Ref. Ares(2024)1122664 - 14/02/2024

CORTEX²

D2.1 – Collaborative environment models for extended reality

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement nº 101070192. This document reflects only the author's view, and the EU Commission is not responsible for any use that may be made of the information it contains.

Project Title	COoperative Real-Time EXperiences with EXtended reality
Project Acronym	CORTEX²
Grant Agreement No	101070192
Instrument	HORIZON Innovation Actions
Topic	HORIZON-CL4-2021-HUMAN-01-25
Start Date of Project	September 1, 2022
Duration of Project	36 months

Name of the Deliverable	Collaborative environment models for extended reality
Number of the Deliverable	D2.1
Related WP Number and Name	WP2 - Multimodal collaborative space
Related Task Number and Name	T2.1 - Abstract multi-view collaborative space model T2.2 - AR/VR Behaviour and intention modelling
Deliverable Dissemination Level	PU
Deliverable Due Date	M12

Deliverable Submission Date	31.08.2023
Task Leader/Main Author	LINAGORA
Contributing Partners	ALE, DFKI, UJI, KUL, ICOM, ACT, CEA
Reviewer(s)	UJI, DFKI

Keywords

CORTEX² Application, Virtual Cooperation, Virtual Collaboration, UML model, Specification

Revisions

Version	Submission date	Comments	Author
v0.1	2023/05/20	Document structure	Yazid Benazzouz & Jean Pierre Lorré
V0.2	2023/07/19	Section 1,2,3	Rivier Sylvain, Helbert Emmanuel, Szymendera Lydia, Grzelak Olivier, Benazzouz Yazid, Jean Pierre Lorré
V0.3	2023/07/24	Section 4, 5, editing	Pagani Alain, Maria-Evgenia Xezonaki, Minaskan Narek, Benazzouz Yazid, Jean Pierre Lorré

V0.4	2023/08/10	Internal review version	Pagani Alain, Cristina Gasch Garcia, Emmanuel Helbert
V1.0	2023/08/31	Final version	Pagani Alain, Yazid Benazzouz & Jean Pierre Lorré
V2.0	2024/02/02	Revised version after review R1. Added scenarios of use in the novel sections 3.1.8, 3.2.8 and 3.3.8, with additional tables Table 1, Table 2 and Table 3	Pagani Alain, Yazid Benazzouz & Jean Pierre Lorré, Rivier Sylvain, Helbert Emmanuel, Szymendera Lydia, Grzelak Olivier, Maria-Evgenia Xezonaki, Minaskan Narek, Cristina Gasch Garcia, Gael de Chalendar

Disclaimer

Acronyms and definitions

Acronym	Meaning
AR	Augmented Reality
APP	Application
FSTP	Financial Support to Third Party
IIoT	Industrial Internet of Things
IoT	Internet of Things
MDE	Model-Driven Engineering
MVP	Minimal Viable Product
OCL	Object Constraint Language
UML	Unified Modeling Language
VCAA	Video Compression and Alternative Appearance
VR	Virtual Reality
XR/MR	Extended/Mixed Reality

Abstract

This deliverable outlines the boundaries of a CORTEX² application and differentiates it from existing virtual applications such as those in the domains of gaming and entertainment. The specificity of CORTEX² applications is their focus on cooperation and collaboration. This leads to the use of functionalities adapted to this use cases, which we classify under the name of CORTEX² applications. To this aim, we have developed an abstract UML model, as well as concepts and knowledge about behaviors and intentions that are suited to the development and use of CORTEX² application. After presenting the developed model of application, behaviors and intentions, we validate the model across the planned pilots and verify the applicability of the models. Finally, we stress societal and ethical factors that are important for CORTEX² applications.

The CORTEX² project

The COVID-19 pandemic pushed individuals and companies worldwide to work primarily from home or change their work model to stay in business. Today, all the signs are that remote work is here to stay. But not all organizations are ready to adapt to this new reality, where team collaboration is vital.

Existing services and applications aimed at facilitating remote team collaboration — from video conferencing systems to project management platforms — are not yet ready to efficiently and effectively support all types of activities. And extended reality (XR)-based tools, which can enhance remote collaboration and communication, present significant challenges for most businesses.

The mission of CORTEX² is to democratize access to the remote collaboration offered by next-generation XR experiences across a wide range of industries and SMEs.

To this aim, CORTEX² will provide the following:

- Full support for **augmented reality (AR) experiences** as an extension of video conferencing systems when using heterogeneous service and devices through a novel Mediation Gateway platform.
- Resource-efficient **teleconferencing tools** through innovative transmission methods and automatic summarization of shared long documents.
- Easy-to-use and powerful **XR experiences** with instant 3D reconstruction of environments and objects, and simplified use of natural gestures in collaborative meetings.
- Fusion of vision and audio for multichannel semantic interpretation and enhanced tools such as virtual conversational agents and automatic meeting summarization.

- Full integration of internet of things (IoT) devices into XR experiences to optimize interaction with running systems and processes.
- **Optimal extension possibilities and broad adoption** by delivering the core system with **open APIs** and launching **open calls** to enable further technical extensions, more comprehensive use cases, and deeper evaluation and assessment.

Overall, we will invest a total of 4 million Euros in two open calls, which will be aimed at

1. recruiting tech start-ups/SMEs to co-develop CORTEX²
2. engaging new use cases from different domains to demonstrate CORTEX² replication through specific integration paths
3. assessing and validating the social impact associated with XR technology adoption in internal and external use cases

The CORTEX² consortium is formed by 10 organizations in 7 countries, which will work together for 36 months.

Contents

1. Introduction

This deliverable summarizes the work on tasks T2.1 "**Abstract multi-view collaborative space model**" and T2.2 "**AR/VR Behaviour and intention modelling**". Its objective is to establish a coherent specification that will characterize a **CORTEX² application**, standardize its functioning and limit its usage domain. In this way, developers using the CORTEX² framework will be able to apply it to the collaboration and cooperation scenarios for which it has been implemented.

In reference to the high-level architecture of the platform in the deliverable D5.1 "**CORTEX² requirements and deployment architecture"**, a CORTEX² application represents the client part. The user carries out all interactions in the XR world through the CORTEX² application

1.1. Methodology

Tasks 2.1 and 2.2 are dedicated to the Abstract multi-view collaborative space model and the modelling of the AR/VR behaviour and intention. These two tasks ran in parallel with the task T5.1 "**U-X, requirements and Pilots**". Following the development of the user experience and of the requirements, the models developed in tasks T2.1 and T2.2 adapted to the results of task T5.1. For this reason, the consortium did not opt for the possibility of developing an authoring tool that generates applications automatically. Rather, the CORTEX² implementation is defined as a framework, specified in deliverable D5.1, open to the design of specific collaborative and cooperative applications involving XR and non-XR software's.

Therefore, the tasks T2.1 and T2.2 focused on the modelling and specification aspects, by defining concepts and rules appropriate to the development of CORTEX² applications. Presentation of the models, discussion among the partners and decisions about the form of the model were conducted during dedicated online meetings. These meetings were held regularly, either weekly or bi-weekly. The last months of the task duration were dedicated to the finalization of the models and the generation of the documentation.

1.2. Main achievement

The work carried out on Tasks 2.1 and 2.2 has resulted in an abstract graphical model in the form of entities and associations based on the UML notation. It describes both the static and dynamic elements of the virtual scenes, as well as the behavioural impact of objects and users on the scenes' changes. The model intends to frame any CORTEX² applications, focusing on cooperation and collaboration, unlike virtual reality gaming or online VR tour applications for museum and real estate agencies. The representation of the model in an abstract way enables to focus on the most important elements of the application, for greater clarity and better implementation. The advantages lie in the way the application model is constructed and the notations used. The model is primarily intended to the development of the pilots, as a guideline for their developments and for the identification of generic functionalities. Moreover, the models will be used beyond the pilot developments with a focus on the features of the CORTEX² framework to enable re-usability and compliance to the specification, targeting third parties to help them producing their own applications. The benefits of this abstraction include improving the productivity of development teams, helping to understand how CORTEX² application works, and enabling the evolution and maintenance of developments.

The evolution of this specification towards a standard could help regulating XR applications development centred around collaboration and cooperation and facilitate the setting of limits to data usage to comply with European Union rules.

1.3. Future work

The instantiation of this abstract model and the enforcement of the specification by the pilots will be the first step towards the exploitation and the validation of this work. We also expect the recommendations in this document to be applied by external partners, such as the third-party partners who will join the project under the FSTP calls, which could provide a critical look that would be beneficial for the project., We will also be looking for opportunities to launch a community around XR collaborative and cooperative applications. Some important elements in CORTEX² applications need to be regulated and may require specification at a European scale, such as rules for the use of certain objects in VR mode, communication from a VR environment, etc., which may have a psychological, social, ethical and privacy impact. In the long term, it could be interesting to develop a software tool that help instantiating an application model from the abstract one for rapid prototyping and building applications.

1.4. Structure of the document

The remainder of the document is organized as follows: section 2 deals with the abstract muti-view collaborative space model. In section 3, the model is applied to the use cases carried by the pilots. Section 4 describes external services requirements to be compliant with the model. Finally, section 5 analyses to what extent the model satisfies ethical, legal and social requirements

2. Abstract multi-view collaborative space model

This section introduces the abstract multi-view collaborative space model and defines the relevant terms to CORTEX² applications development. The abstract model defines the CORTEX² domain of applications as the domain of collaboration and cooperation environments using XR technologies. In these environments, users play different roles depending on their involvement, and use different devices to cooperate with other participants. Moreover, additional features can be added to facilitate collaboration, such as 3D object scanning, augmented reality annotations and automatic notetaking. In addition to voice and video communication, there is a highlighting of the subjects discussed during cooperation through the manipulation of 3D objects and immersive interaction. Therefore, the collaborative environments in CORTEX² is different form the classical environments such as VR gaming, AR for user face augmentation etc. The question that we have to answer is: how can users use a collaborative application in such a way that the 3D scenes are adapted to their roles and to the execution environment?

