FR.41.1	OR	Active watch	The end user inside the virtual space of VR-Together Pilot 3 MUST be able to become a character within the storyline that is being projected	MUST	
FR.42.1	VRT	Movement	The end user inside the virtual space of VR-Together Pilot 3 MUST be able to move (translation). 6DoF	MUST	

FR.43.1	OR	Derived actions	The end-user's actions inside the virtual space of VR-Together Pilot 3 MUST lead to changes in the storyline that is being projected	MUST	
FR.44.1	VRT	Pattern recognition interaction	The VR-Together Pilot 3 platform MUST support multi modal pattern recognition mechanics for changing the storyline according to user's choices	MUST	
FR.45.1	VRT	Pointing interaction	The VR-Together Pilot 3 platform MUST be able to recognize pointing gestures of end users and change the storyline accordingly	MUST	
FR.46.1	VRT	Speech interaction	The VR-Together Pilot 3 platform MUST be able to recognize the speech of end users and change the storyline accordingly	MUST	
FR.48.1	VRT	Interactive character	The system for Pilot 3 SHOULD integrate and use interactive character animation	SHOULD	ARTANIM
FR.49.0	VRT	Networks	The VR-Together platform MUST support bandwidth configuration options for the end user	MUST	TNO, i2CAT, CWI, CERTH, MSE
FR.50.0	VRT	Networks	The VR-Together platform MUST support delay constraint configuration options for the end user	MUST	TNO, i2CAT, CWI, CERTH, MSE
FR.51.1	VRT	Self-representation	The VR-Together platform MUST support self-representation projection configuration options for the end user.	MUST	TNO, CWI, CERTH
FR.54.0	VRT	VR Scenario	The VR-Together platform MUST allow one end user to see a dynamic projection of another end-user's body representation within the virtual space.	MUST	CWI, CERTH, TNO
FR.55.0	VRT	VR Scenario	The VR-Together platform MUST allow one end user to see the projection of another end-user's body representation positioned at various distances within the virtual space.	MUST	i2CAT, TNO
FR.57.0	CA	**RGB-D capture**	The VR-Together hardware capturing component/system MUST capture RGB-D data from 4 RGB-D devices connected to 4 capturing nodes (RGB-D nodes)	MUST	CERTH
FR.58.0	CA	RGB-D Capture	The VR-Together hardware capturing component/system RGB-D devices SHOULD be automatically calibrated (extrinsic calibration).	SHOULD	CERTH

FR.59.0	CA	RGB-D Capture	The RGB-D frames from the RGB-D nodes MUST be synchronized and grouped in a central node, resulting in an RGB-D group frame.	MUST	CERTH
FR.60.0	CA	VR content formats	The VR-Together platform MUST process end-user's live coloured 3D point cloud to reconstruct a 3D time-varying mesh in real-time.	MUST	CERTH
NF.66.1	CA	Latency	The input image captured by the hardware sensors of the capturing component MUST use a framerate of at least 25 fps.	MUST	CERTH, TNO
NF.67.0	CA	Face capture	The VR-Together hardware capturing component/system MUST capture the end-user's face from at least two different sides.	MUST	CERTH, TNO
NF.72.0	EE	Compression	The VR-Together platform MUST be able to achieve a compression ratio of up to 1:10 in point cloud streams	MUST	CWI
NF.73.1		Latency	The VR-Together platform MUST achieve an end to end (capture to projection) latency that is lower than TBD.	MUST	CERTH, CWI
NF.74.1	EE	Compression	The VR-Together platform SHOULD support point cloud compression of arbitrary topology (Topology TBD).	SHOULD	CWI
NF.75.0	VRT	Quality assessment	The VR-Together platform SHOULD be able to evaluate the expected quality of experience according to the objective metrics TBD	SHOULD	CWI, i2CAT
NF.77.0	EE	Compression	The VR-Together platform SHOULD be able to achieve a compression ratio of up to 1:30 for textured mesh (3D geometry and textures) content	SHOULD	CERTH
NF.78.0	EE	Compression	The VR-Together platform MUST support compression for textured 3D time varying mesh content of arbitrary topology.	MUST	CERTH
NF.92.1	OR	Configuration	The VR-Together platform orchestration module MUST be able to configure the native end-user play-out component	MUST	MSE, VO
NF.95.0	PL	VR content visual quality	The VR-Together platform MUST support playback of end-user's representation of at least 960x540 pixels	MUST	CWI, CERTH, i2CAT, ARTANIM, ENTROPY, TNO