The abstract model is designed to simplify the developers' understanding of the CORTEX² framework and the design of an application based on it. The idea is to ensure that the conceptual vision is reflected in the implementation, and to avoid misunderstandings by checking the coherence and consistency of the instance model produced from the abstract one. This abstraction aims to be generic enough to support any possible device, scene or object involved while leaving developers as much flexibility as possible. The abstract model is also intended to provide a development guide for CORTEX² applications development to increase reusability and encourage exploitation for other use cases.

In the following section we survey existing technologies for XR application development and their associated methodologies for developing an XR application.

2.1. An insight into XR application development

A look at how XR applications can be developed will enable to verify the existence of methodologies, packages or models for specific XR application domains. The idea is to see if it is possible to recover certain notions or create a unified view for cooperative applications from these knowledge assets.

2.1.1. WebXR

WebXR[1,2], with the WebXR Device API at its core, defines the elements needed to bring both augmented and virtual reality (AR and VR) to the web. Basic concepts include XR session, frame, spaces, views, geometric primitives, and many others. It is not a rendering technology, rather a group of standards to support the rendering of 3D scenes on hardware designed to present virtual worlds or to add graphics to the real world. WebXR-compatible devices include fully immersive 3D headsets with motion and orientation tracking, eyeglasses which overlay graphics atop the real-world scene passing through the frames, and handheld mobile phones which augment reality by capturing the world with a camera and augment that scene with synthesized images.

Most applications using WebXR will follow a similar overall design pattern: checking to see if the user's device and browser are both capable of presenting the XR experience, user activation of WebXR functionality and finally the management of WebXR session for the duration of the WebXR experience. This involves managing inputs, animations, and rendering. In most cases, **WebGL**[3] is used for rendering the 3D world into the WebXR session, but, one of the frameworks or libraries that are built atop WebGL will be more convenient to use. These frameworks are good for general-purpose application programming as well as for game development among them **A-Frame**[4] (specifically designed for creating WebXR-based apps), **Babylon.js**[5] and **Three.js**[6].

1 https://immersiveweb.dev/
2 https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API
3 https://get.webgl.org/
4 https://aframe.io/
5 https://www.babylonjs.com/
6 https://threejs.org/

2.1.2. Commercial game engines

A game engine is a software framework or platform that provides developers with a set of tools, libraries, and features to create, design, develop, and run video games. Game engines are essential for game development as they streamline the process, saving developers significant time and effort in creating the fundamental components of a game, such as graphics rendering, physics simulation, sound, and more. Game engines often come with a built-in editor, scripting or programming capabilities, and support for various platforms (PC, consoles, mobile devices, etc.). Games engines can be used to generate a XR application, either directly for engines supporting AR/VR, or indirectly by using the engine capabilities together with external libraries for the registration. The following list describe the most important game engines:

- **Unity**[7]: Unity is one of the most popular and widely used game engines in the industry. It's known for its versatility, supporting 2D and 3D game development across multiple platforms, including PC, consoles, mobile, and AR/VR.
- **Unreal Engine**[8]: Developed by Epic Games, Unreal Engine is another highly regarded game engine known for its stunning graphical capabilities. It's commonly used for creating AAA games, simulations, architectural visualization, and more.
- **CryEngine**[9]: CryEngine is known for its advanced rendering capabilities and is often used for creating visually impressive games. It's used in titles like the Crysis series and Star Citizen.
- **Godot Engine**[10]: Godot is an open-source game engine that has gained popularity for its user-friendly interface and its ability to export games to various platforms. It's particularly well-suited for indie game developers.
- **O3DE**[11]: Open 3D Engine is a free and open-source 3D game engine developed by Open 3D Foundation, a subsidiary of the Linux Foundation, and distributed under the Apache 2.0 open-source license. The initial version of the engine is an updated version of Amazon Lumberyard, contributed by Amazon Games.

[7] https://unity.com/
[8] https://www.unrealengine.com/
[9] https://www.cryengine.com/
[10] https://godotengine.org/
[11] https://o3de.org/

- **GameMaker Studio**[12]: GameMaker Studio is a game engine tailored for 2D game development. It's popular among indie game developers and has a user-friendly drag-and-drop interface.
- **Phaser**[13]: Phaser is a free and open-source framework for creating 2D games using HTML5 and JavaScript. It's primarily used for web-based games.

In order to ensure compliance with recent AR and VR hardware, the consortium made the choice to use Unity as one of the solutions for the development of CORTEX² applications.

Unity is a Cross-platform game engine. It allows to develop within an integrated development environment as whereas games or industrial application on a variety of desktop, mobile, console and virtual reality platforms such as XR or AR headsets. Unity is one of the most popular development tools for such cases. In the context of CORTEX², the capability to address most of all Virtual Reality device with an easy developer adoption made us select Unity for the project. Unity will help to develop all client's component, from the 3D environment definition to the media integration and interactions. Unity is not open source and may be a barrier to some developers, but its XR capabilities are unavoidable and may be a motivation for more advanced Cortex-based development. Therefore, to support Unity environment, a dedicated Rainbow™ SDK for Unity package will be developed over the existing Rainbow™ C# SDK.

2.1.3. Graphical models for the XR

The construction of XR applications involves, to a large extent, the graphical design of the immersive environment. Thus, a large part of application development implies scene representation, objects, animations, avatars and their characters to form the virtual world. In some cases, development of XR application is then reduced to loading and combining these elements to fine-tune the way in which the user evolves to the space provided by the application. This has led to several file format standards for the use and exchange of 3D graphic models. These include:

[12] https://gamemaker.io/
[13] https://phaser.io/

- glTF™ (Graphics Language Transmission Format)[14] is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by engines and applications.
- FBX® (Filmbox) data exchange technology[15] is a 3D asset exchange format that facilitates higher-fidelity data exchange for 3ds Max.
- OBJ files[16] are used by Wavefront's Advanced Visualizer application to define and store the geometric objects. Backward and forward transmission of geometric data is made possible through OBJ files. Both polygonal geometry like points, lines, texture vertices, faces and free-form geometry (curves and surfaces) are supported by OBJ format. This format does not support animation or information related to light and position of scenes.
- Universal Scene Description (USD)[17] is an open, extensible ecosystem originally invented by Pixar Animation Studios. It enables the description, composition, simulation and collaboration of 3D worlds. Thanks to its design and functionalities, USD is poised to become the open standard enabling the 3D evolution of the Internet: the metaverse.
- "VRM" (Virtual Reality Model) [18,] is a file format for handling 3D humanoid avatar (3D model) data for VR applications. It is based on glTF2.0.

2.1.4. Helios-H2020

The European project HELIOS[19] raised the challenge of the democratization of VR development, i.e., providing people with access to technical expertise (application development) via a radically simplified experience and without requiring extensive and costly training. No-code models are examples of democratization. Their analysis led to three ways to create immersive applications.

1- Traditional method involves negotiation with a studio, namely with application designers and artists, known as user requirements analysis and definition, which in turn must provide input to programmers which are responsible for writing the source code of the

[14] https://www.khronos.org/gltf/
[15] https://www.autodesk.com/products/fbx/overview
[16] https://docs.fileformat.com/3d/obj/
[17] https://www.nvidia.com/fr-fr/omniverse/usd/
[18] https://vrm.dev/en/
[19] https://helios-h2020.eu/wp-content/uploads/2022/01/D5.2_3D-authoring-tools-for-V-space-creation.pdf

application in C# for Unity[20] graphics engine. Alternative graphics engines are Unreal and Godot. The costs for simple applications are significant, however, the quality and portability achieved is very high due to the ability of Unity to exploit open source and in-house made compilers.

2- In this second approach, the design is achieved through an authoring interface that abstracts the most important functionalities of a full-fledged authoring environment. The output is only available for web browsers, a WebGL JavaScript based application. The WebGL experience is of low quality with respect to the binary's outputs of Unity graphics engine.

3- The third method is achieved by adding an extra step to web editors, named, transpiling, which is the missing link between Web Browsers (WebGL) and Unity (C#).

2.1.5. Conclusion

The major trends in mainstream XR development are based on WebGL and Unity. The development methodologies were described by the European Project Helios. The first approach is very expensive, and the two latter are very complex and only feasible for a limited number of applications, especially non-shared ones. It is also obvious to involve designers for the creation of the 3D scene graphics, or just to elaborate the interface of the XR application. In light of these approaches, CORTEX² project aims to provide an open framework for collaborative and cooperative applications that can involve several personalized scenes per user role. The emphasis is on functionality, not on automating the creation process. In this sense, an abstract model with sufficient specification would be more appropriate for developers, both to give them flexibility in creating applications and to allow the framework to be extended.

2.2. CORTEX² Application

A CORTEX² application is a virtual, augmented, mixed or extended reality application that allows remote people to work cooperatively or collaboratively to achieve a goal. The key is to have a design model for such applications and not covering all kinds of immersive applications. Having

[20] https://unity.com/

this focus will provide a solid foundation for reaching a promising market increasingly interested in remote collaboration and cooperation use cases.

There is a need to identify the characteristics and specific features of this category of applications to build a model that makes the development of an application in this category easier or even governed by strict specification. Its initial exploitation will be undertaken by the pilots to assist them in the application development process, and as a testing exercise.

2.2.1. Type of CORTEX² applications

The aim of this section is to provide a general functional description for CORTEX² applications. The requirements analysis of D5.1 "**CORTEX² requirements and deployment architecture**" and the exchanges between project partners led to identify 3 types of immersive applications represented by Figure 1, from which we have identified the one that best match the needs of this project and are compatible with the framework already established in D5.1.

1. **non-shared application**: the application is hosted locally or remotely, and the user joins the immersive world and interacts with it without sharing its presence with other users as if you were launching a virtual drawing application on Meta Quest device.
2. **shared application**: in this case, the same scene in the virtual world is shared by several users, as in video games.
3. **sharing through different applications**: users in this situation use different mixed or virtual reality applications but are able to share their presence and to work together.

In the last two types, access to applications may be restricted to particular devices due to compatibility issues. Furthermore, some users may use the Rainbow™ videoconference application[21] (Rainbow™ is an application from partner Alcatel that is used as a backbone of the CORTEX² framework - see deliverable D5.1) to join a shared virtual session using, for example, a laptop or a smartphone. The aim is to promote interaction between users who do not have access to XR technology, and to overcome the lack of democratization of this new technology. It's becoming clear that type 3 is the one that best reflects our ambitions in this project.

[21] https://www.openrainbow.com

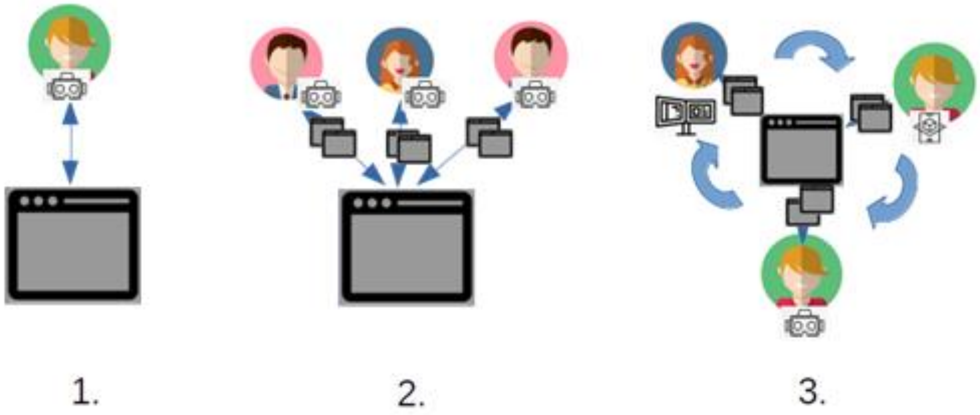

Figure 1: Categories of CORTEX² applications

2.2.2. CORTEX² Application Abstract Model

Graphical modelling languages are the preferred choice for many designers when it comes to defining the structural aspects of a domain i.e., its main concepts, their properties and the relationships between them. The most typical example of a graphical notation is Unified Modelling Language (UML), especially its class diagram which is by far the most used UML diagram. It was expanded by Object Constraint Language (OCL)[22], a constraint language for UML but quickly expanded its scope and now OCL has become a key component of any model-driven engineering (MDE) and will be used for our abstract model.

The model of Figure 2 embodies the idea of a shared virtual world for collaboration and cooperation. The world is made up of one or more scenes running on one or more devices, in compliance with the constraints of code compatibility. A scene is a central element of the application. It is made up of graphical objects. It has unique shape, and each object is associated with a set of behaviours: for example, a door can be opened or closed, but not moved within the scene. The avatar is considered as an object and also has defined behaviours in the world. Status changes only occur when an action is performed by a participant, i.e., the real user or a virtual agent such as an assistant, which can have autonomous, intelligent actions in the world. Objects can also be influenced by external actions, such as the status change of an IoT object connected to the virtual world.

[22] https://modeling-languages.com/ocl-tutorialObject Constraint Language (OCL), Jordi Cabot | Mar 21, 2020

An actor (participant, organiser, or virtual agent) has the choice of joining a conference from Rainbow™'s classic platform or from the CORTEX² application. The CORTEX² application runs on a virtual or augmented reality device. Similarly, the Rainbow™ application can be run from a PC, smart phone or tablet. This is expressed by an OCL constraint in model of Figure 2. A constraint represents some condition, restriction or assertion related to some element. The constraint is shown as a dashed line between the elements labelled by the constraint string in curly braces. The model also includes some instances in blue with underlined titles to help understand better some characteristics of a CORTEX² application.

The UML model is complemented by an ontology, described in Figure 3. An ontology makes it possible to specify an IT application using a formal model. It is essentially defined by a set of concepts and relationships between these concepts, which structure the knowledge domain underlying the use cases targeted by the application. The fact that it is formal means it can be verified, unlike other models. This approach looks promising for achieving our ambition of bringing together a large community around a single standard of applications focused on XR-enabled collaboration and cooperation.