NF.96.1	PL	VR Content Visual Quality	The VR-Together platform MUST support playback of end-user’s representation at a framerate of at least TBD fps.	MUST	CWI, CERTH, i2CAT, ARTANIM, ENTROPY, TNO
NF.99.0	PL	WebVR	The VR-Together play-out component's web player MUST operate in a browser that supports WebVR and A-frame.	MUST	TNO
	PL	VR Content Format	The VR-Together play-out component's native player MUST support the reproduction of hybrid VR contents (TBD what is the hybrid VR content) in virtual space.	MUST	ENTROPY, ARTANIM, i2CAT, MSE
NF.107.0	PL	VR Content Visual Quality	The VR-Together play-out component's native player SHOULD be able to alter the lighting of specific objects within the virtual space, on the basis of custom shaders.	SHOULD	i2CAT, ENTROPY
NF.109.1	PL	VR Content Visual Quality	The VR-Together play-out component's native player stereo effective display resolution MUST be up to 4K.	MUST	i2CAT, ENTROPY
NF.110.1	VRT	Latency	The VR-Together play-out component's native player self-representation projection MUST have latency under TBD.	MUST	CERTH, CWI, TNO
NF.111.1	PL	Synchronization	The VR-Together play-out component's native player SHOULD support synchronization between different input formats with less than TBD of delay	SHOULD	CERTH, CWI, i2CAT, ENTROPY, MSE
NF.112.1	PL	Synchronization	The VR-Together play-out component's different players SHOULD support synchronization of frame accurate with a delay lower than TBD.	SHOULD	i2CAT, ENTROPY, MSE
FR.115.0	CA	Quality Assessment	The VR-Together platform capturing component SHOULD record and store the recordings of the HMD for further future analysis purposes.	SHOULD	i2CAT, ENTROPY
FR.117.0	OR	Configuration	The VR-Together platform orchestration component MUST support remote operation.	MUST	MSE, VO

FR.118.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 1, MUST manage sessions where 2 end users participate in a virtual space.	MUST	MSE, VO
FR.119.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, MUST manage sessions where at least 2 end users participate in a virtual space.	MUST	MSE, VO
FR.120.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 1, SHOULD support at least one session.	SHOULD	MSE, VO
FR.121.0	OR	Configuration	The VR-Together platform orchestration component, for Pilot 2 and 3, SHOULD support at least two parallel sessions.	SHOULD	MSE, VO
NF.122.0	CA	RGB-D Capture Framerate	The VR-Together hardware capturing component/system MUST achieve a capture rate of at least 25 fps.	MUST	CERTH
NF.123.0	CA	Latency	The VR-Together platform MUST perform the People live 3d reconstruction with a delay lower than 80ms.	MUST	CERTH, TNO
FR.124.0	CA	VR experience	The VR-Together hardware capturing component/system MUST store the captured end-user's face data. The information must be stored (on disk or in memory) and must be accessible in real-time by the face inpainting algorithm.	MUST	CERTH, TNO
FR.125.0	CA	Benchmarking	The VR-Together platform MUST record the position of the end user in the 3D scene at regular time intervals.	MUST	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI, MSE
FR.126.0	CA	Benchmarking	The VR-Together platform MUST record the viewport video visualized by each end user with timestamp.	MUST	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI, MSE
FR.127.0	CA	Benchmarking	The VR-Together platform MUST record the audio information (speech) from the end user with timestamp.	MUST	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI, MSE