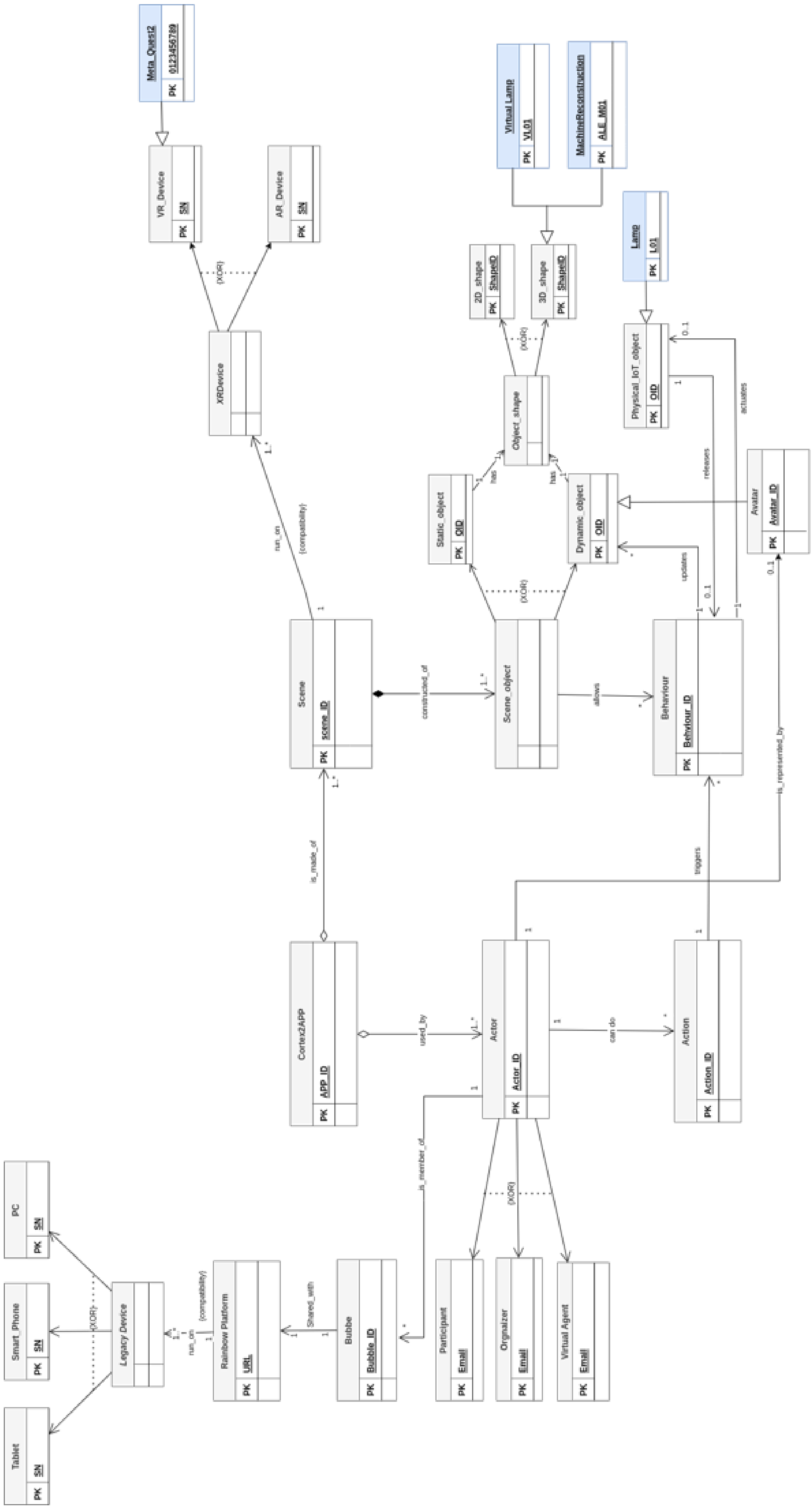

Figure 2: CORTEX² application model

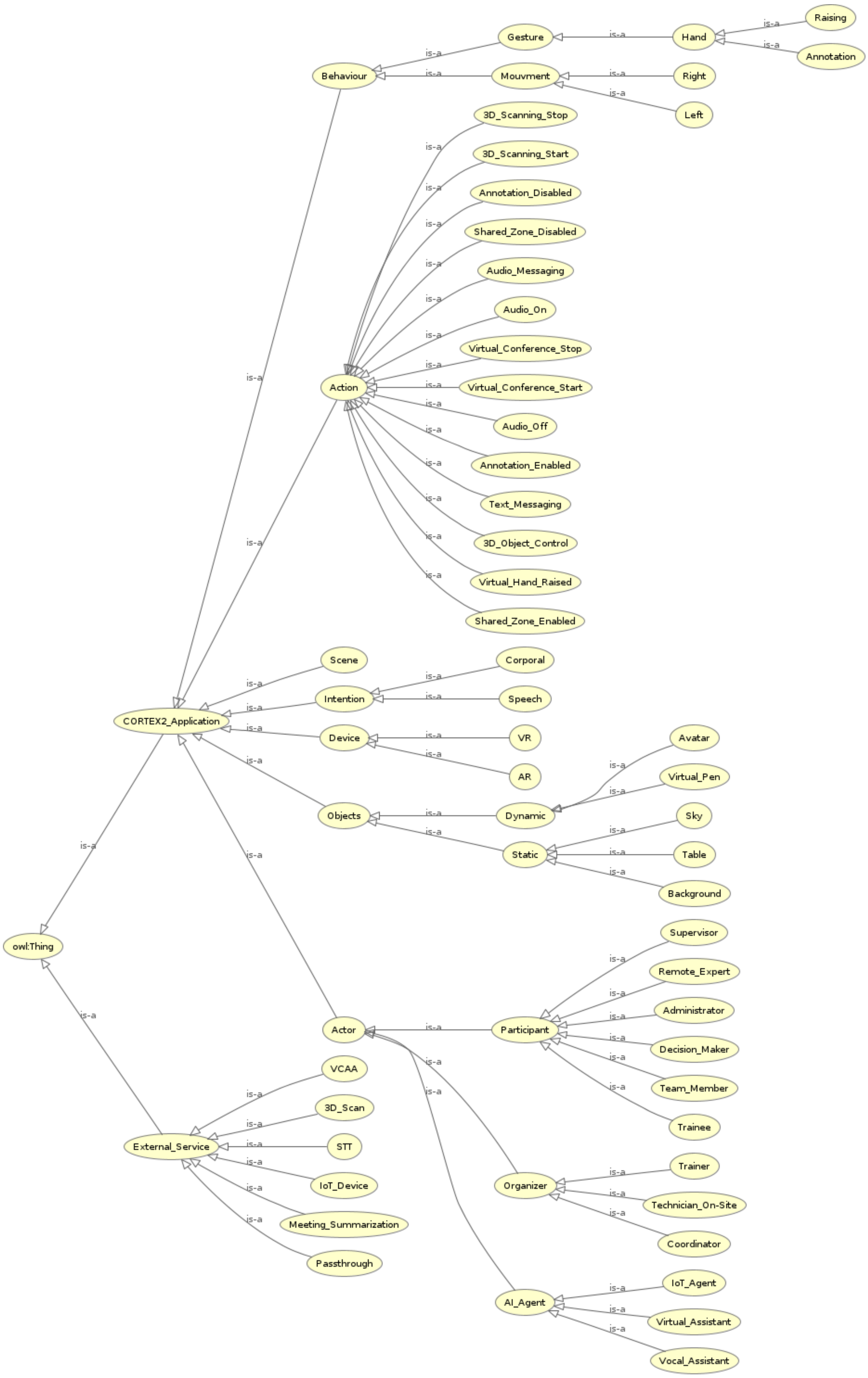

Figure 3: CORTEX² application Knowledge ontology

2.2.3. CORTEX² Application Guideline

The following subsections will highlight each step of the CORTEX² application guideline from a developer point of view. The CORTEX² application guideline is a manual containing a specification adapted to the concepts and nature of the applications developed within the CORTEX² project framework. It can be used as a collection of knowledge and rules to be followed by developers. The application guideline will be followed for the development of the pilots during the Tasks 5.3, 5.4 and 5.5. The benefit of this manual is twofold. The first is to ensure the software quality of the applications developed, i.e., their ability to satisfy user needs in terms of cooperation, and to evolve in line with collaborative requirements. The second is to help developers move faster in producing applications compatible with the objectives of this project. This work is also a first step towards defining new standards for use, security and, communication through new protocols adapted to these virtual environments. The following subsections highlight each step of the CORTEX² application guideline from a developer point of view.

2.2.3.1 *Application*

A CORTEX² Application is a software implementing a cooperative and/or collaborative virtual or augmented world to achieve a business goal through CORTEX² scenes, designed to show a virtual reality or augmented reality environment. The CORTEX² framework, cf. Figure 4, allows the appearance of the user in the virtual world with different forms: real face or alternative appearance if he is using Rainbow™ videoconferencing application, as an avatar if he is using a virtual headset. This is made possible by Rainbow™ core, where all communications are routed.

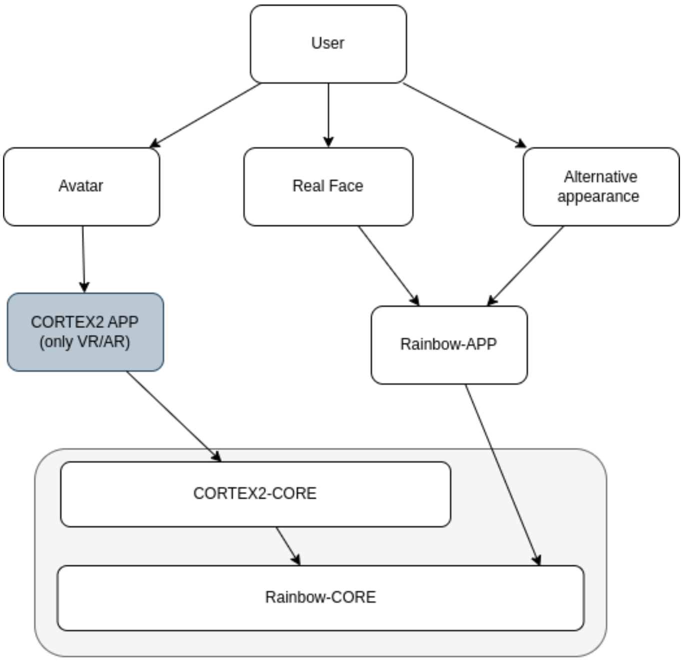

Figure 4: User login options

The management of users, roles and workspaces is a major task in the development of CORTEX² applications. Considering CORTEX² as a framework, the aim is to give the user the possibility of creating his own interface using the Rainbow™ SDK and to benefit from the Rainbow™ platform to accomplish this task in a very short time, allowing him to quickly carry out the necessary tests before moving on to a final product.

2.2.3.2 *Registration and Authentication*

User registration and authentication in CORTEX² applications is achieved through Rainbow™ SDK which is the central element for any data exchange including audio, video and meta data when using CORTEX² core (cf. D5.1, **CORTEX² requirements and deployment architecture**). In fact, user management is common to both CORTEX² and Rainbow™ core platforms to enable interoperability between virtual and classic videoconferencing. For rapid prototyping it is possible to skip this step and create users directly in Rainbow.

2.2.3.3 *Role-Based Access Control*

User Role management in CORTEX² application is controlled by CORTEX² Rainbow™ SDK[23]. It is made available and considered as the first means to handle user role. Developers have the

[23] https://developers.openrainbow.com/

choice of offering this service via Rainbow™ platform or developing a simple interface on top of the provided API which supports Unity and JavaScript.

2.2.3.4 *Scene*

CORTEX² scene is a user interface allowing immersion into a virtual world. It is made up of graphic objects that evolve in a space that integrates the physical world (AR) or designed to be entirely virtual (VR).

2.2.3.5 *Bubble*

Bubble is a team workspace where members can be all together in the same video conference. It is independent of the virtual scene, enabling users of the same bubble to be in video conference even if they are using different CORTEX² applications or classical Rainbow™ video conference platform. This makes the third type of CORTEX² application, (cf. Figure 1) possible. However, bubble management and member authorizations are handled by Rainbow™ SDK. Developers can manage this from their interface using Rainbow™ SDK. However, this step can be performed quickly within Rainbow™ platform to focus on prototyping.

2.2.3.6 *Actor*

A CORTEX² actor is a character of the CORTEX² scene. It can be a human user or an artificial agent acting autonomously or in reaction to specific events in the scene. An actor plays a specific role in the CORTEX² application virtual world.

2.2.3.7 *Role*

A role represents the position assigned to the actor. It restricts or extends the actor's ability to act in the scene, i.e., the actions he can or cannot perform. The are three principal roles: participant, organizer, and artificial agent. To each one can be associated a true responsibility, for example, one can participate as an operator or as an expert; a virtual agent could be an assistant or an IoT agent, etc.

2.2.3.8 *Action*

Actors can perform actions in the virtual or extended world according to their associated roles. These actions are those that enable collaborative or cooperative work. They are governed by the ontology presented in Figure 3. For example, the video sharing action is not mentioned, as we consider that users in a virtual world don't need video to see each other and are substituted by avatars. In addition, we've added the voice-messaging action to allow easier messaging from the virtual scene.

2.2.3.9 *Scene objects*

Scene object is a 2D or 3D graphical representation that belongs to the scene or to a real object such as an annotation or a sign. We distinguish between static objects, which cannot be manipulated or changed in the virtual environment, and dynamic elements, which can be manipulated by an actor's action. The ontology shown in Figure 3 provides a simplified classification for the Business Meeting pilot, which will be described according to this guide in Section 0.

2.2.3.10 *Avatar*

The avatar is a scene dynamic object. It's a visual representation of the user, with or without a likeness, or a physical form of an AI agent. It can take the form of a complete body, or just the face and hands. The avatar's behaviour imitate the user and it is not predefined for CORTEX² application. It must be implemented depending on the CORTEX² application use case.

2.2.3.11 *IoT Virtual Object*

An IoT virtual object is a scene dynamic object which represents a real-world physical IoT object that is linked to CORTEX² system. Once a real IoT object (e.g., a sensor and/or actuator) is registered to CORTEX², a virtual (i.e., graphical – 2D/3D) representation of the real object is created for the CORTEX² application with which the object is associated and is added to the virtual environment seen by the authorized participants of this application. The virtualization of the IoT object includes the virtual (graphical) representation of the real object's measurements values in the case of a sensor or the current state in the case of an actuator, which are always synchronized among all the participants. Each participant, depending on their permissions, can interact with the IoT virtual objects (if applicable, e.g., in the case of actuators) through their hand gestures/body movement or speech recognition, which are translated by CORTEX² in an IoT-compatible format as actuation commands. This depends on the action types (e.g., gesture movement, speech commands etc.) supported by the actor’s equipment capabilities and the behaviours supported by the specific IoT object (as described in Figure 2). By sharing the same IoT virtual objects, the participants can experience a more effective and immersive collaboration with each other and gain better understanding of the topic in discussion.

2.2.3.12 ***Device***

Two device categories can be identified within the scope of a Cortex² application, the IoT device (IoT object) and the participant/user device (end device).

An IoT device is a real IoT object which can be a sensor and/or an actuator. An IoT sensor will provide to CORTEX² framework useful information and measurements about a physical condition that needs to be monitored, such as temperature, humidity, air quality, dust, smoke etc., or even send alarms on severe issues e.g., to report industrial equipment failures. An IoT actuator will receive instructions from CORTEX² framework in order to execute an actor's action and activate different functionalities of the industrial equipment according to the behaviour triggered from the action. Such user actions could be for example pressing specific buttons of the industrial equipment to switch on/off the device or change dimming levels of a light. Both types of IoT devices (actuators and sensors) will be registered to the system through the relevant CORTEX² components and relevant permissions will be set on them to enable users' authorized access to them.

On the other hand, each participant of a Cortex application can use one or more end devices to connect to the virtual meeting. These devices can be any XR device, such as VR glasses and headset, or legacy devices such as laptops and smartphones. The user will be registered to CORTEX² framework and will need to log in with their credentials from any device that they want to use. Therefore, a user can participate to a remote meeting using any of their end devices.

2.2.3.13 ***Scene synchronization***

Scene synchronization consists in reporting changes to dynamic objects in other scenes of the application virtual world. The application must refer to data synchronization libraries provided by Rainbow.

2.2.3.14 ***External services integration***

The easiest way to integrate external services to CORTEX² application, is to carry it out through virtual agents. Virtual agents are not real users but could be members of the conference bubble allowing them to listen to the conference and to share information with the members. Other services can be entirely part of CORTEX² and can be used by any developer. Some of these services have been developed as part of the project and are intended to be used by pilots, for example:

- Alternative appearance:
- Speech to text
- Meeting summarization
- IoT object sensing and acting
- Scene reconstruction

Section 4 provides more details on how to implement and use external service. Moreover, developers should rely to CORTEX²CORTEX² architecture and Rainbow™ infrastructure for further technical documentation.

2.2.4. CORTEX² Application Behaviour and intention Execution Model

This section deals with the specification of actors' behaviours and intentions in a CORTEX² application. Their integration to the extended world is essential to deliver a better user experience. The emphasis is on those that are more suitable for collaborative and/or cooperative work, as opposed, for example, to the behaviours of an actor in a virtual museum.

In the virtual or augmented world, the possibilities of interaction are far greater than the usage of a simple laptop interface. Indeed, this is one of the motivations behind the use of extended reality. However, there are certain risks associated with these possibilities, which can lead to the user physical harm or to him surrounding, such as during a confusion between virtual and real machine interface when using augmented reality. That is why it is important to have a safety level for each of them. Knowing, that a regulation for safety usage at wider range of stakeholders will be of great help.

The CORTEX² application guideline suggests that the usage of behaviours and intentions follows rules forms that control the execution of resulting actions and events (cf. Figure 5). Rule-based execution models are easier to understand and provide more flexibility to add new, modify, or remove existing rules. The advantage for the virtual environment is that they are executed much faster than machine learning based approaches, considering performance and hardware requirements for virtual and augmented reality in distant cooperation and collaboration. However, machine learning approaches are more involved in determining actors' behaviours and intentions.

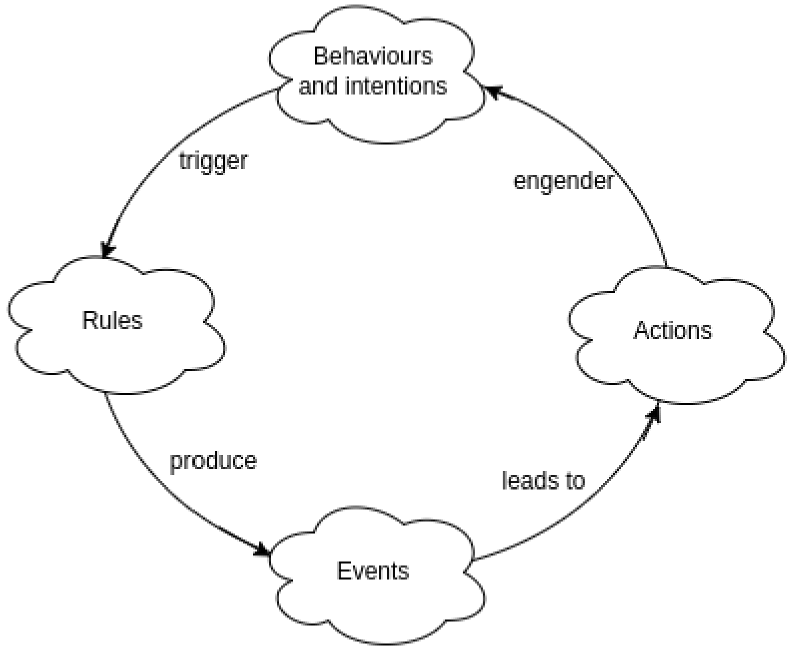

Figure 5: CORTEX² Behaviour and Intention model

The rules decision model provides a common notation easily understood by all developers who can automate decisions in any process with execution semantics easy to implement.

```
loop-for-ever {
        Sense ([ behaviour | intention ])
        React ([ action| event ])
}
```

Figure 6: Execution loop without synchronization

At the top, each rule is defined by a run procedure that consists of an infinite loop cycle that contains 2 procedures, cf. Figure 6. The first is to sense the behaviour or the intention and the second to react and action or trigger an event in response to the detected pattern.

In Shared CORTEX² application, the virtual or extended world is affected by the actions of all actors that have joined in that session. Since audio and video conferencing are already supported and provided by Rainbow™ API, no additional work has to be done at the developers' side. But for the virtual world scene, a Shared CORTEX² application uses all actors' interactions data in the rendering process so that they can all see the changes in the scene from their perspective. This is not a concern in a non-shared application or in case a sharing through multiple application (cf. Section 2.2.1), the rendered scene depends only on the actor. The rule model is then update accordingly, cf. Figure 7, as follows:

```
loop-for-ever {
        Sense ([ behaviour | intention ])
        React ([ action| event ])
        Synchronize (data)
}
```

Figure 7: Execution loop with synchronization

When an actor interacts with the virtual world or an IoT event is received, the information is transmitted into the Rainbow™ server to be processed and transmitted to CORTEX² applications. Because of possible message latency, inconsistencies will occur in the virtual and augmented scenes of some actors. If the server or the application lost the update message the state will not change until a recovery message is received. The necessary implemented strategies on the CORTEX² server side are very similar to that found in online video games.

2.2.5. Behaviour and intention Guideline

The guide aims to help application developers understand the behaviours and intentions that not only enhance users' virtual and augmented experience, but also help them find alternatives to existing features in standard video conferencing solutions, and adapt them to the virtual world, such as the hand-raising gesture.

2.2.5.1 *Visual behaviour*

A visual behaviour is a behaviour that can be visually observed by other users sharing the scene, and if identified by the application it can trigger a decision rule according to the model presented above. Changes and movements of objects in the virtual world scene are those of interest. This is why, in the UML model presented in Section 2.2.2, behaviours are associated with objects and not with users.

a. Avatar behaviours

There are two main patterns involved but others are left to the imagination of developers or project contributors via FSTP calls.

- **Eye gaze perception**: this is when the user changes his or her gaze to one direction in the scene, for example, to look at the presentation. It involves comparing the object's new coordinates in relation to the movement of the virtual headset, the user's head. This behaviour can be an indicator of a particular interest in something in the scene, for example, knowing whether users are interested in the presentation can help assess knowledge acquisition (in the case of an apprenticeship, training use case, or help recommend skipping certain parts because it seems that participants aren't interested in a particular point).
- **Hand raising gesture**: When the user is in immersive session, then the speech request gesture becomes visual via him avatar following a real hand-raising of the user using touch controller.

b. Object behaviour

Object behaviour concerns all the scene's dynamic objects except the avatars which has a particular processing. The behaviour can be a change of the shape, the colour or position. It's clear that objects can't change by themselves, but only as a result of user actions or events. For example, a different colour of a virtual IoT object may indicate an insufficient heating temperature level in the user's physical location, detecting the change in position of a stick on a machine during a scene reconstruction (using AR) can be an indicator of failure. Frequent behaviour

c. Frequent behaviour

Frequent behaviour corresponds to a particular habit of the user or his avatar regarding the arrangement of objects in the virtual world, e.g., orienting the camera to the sharing zone when connected (a sharing zone is where a presentation is displayed). This type of pattern can be detected and programmed to accommodate the user the next time. From CORTEX² application perspective, the aim is to have a modular construction of the code to allow such behaviour.

2.2.5.2 *Hidden behaviour*

This corresponds to measurable parameters that can be used to explain how the meeting went. For example, the duration of discussions can be used to manage the meeting's time by signalling overruns or abuse by a user. We can also imagine an evaluation of sentiment by detecting very high vocal frequencies with a very rapid change of speakers.

2.2.5.3 *Serial behaviour*

Successive user actions can be interpreted as behaviour that induces to a specific response. we distinguish two types:

a. Short-term serial behaviour

These are repetitive actions that run successively for a short period of time, for example, it is not suitable to use touch control to load a presentation; the user often uses three clicks to reach his goal. Given the time taken by this operation, a recommendation can be displayed to the user to indicate that it is possible to use speech to do so by citing the name and location of the file to be opened.

b. Long-term serial behaviour

These are distant actions in time, but a relationship exists between them. e.g., if a VR interrupting device, represented as an IoT object in this scene, goes on and off periodically, this may indicate a corporate emergency.