FR.128.0	OR	VR Scenario	The VR-Together Pilot 1 platform MUST allow for 2 end users to simultaneously be in the same virtual space.	MUST	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI, MSE
FR.129.0	OR	VR Scenario	The VR-Together Pilot 2 platform MUST allow for 4 end users to simultaneously be in the same virtual space.	MUST	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI, MSE
FR.130.0	OR	VR Scenario	The VR-Together Pilot 3 platform MUST allow for 10 end users to simultaneously be in the same virtual space.	MUST	ENTROPY, i2CAT, ARTANIM, TNO, CERTH, CWI, MSE
FR.132.0	VRT	Compression	The VR-Together platform MUST support TVM compression configuration options for the end user.	MUST	CERTH
FR.134.0	PL	VR Content Visual Quality	The VR-Together play-out component's native player SHOULD be able to alter the lighting of specific objects within the virtual space, on the basis of custom shaders.	SHOULD	i2CAT, ENTROPY
FR.135.0	PL	VR Experience	The VR-Together play-out component's native player MUST be able to reproduce content adapted to 3DoF or 3DoF+ movements.	SHOULD	i2CAT, ENTROPY
FR.136.0	EE	Compression	The VR-Together platform MUST use typical browser supported audio encoding.	MUST	TNO, MSE
FR.137.0	EE	Compression	The VR-Together platform MUST use typical browser supported video encoding.	MUST	TNO, MSE
FR.138.0	EE	Compression	The VR-Together platform MUST use typical browser supported audio encapsulation.	MUST	TNO, MSE
FR.139.0	EE	Compression	The VR-Together platform MUST use typical browser supported video encapsulation.	MUST	TNO, MSE
FR.140.0	PL	VR Content	The VR-Together platform web player MUST support playback of 2D VR video content.	MUST	TNO
FR.141.0	PL	VR Content	The VR-Together platform web player MUST support playback of 2D end-user representation projection.	MUST	TNO

FR.142.0	PL	VR Content	The VR-Together play-out component platform SHOULD support spatial audio.	SHOULD	i2CAT, ENTROPY, TNO
FR.143.0	PL	VR Content	The VR-Together play-out component MUST support input of separate VR content and end-user representations streams.	MUST	CWI, CERTH, MSE, TNO
FR.144.0	PL	Network	The VR-Together play-out component's web player SHOULD support content bandwidth adaptation.	SHOULD	TNO
FR.145.0	CA	Synchronization	The VR-Together platform capturing component MUST timestamp media content in relation to a platform-wide common clock.	MUST	i2CAT, ENTROPY, MSE, CWI, CERTH
FR.146.0	PL	VR Content	The native player MUST support play-out of content for Point Clouds.	MUST	ENTROPY, ARTANIM, i2CAT
FR.147.0	PL	VR Content	The native player MUST support play-out of content for Static/Dynamic meshes.	MUST	ENTROPY, ARTANIM, i2CAT
FR.148.0	PL	VR Content	The native player MUST support play-out of content for mono/stereo 2d video.	MUST	ENTROPY, ARTANIM, i2CAT
FR.149.0	PL	VR Content	The native player MUST support play-out of content for 360 video.	MUST	ENTROPY, ARTANIM, i2CAT
NF.150.0	PL	VR Content Visual Quality	The VR-Together play-out component's native player mono effective display resolution MUST be up to 2K.	MUST	i2CAT, ENTROPY
NF.151.0	OR	Configuration	The VR-Together platform orchestration module SHOULD be able to configure the web-client end-user play-out component.	MUST	MSE, VO
FR.152.0	PL	Network	The Point Clouds (PC) used to represent the VR together end user content COULD be transmitted using an adaptive bitrate streaming technique.	COULD	CWI
NF.153.0	PL	Network	The adaptive bitrate streaming technique used SHOULD be the Dynamic Adaptive Streaming over HTTP (DASH)	SHOULD	CWI

NF.154.0	PL	Network	NF.154 The DASH adaptation set SHOULD include a low-quality version of the PC user representation, for the DASH client to download it when the bandwidth conditions are bad	SHOULD	CWI
NF.155.0	PL	Network	NF.155 The DASH server COULD provide (as an option) a tile-based adaptation set, for the DASH client to download only the portion of the PC that falls in her/his field of view.	COULD	CWI
FR.156.0	VRT	VR content	End users SHOULD be able to see photorealistic Live content (mono or stereoscopic video) in the VR environment	MUST	ENTROPY, i2CAT, TNO, CWI, CERTH

Table 16: D2.2 Requirement Matrix

10. ANNEX III. END USER QUESTIONNAIRE USED IN VR DAYS EVENT

VR Together

Our mission is to make VR experiences a social space, where you can share and communicate with your family or friends and to experience VR together.

VR Together is an European research project (funded by the EU). In this project we will create an end-to-end system for the production and delivery of photorealistic and social virtual reality experiences.

With this questionnaire we like to get your feedback about some of the research we like to address in VR Togehter.

Thank you for your time, it will only take a few minutes.

We really appreciate it.

Regarding what you just experienced, how would you rate...

1. ...the video quality?
Mark only one oval.

	1	2	3	4	5	6	7	
Very bad								Very good

2. ...the audio quality?
Mark only one oval.

	1	2	3	4	5	6	7	
Very bad								Very good

3. ...the overall experience?
Mark only one oval.

	1	2	3	4	5	6	7	
Very bad								Very good

Experiences with VR

4. Have you ever experienced VR before?
Mark only one oval.

- Yes
- No

5. Are you interested in Social VR experiences?
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all								Absolutely

Would you like to experience the following topics in Social VR?

6. *Mark only one oval per row.*

	Not at all interested	low interest	Slightly interested	Neutral	Moderately interested	Very interested	Extremely Interested
Sports							
Movies							
Theatre							
Videogames							
Education							
Music experiences							
Live TV shows							
Videoconferencing							
Dating							
Adult entertainment							

7. Is there anything else you would like to experience within a VR environment?

In a VR experience, how important would it be for you to...

8. ...share the experience with someone?
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all								Absolutely

9. ...interact within the experience?
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Absolutely

10. ...enjoy the overall the experience?
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Absolutely

11. ...being able to move within the experience?
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Absolutely

Demographic questions

12. Gender
Mark only one oval.

- ◯ Female
- ◯ Male
- ◯ Other: ____________

13. Age
Mark only one oval.

- ◯ Less than 18
- ◯ Between 18 and 30
- ◯ Between 30 to 45
- ◯ Between 45 and 65
- ◯ More than 65
- ◯ Other: ____________

14. Are you interested in this VR project? If so...
Check all that apply.

- ☐ I would like to receive updates about the project
- ☐ I would like to participate in user studies
- ☐ I would like to give my expert input / feedback
- ☐ Other: ____________

15. Email

____

16. Do you have any other comments or information you like to share with us?

Powered by

11. ANNEX IV: VR-TOGETHER LAB NODES

In the following part some of the existing infrastructure for the labs is presented in order to show where different partners will perform targeted evaluations.

11.1. Artanim Lab Node

Figure 25: Artanim's User Lab

Artanim is housed within a facility of over 273 m² with a motion capture studio of the following size: 15 m x 8 m x 3.7 m (see Figure 26 and Figure *26*). The lab is equipped with diverse high- and low-end motion capture equipment and VR/AR equipment:

- Vicon MXT40S with 24 cameras (up to 515 fps)
- Xsens MVN 17 MTx inertial trackers
- RGB-D cameras
- Variety of head mounted displays (HMD): Oculus CV1, HTC Vive, HoloLens (see-through HMD).
- Set of 6 HTC VIVE trackers

The lab is also equipped with a photogrammetric 3D scanner comprising 96 cameras for polygonal mesh reconstruction of users and objects. For production and VR/AR applications, Artanim uses a full range of software: Vicon Blade, Vicon Tracker, MVN Studio, Autodesk Creation Suite (3ds Max, Maya, MotionBuilder), Adobe Production Premium (After Effects, Premiere, Photoshop), and Unity 3D.