2.2.5.4 *Intent*

Unlike behaviours, the aim of detecting intentions is to try to understand the meaning of a gesture in a particular context. Two types are defined in CORTEX² applications. They are purely indicative. Their development can be complex depending on the use case. CORTEX² provides concrete examples that can be adapted to other implementations.

a. Verbal intent

They are the intentions deduced from speech recognition. For example, "we'd like to diagnose the PR2X machine" triggers a search for the machine's manual and its loading into the scene. In another context, this same sentence triggers the superimposition of the machine's technical image on the physical machine.

b. Corporal intent

Body intentions indicate a particular interpretation of body movement considering an interconnection between user corporal movement and him avatar. For example, if the movement of the avatar head goes left and right during the exchange between participants, this may mean a question rejection.

3. Instantiation of CORTEX² application model to pilots

In this section, the aim is to take the model and concepts of a CORTEX² application and instantiate variant versions depending on the use cases. This validation of the application model through the pilots allows us to generate more details about the developments to be carried out and how they will be performed. This validation will also enable to check that all pilots' requirements are addressed by the model, which may then be updated accordingly.

3.1. Maintenance Pilot

Innovative convergence of technologies like extended reality (XR), Industrial Internet of Things (IIoT), and artificial intelligence (AI) is reshaping industrial operations. A challenge that this pilot is keen to demonstrate through this maintenance application. The maintenance pilot proposes a simple way for a technician on site to have remote assistance by using heads-up and hands-free Mixed reality devices as illustrated by Figure 8.

In this case, Remote Maintenance CORTEX² application is of **type 3: "sharing through different applications"** (cf. Section 2.2.1). The users use different mixed or virtual reality applications but are able to share their presence and to work together.

In the Augmented Reality mode, participants will have the possibility to share their immediate surroundings through a simplified digitalisation process, which will result in a textured 3D model of their environments. This model will be used by distant users to identify, select and point to specific areas. In turn, these areas will be highlighted in the original users' view using Augmented Reality techniques (virtual arrows, virtual highlight).

In order to make the experience more immersive, rich contextual IoT information will be integrated into video streams, rendered as AR annotations on top of displayed objects and persons. To this end, data gathered from a number of heterogeneous IoT devices will be ingested, aggregated processed and prepared, ultimately generating layers of insightful information related to smart assets of various vertical domains. To do so, a versatile IoT Platform will be developed to collect data from connected devices and sensors and bring them into a unified, IoT-protocol-agnostic view that will allow the seamless management of IoT information and its custom “shaping” into layers of aggregated IoT information.

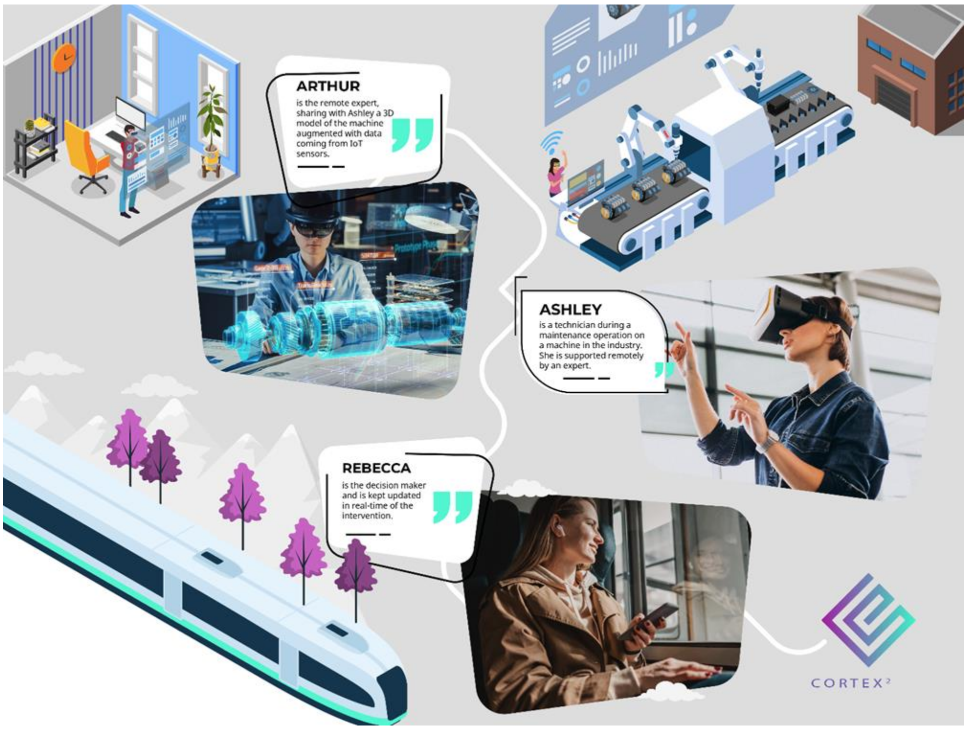

Figure 8: Pilot 1 illustration

3.1.1. Actors and roles

All the actors involved in the cooperative telepresence assistance are connected in audio conference and supported by vocal assistant for facilitating hands-free operations during some scenario tasks such as when the remote expert needs to load a specific machine manual from his repository. According to CORTEX² application guideline many participants with different sub-roles can be part of the maintenance session, listed below:

- **Technician on-site** receives assistance and can work hands free and heads-up. He gets the true scene view and shared information such as those related to IoT and annotated objects using AR. The technician could be the one requiring assistance with his machine and thus organizes a maintenance session.
- **The expert,** participating remotely, gets what the technician is currently seeing in real time. With the help of 3D scene reconstruction, he can see the surroundings of the intervention area and he will be able to push information, like 3D objects markers, machine blueprint or sharing IoT metrics, on the technician display.

- Additional stakeholders like decision makers can follow-up the maintenance operation but only video or screen sharing can be provided for others using legacy devices.

3.1.2. Scenes and objects

The Remote maintenance use case is not a classical multi-player Unity environment – the participants are not sharing the same “game” world. The remote expert and the technician on site have to share:

- Mesh data for complete 3D reconstruction, orientation and RGB data for panorama approach.
- Annotation data.
- Positions and orientations of 3D objects, object description.

We envision having a virtual pen to circle an area or some arrows to pinpoint a zone. We investigate also hand-tracking topic and the way for the remote expert to push the right-hand gesture in the technician display.

3.1.3. IoT objects

We will have to manipulate 3D model of the machine and a specific button on the machine will be linked to the right physical associated IoT sensors for real-time and historical values display. Depending to user roles, managed by Cortex core, we may allow either the technician or the remote expert or both to actuate IoT devices.

3.1.4. Intentions and behaviour

As we are targeting heads-up and hands-free experience, we will leverage voice command and hand-gesture recognition to make the interaction more intuitive. We may leverage the hand tracking capabilities of the device itself when available or interwork with an external service based on the video captured. We aim to integrate basic voice command, for instance to take a screenshot and have it integrated in the intervention summary or mute/unmute the microphone. Besides, eyes or gaze tracking are not integrated in the pilot. In future implementations, we may use this type of info accessible through the device SDK to have a contextual response of a gesture when the user is looking to a particular area. FSTP will propose applicants to explore those topics during the Open Calls.

3.1.5. Execution devices

The "behavioural model" is not totally independent from the actual device. So, the type of device that will be used (VR headset, AR headset, a hand-held tablet, a laptop, a PC) has an influence on the behaviour of the user. Actually, the capabilities of the model are strongly linked to what the device can do.

- Technician on site:
  - MR HMD device: tracking (6DoFs) - VR with Passthrough mode and capability to retrieve the captured video via SDK.
  - Tablet PC (DFKI has a demo with specific dll allowing the tracking based on the local video stream).
  - Possibility to use smartphone for basic AR use_case.

For the pilot we are focusing on Lynx-R1 device

- PC for the remote expert:
  - If hand tracking is required, we may use also a VR device or integrate some 3rd-party hand-tracking (like ultraleap).

Unity 3D framework is used for the bespoke technician and remote expert application as it provides the developer to support a wide range of device and its dedicated capabilities and SDK: Tracking (for the anchoring of virtual objects), Passthrough video capture or hand gesture and OpenXR support)

- PC or smartphone for the other stakeholder:
  - The stakeholder can use Rainbow™ standard UCaaS client to access to telepresence information: what the technician on site is currently seeing and what the remote expert may share.
  - Video alternative appearance can be pushed by the remote expert.

3.1.6. Reusable features

Remote Maintenance CORTEX² application has been developed to be reusable as a template in terms of behaviour and intention model and it demonstrates how to implement the below features:

- 3D scan of the scene based on the video capture performed by the technician on site: that allows a remote assistance based on a realistic display of the local scene and the capability to position AR elements at the right place.
- Vocal Assistant to recognize different intents and document search.
- VCAA: the pilot will develop the VCAA scenario restricted to the VCAA rendering in the HMD.
- IoT integration.
- XR in passthrough mode (device dependent) with AR annotations.
- Hand-gesture.
- Collaboration: conference audio/video/sharing.

A developer will be able to adapt the pilot sample application / MVP to fit with its particular maintenance process. We understood each use case will bring its working on its own 3D model.

3.1.7. Genericity of the application

The application requires to be customized to take into account new 3D industrial machines models, but the concept is totally replicable across different industries where remote maintenance is needed.

3.1.8. Scenario of use for the Industrial Maintenance pilot

This section details a “scenario of use” for the Industrial Maintenance pilot with a corresponding temporal sequence presented as step/action/result. In addition, at each step is associated the KPI being assessed.

The industrial maintenance use cases involves three participants: Ashley (A), a young maintenance technician, who is on the production site, Gilbert (G), an expert of the technical installation, who is not currently on the production site, and Rebecca (R), the site manager, who is currently travelling in a train. Ashley is called on the production site to repair a machine who suddenly stopped working. Since Ashley does not have the full required expertise to diagnose and repair the machine, she decides to use the CORTEX² solution to start a conference with Gilbert and Rebecca.

Ashley wears an Augmented Reality headset and is able to use the CORTEX² application directly form the headset. Gilbert is using a desktop computer with a classical monitor. Rebecca is using her smartphone.

The scenario of use depicts a situation in which Ashley is able to capture her environment and share it with Gilbert and Rebecca. Gilbert is then able to visualize the situation, and can interactively add virtual elements in Ashley's view to guide her towards the solution. Rebecca can follow the intervention by looking at a video stream from both other participants. In addition to interaction through audio, video and interactive augmented reality, the participants can connect IoT devices to the session, and can thus check sensor values or trigger actions through the IoT interface.

A full narrative of the scenario of use for the industrial maintenance use case if provided in *Section 2.1 "Remotely assisted AR Industrial Maintenance"* of the Deliverable D5.1 (p. 16). There, we describe the actors, the devices and the three scenarios. In addition, we provide in the present deliverable a complete scenario of use with individual steps to perform the scenarios, corresponding actions, expected results, and corresponding KPIs[24]. The following table shows the scenario of use of the industrial maintenance use case.

Table 1: Scenario of use for the Industrial Maintenance Pilot

Scenario IM1				
Ashley (A) is a technician on the field who receives assistance and can work hands free and head-up. A is on site and needs to repair an industrial machine. Since she does not have enough expertise, she calls Gilbert (G), an expert who is not on the same production site. A and G start a CORTEX² session and invite Rebecca (R) to join the conference. The environment of A is scanned and transferred to G – now A and G share the same reality.				
Nr	Step	Action	Result	Related KPIs
1	G launches the XR conference and logs in as a user	Start of the CORTEX app on the expert side and login	Conference started	Number of concurrent conferences KPI 5.20
2	A puts the HMD (Lynx R1) on her head	Launch of the application	CORTEX² app starts on the device	Number of scenarios /

[24] The list of KPIs if provided in deliverable D1.1, section 6.2

				devices supported KPI 5.2
3	A logs in as a user	Login /authentication using a QR code	User could log in the system (QR code recognized)	Number of participants KPI 5.21
4	A joins the collaborative space to connect with G and R	Selection of an active bubble through the CORTEX interface	A could join the bubble in the app	Number of participants KPI 5.21
5	G is able to see the live video stream as seen by A (“see what I see" mode)	Video stream from AR glasses transmitted to G side	Visualization of video stream without delay in acceptable quality	Delay for synchronization KPI 5.6
6	G can enable his webcam and share this video with A	Video stream from G’s webcam to A	Visualization without delay in acceptable quality	Delay for synchronization KPI 5.6
7	G ask A to scan the environment	Asking via audio	n/a	
8	A starts the environment scanning procedure	A takes a number of pictures or a short video of the environment, then selects the option to send them to the reconstruction service.	A is able to take sufficient pictures and upload them to the CORTEX reconstruction service.	3D reconstruction KPI 3.1 KPI 3.4
9	End of the scanning procedure: the 3D model is sent to G	3D reconstruction service generates a 3D model with texture and returns it to A and G.	Quality of 3D model is sufficient on G side and allows for tracking the position on A side. Generation time is less than 3 minutes.	3D reconstruction KPI 3.2 KPI 3.3
Scenario IM2				
AR environment for the remote expert. G is able to visualize the environment of A, and can start manipulating virtual objects to augment interactively the scene of A. Modifications made by G are instantaneously propagated to A. In addition, the system can connect to IoT devices located on the production plant, and all participants can visualize sensors value in their augmented view. Participants can also trigger an action through the IoT interface.				
Nr	Step	Action	Result	Related KPIs
10	G can load and navigate in the reconstructed "3D scene"	G changes the viewing angle, position and	Intuitive navigation, speed and rendering quality are acceptable.	KPI 5.4

		zoom in the virtual scene.		
11	G wants to augment the scene with additional virtual objects such as arrows, virtual marks etc.	G "augments" the scene manually from a virtual set of tools to choose from (pinpoint, mark, arrow), then drag and drop to the 3D scene	The additional 3D objects are registered (anchored) with the virtual environment.	Stability of the augmentations KPI 5.5
12	G wants to share the additions with A	Scene synchronization (push-based on demand or continuous)	No delay when sharing virtual objects	Delay KPI 5.6
13	A sees the augmentations of G on the real environment	A can move freely and the augmentations are seen as being anchored on the real objects.	Tracking of the HMD has been "registered" with the coordinate system of the scanned 3D model	Stability of the augmentations KPI 5.5
14	A (or G) asks per voice command to display information from the local IoT objects	Voice is analyzed and leads to the correct action	Correct intent detection	Voice transcription and analysis KPI 3.13 KPI 3.14 KPI 3.16
15	A can see the real machine, augmented with the data from the IoT sensors (in addition to the data generated by G)	Information collected by the IoT devices is represented in the scene (either in 3D near the corresponding object or as floating window)	The list of available IoT devices can be retrieved and correctly displayed.	IoT devices KPI 3.10 KPI 5.22
16	A wants to switch off the power of a module using the IoT connection.	A (or G) can actuate some IoT objects by using the interface	Control of access right and possibility of actuation of IoT devices	Virtual IoT representation KPI 5.24
Scenario IM3				
Cooperation with other participants. A start to repair the machine with the help of G. A can receive additional information from G, which is displayed in her view. R is participating to the meeting with reduced bandwidth. The three participants have a videoconference to discuss the problem.				
Nr	Step	Action	Result	Related KPIs
18	A start manipulating the real machine and wants to be guided in the exact manipulation by G	A sees the hands of G (virtual reconstruction of the hands) in his virtual view of the machine	Correct tracking of the user's hands on G side. Correct positioning of the virtual hands-on A side.	Hand gesture recognition KPI 3.9

19	G wants to add additional diagrams to the augmented view (notice or 3D model explosion)	G should choose additional information and position in the AR view (or at fixed position).	Different types of data can be visualized (images, 3d models etc.)	UI evaluation KPI 5.4
20	Thanks to the indications of G and the augmentations, A can do the right action (e.g. check if a fuse is on)			Task completion KPI 5.3
21	The problem is solved			Task completion KPI 5.3
22	R was attending the meeting and could follow the actions in video views.	A and G summarize the problem with R in a “classical video conference “: R sees G and the real augmented view of A.	Multi-party and heterogeneous conference using classical Rainbow interface.	Number of participants KPI 5.21
23	R wants to reduce the bandwidth when transmitting her video	R uses the VCAA to send her video in reduced bandwidth	Video compression produces a video in acceptable quality.	VCAA bandwidth reduction KPI 3.8
24	End of the conference			

3.2. Training Pilot

The goal of this pilot is to offer a connected and immersive remote training experience assisted by a trainer in a virtual environment. The VR training environment is open to several kinds of participants, first and foremost a trainer that remotely participates in training classes and can give advice to trainees whenever needed. Connecting trainees virtually offers the possibility to open a classical single-session lesson to several participants at the same time. VR training with several trainees and one trainer in a virtual classroom is also possible. In this later, both group classes and single-user training sessions are possible, so as collaborative work.

Figure 9: Pilot 2 illustration

Virtual training, presented in Figure 9, in industrial programs is designed to help trainees learn and improve their skills before they start working on a machine. The main objectives of these trainings aim at:

- The comprehension of the main components of the machine.
- The correct operation of the vehicle in a safe way, linked with its surrounding environment.
- The improvement of the efficiency while using the machine, improvement of the skills and the tuning of the settings.
- The use of the virtual world allows to simulate dangerous situations, the collaborative aspect shall allow also to illustrate misuse scenarios of the machine.

3.2.1. Actors and roles

Four main roles are defined for the execution of the scenarios in this Training Pilot:

- Trainee: represents a human actor (participant) that takes part to a training session to learn or assess some knowledge.

- Trainer: represents a human actor (organizer) that can teach and help the trainees during the training sessions. He initiates the meeting in the Rainbow™ bubble. He should be able to mute/unmute any of the participants if necessary.
- Supervisor: a human user too (participant), head of the training department, responsible for training quality and customer satisfaction with the training program. He may also follow the trainer's performance.
- Administrator: is an actor of type organizer. He can add or remove training sessions, administrate the solution and the participant's' rights.

All the actors involved in the training session are connected in audio conference and we may provide a virtual assistant for facilitating hands-free operations in some scenario sequences or if the actors need to hear instructions or description of training steps instead of reading them.

3.2.2. Scenes and objects

The scene is the base container of the training, it gives a context to the training session and a content. Several kinds of scenes are planed:

- An Office: specific scene dedicated to the group sessions, allowing to present contents like miniature 3D interactive objects and 2D placeholders (to display photo, video or presentations). Streams coming from the Rainbow/CORTEX² services can also be broadcasted on these surfaces.
- A Training Scene: this scene will probably be textured specifically to match each of the training goals. This means a training for an excavator shall be represented in the field with a construction site background.

Both static and dynamic objects will be available to enable collaborative interactions during group sessions.

3.2.3. IoT objects

In the context of the use case, featuring the excavator, we don't see any direct application or immediate integration of real IoT sensors in the scene. However, we do see the value of integrating them into a wider training context.

Pilot 1 already provides for the integration of IoT components into the platform, and as the technological platform of these 2 pilots is identical, we will prepare the integration for a later use of the module already tested in other pilots.

3.2.4. Intentions and behaviour

Trainees will participate by using head-mounted devices, so it will be impossible for us to analyse the emotions on their faces or their face movements. However, by analysing their behaviour via hand-tracking or eye-tracking, hesitations or repeated errors, we will be able to offer personalised help via virtual assistants or raise alerts with the trainer to request assistance. The trainer will organize the sessions and choose them in adequacy with the level of knowledges of the participants. He can see each participants training session on a panel of live screens. Furthermore, the training is adapted to the user's level in real time.