Figure 26: Artanim's User Lab.

11.2.CERTH Lab Node

CERTH has two available rooms (studios) for the user lab, one in Building A of dimensions 4.5m x 4.5m x 2.5m, and one in Building B (see Figure *27*) of dimensions 5m x 5m x 4m. The laboratories are equipped with RGB-D, Motion Capture and VR/AR equipment. In particular:

- Motion capture
- XSens MVN 9 MTx inertial trackers - motion capture suit
- RGB-D cameras for skeleton tracking - 6x Kinect v2, 6 Kinect v1
- Other 3D cameras
- 1x ZED Stereo Camera
- AR/VR HMD
- 1x HTC Vive
- 1x Microsoft HoloLens
- 3x Drones (4K) (to be purchased)

CERTH's software includes MS Visual Studio, Unity 3D, and Photogrammetry Software (to be purchased).

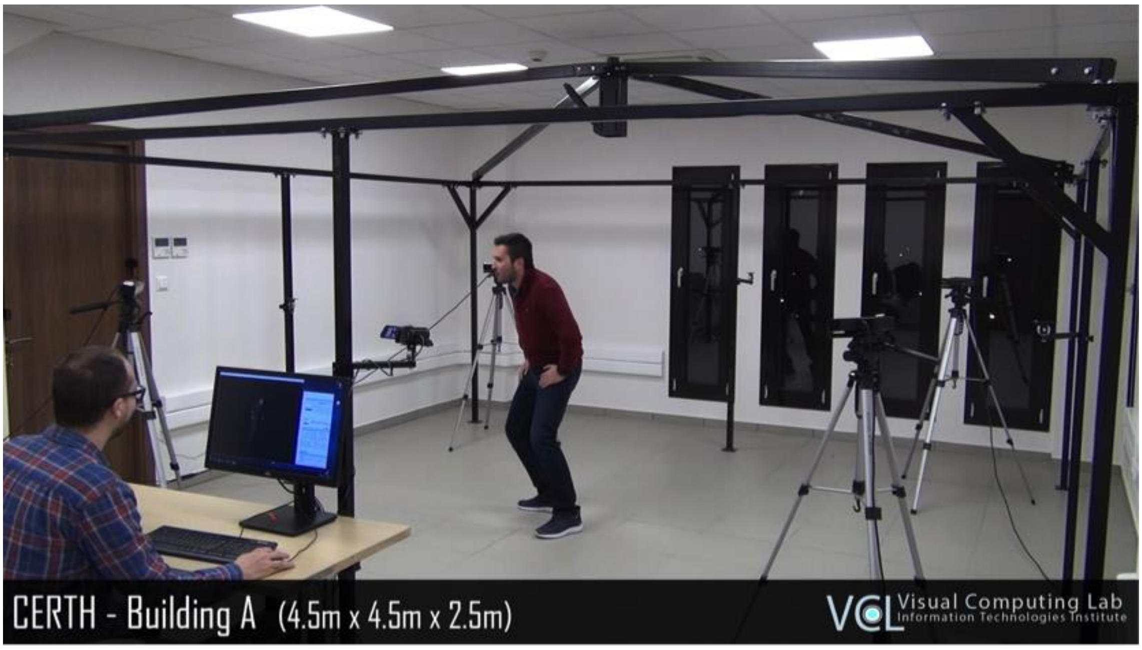


Figure 27: CERTH's User Lab

11.3.CWI Lab node

CWI has two available rooms: Pampus (see Figure 28) and the QoE Lab (see Figure 29). Pampus is a living room like lab, where experiments about user experience can be performed. It includes two sofas, a television, cameras, and a microphone array. The room has as well an interactive table that we don't expect to use during the project. The QoE Lab, under construction, will eventually become a hub for the project. It has been used to run experimentations for MPEG call for proposals in point clouds, and includes accessories, a top quality 55'' TV set (LG OLED 55C7V) and capture and rendering equipment (to be purchased).

Figure 28: CWI's User Lab (Pampus)

Figure 29: CWI's User Lab (QoE Lab)

11.4.TNO Lab node

At TNO premises, we have a media lab of approximately 8mx12m, as well as regular meeting rooms which we can reserve for whole days to run user tests. None of these rooms allow for the setup of a dedicated and (semi-)permanent user lab. The aim is to develop and release a virtual user lab (i.e., a software platform) that can be setup at physical locations for user tests. TNO has equipment for a social VR setup of up to four persons:

- Two VR capable PC systems and three VR capable laptops;
- Four Oculus Rift VR HMDs, including two sets of touch controllers;
- Four Microsoft Kinect RGB+D cameras for user capture;
- Four general-purpose headphones and microphones.

Figure 30: TNO's Media Lab

11.5.i2CAT Lab Node

The Lab node of i2Cat in Barcelona will be composed of:

Space Setup

- A place allowing to run multi-user VR experiences.
- 2 separated rooms that allows two users, each in a minimum clean space of 5x5x2,5m (space for cameras and PCs is extra)
- Furniture (coach, table, chairs, etc.)

Hardware for consumer setup:

- Render:2 VR-ready pcs + 2 HTC Vive + 2 headphones + microphones
- Motion trackers
- Communication: LAN connections
- Content streaming: 1 server PC to test services locally
- Mesh capture: 2 Capture rigs based on RGBD cameras (each involves 1 cam for face, 4 Kinect for Xbox one for the body reconstruction, 5 pcs + CERTH software)
- 4 Intel RealSense cameras

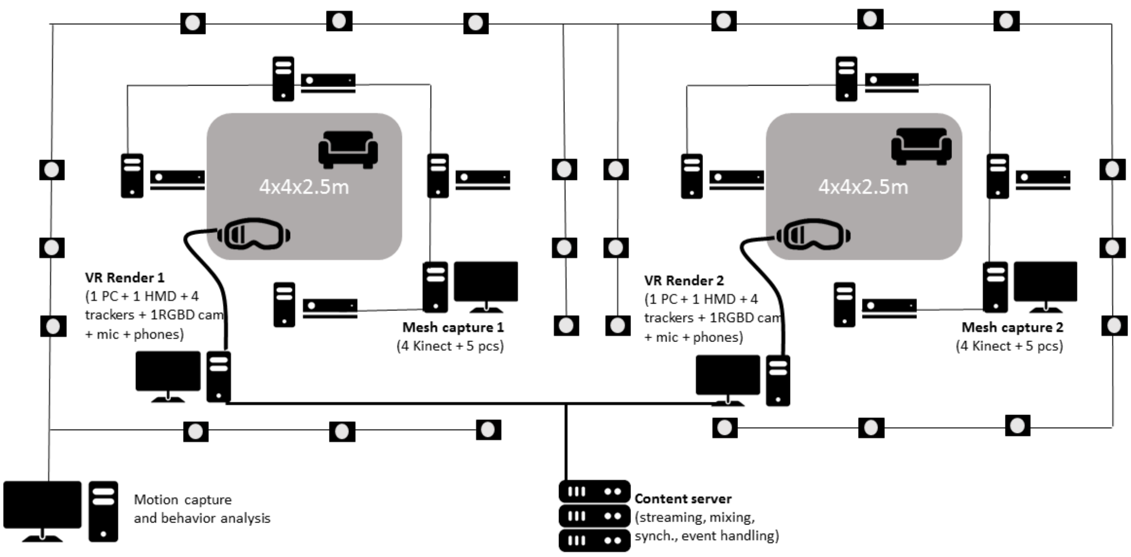


Figure 31: i2CAt lab infrastructure

In conclusion the partners of the projects have adequate facilities for testing and experimentation. The initial six months of the project will be dedicated to one the one hand run some initial experiments in the user labs for gathering requirements and to on the other hand construct the hubs for VR-Together.