The supervisor will intervene if a risk is detected (eg. motion sickness).

3.2.5. Execution devices

In order to ensure an effective and positive learning and teaching experience, the actors in the training use cases should be equipped with the following devices:

- PC with camera for the trainer and the supervisor.
- VR device for the trainees.

For the first release, the development is tested on the 2 following standalone devices: the Meta Quest 2[25] and the HTV Vive Focus 3[26]. However, the use of hardware abstraction layers in the development aims having the highest compatibility with the largest number of devices on the market.

3.2.6. Reusable features

During the development and in conjunction with the creation of the base scenarios defined for the pilot, we will already create and integrate following components or features:

- To populate the training scenes with featured objects, Prefabs of game objects are created and will be reusable.

[25] https://www.meta.com/fr/quest/products/quest-2/
[26] https://www.vive.com/fr/product/vive-focus3/overview/

- 3D scans of real machines/devices/environments to create the training scene.
- Eventually chatbot with speech to text function to search information directly in the training's documentation (CORTEX² speech prefabs)

3.2.7. Genericity of the application

Each time a new training course is integrated, the scenario that the trainees will have to follow and carry out will have to be configured/developed. To enable the CORTEX² platform to be opened, we rely on two existing authoring tools, Unity3D and Mozilla Hubs, to ease the developers in CORTEX² specific modules integration within a 3D application. Unity Editor is bringing all what we need for CORTEX² from 3D scene construction and animation point of view and for integrators/developers that makes it easy for them to enter quickly and efficiently in the proposed CORTEX² framework.

3.2.8. Scenario of use for the Training pilot

This section details a "scenario of use" for the Training pilot with a corresponding temporal sequence presented as step/action/result. In addition, at each step is associated the KPI being assessed.

The remote training use cases describes Jane's remote training experience for a new excavator license using VR technology. Jane and her instructor, Thomas, connect from different countries using VR headset and PC to simulate a hands-on training session without disrupting the excavator's use on-site. They communicate through a video conference where Thomas guides Jane on using the VR headset.

Through the VR glasses, Jane sees the training environment and interacts with tasks related to operating the excavator. Thomas adapts the difficulty of the lessons based on Jane's progress and can change settings in real-time. They communicate using voice and the system provides a summary of their discussions for reference.

Additionally, Thomas can conduct training sessions with multiple participants. These sessions involve a theoretical and practical aspect, with participants using VR headsets to interact in a virtual classroom. Thomas shares presentations on a virtual screen and then guides individual practical sessions for each participant, offering assistance when needed. He can manage audio and room settings and occasionally has a supervisor, Maria, observing the sessions to provide feedback on Thomas's training.

A full narrative of the scenario of use for the industrial maintenance use case if provided in section 2.2 “VR Remote Training” of the Deliverable D5.1 (p. 26). There, we describe the actors, the devices and the two scenarios of the use case. In addition, we provide in the present deliverable a complete scenario of use with individual steps to perform the scenarios, corresponding actions, expected results, and corresponding KPIs[27]. The following table shows the scenario of use of the remote training use case.

Table 2: Scenario of use for the Training Pilot

Scenario TM1				
J and T connect from different countries using VR headset and PC to simulate a hands-on training session without disrupting the excavator's use on-site. They communicate through a video conference where T guides J on using the VR headset. T adapts the difficulty of the lessons based on Jane's progress and can change settings in real-time. Note: Jane is a trainee and is named “J”, Thomas is a trainer and is named “T”				
Nr	Step	Action	Result	Related KPIs
1	T starts the Cortex² Trainer App	Start of the Cortex² Trainer app on his PC and logs in	Cortex² PC Application is started	KPI 5.20
2	T logs in as a trainer	Authenticate using valid trainee’s credentials	The user is logged on as a trainer	Max concurrent Trainer connections: 1
3	T chooses his training Session (Rainbow Bubble)	Click on the desired session	Trainer is connected to the bubble. A view with all connected trainees is shown	Max available sessions: 20 KPI 5.21
4	J puts on a VR headset and starts the Cortex² VR Application on the device	Launch the application	Cortex² VR Application is started on the headset	Latency: 100ms Compatible with open xr devices KPI 5.8

[27] The list of KPIs if provided in deliverable D1.1, section 6.2

5	J joins a first lobby room where she can find a view to authenticate	Move in the Lobby, Interact with the environment, Select the authentication form	Cortex² VR Application shows the lobby and allows moving in the scene and interact with some objects, as a virtual keyboard to log in	VR quality High frame rate min 50 FPS KPI 5.9
6	J logs in as a user	Authenticate using valid trainee's credentials	The user is logged on as a trainee	Max concurrent trainee connections: 10 KPI 5.21
7	J chooses her training Session (Rainbow Bubble)	Click on the desired session.	Trainee joins assigned course (scene)	Max available sessions: 20 Data/file size 1Gb KPI 5.20
8	The assigned scene of the course is launched in J's environment	Start of the Training scenario	The Training Scenario is launched, and the trainee can start discovering the contents and interact with it.	VR quality High frame rate min 50 FPS KPI 5.9
9	J can interact with the scene in the training scenario	read text contents, listen to audio contents, visualize video contents, navigate to the next content, move and/or teleport at several stages in the scene, Interact with objects	Contents is shown, Contents is browsed, User moves around the scene.	VR quality High frame rate min 50 FPS KPI 5.12
10	T can see J's current viewpoint on his control screen	View the scene, Display it full screen,	Scene from the trainee is shown on the monitoring view of the trainer	Resolution quality 720p Real time KPI 5.11
11	T and J can discuss together	Activate sound communication. Mute/Unmute	Trainer and Trainee can have a discussion, The trainer can mute and unmute the trainee	

12	T can freeze J in case of incident	T clicks on Jane’s screen and then on the freeze/unfreeze button	J cannot interact or move but can hear T	Motion sickness prevention
Scenario TM2				
T can conduct training sessions with multiple participants. These sessions involve a theoretical and practical aspect, with participants using VR headsets to interact in a virtual classroom. T shares presentations on a virtual screen and then guides individual practical sessions for each participant, helping when needed. He can manage audio and room settings and occasionally has a supervisor, M, observing the sessions to provide feedback on T's training. Note: Jane is a trainee and is named “J”, Thomas is a trainer and is named “T”, Maria is a supervisor and is named “M”				
Nr	Step	Action	Result	Related KPIs
13	T starts the Cortex² Trainer App	Start of the Cortex² Trainer app on his PC and logs in	Cortex² PC Application is started	
14	T logs in as a trainer	Authenticate using valid trainee’s credentials	The user is logged on as a trainer	KPI 5.21
15	T chooses his training Session (Rainbow Bubble)	Click on the desired session	Trainer is connected to the bubble. A view with all connected trainees is shown	Max concurrent Trainer connections: 10 Max available sessions:20 KPI 5.20
16	T can add another trainee to the training session thanks to his or her email address	Invite a new trainee	New trainee’s information is in the list	Latency:200ms KPI 5.13
17	T can manage courses assignments	T chooses scene in the trainee’s drop-down menu	Trainee’s scene changes	KPI 5.19
18	If needed, T can delete one or many trainees	T clicks on a trainee’s name	Trainee is logged out	
19	J and other trainees log in as users	Authenticate using valid trainee’s credentials	The users are logged on as trainees	Max concurrent Trainee connections:10

				KPI 5.21
20	Each user chooses her/his training Session (Rainbow Bubble)	Click on the desired session.	Trainee joins assigned course (scene)	Max available sessions:20 Data/file size:1Gb KPI 5.20
21	The assigned scene of the course is launched in each user's environment	Start of the Training scenarios	The Training Scenarios are launched, and the trainees can start discovering the contents and interact with it.	VR quality High frame rate min 50 FPS Number of concurrent training sessions:20 KPI 5.10
22	Users can interact with the scene in the training scenario	read text contents, listen to audio contents, visualize video contents, navigate to the next content, move and/or teleport at several stages in the scene, Interact with objects	Contents is shown, Contents is browsed, User move around the scene.	VR quality High frame rate min 80 FPS KPI 5.12
23	T can see all trainees cameras on his control screen	View the scenes, Choose to display one trainee's view on full screen, Intervene in one of the training sessions.	Scenes from several trainees are shown simultaneously on the monitoring view of the trainer.	Max visible parallel sessions:10 KPI 5.11
24	Users can switch anytime to the lobby room	Leave the scene and switch to the lobby room	Users are in the lobby room	Maximum numbers of users in the lobby room:10
25	T can share his screen or camera in the lobby room screen panel	Share screen or Share camera	Trainees can see the screen (resp. the camera) of the trainer on the big presentation screen of the lobby	Image quality/resolution: 720p

26	T can present some specific and targeted training content to J through screen sharing. T & J can discuss the points together.	Share a presentation or a specific content that is not present in the training scenario but can help the users' training	J follows the content and hears the trainer's explanations, she can also ask questions	Resolution quality (video/presentation):720p KPI 5.12
27	M logs in as a supervisor on the beginning of the training session	Authenticate using valid supervisor's credentials	The user is logged on as a supervisor	Max concurrent Supervisor connections:1 KPI 5.21
28	M chooses his training Session (Rainbow Bubble)	Click on the desired session	Supervisor is connected to the bubble, A view with all connected trainees is shown	Max available sessions:20 Data/file size:1Gb KPI 5.20
29	M listens and reports about the quality of the session (view-hear-talk functionalities)	Only use of same view rights as the trainer, no interaction with the trainees	Trainee's scene is shown on the monitoring view of the supervisor.	Resolution quality (video/presentation):720p KPI 5.11
30	End of the training session			

3.3. Business Meeting Pilot

This pilot consists of a virtual reality business meeting in which users can participate via conventional devices, cf. Figure 10. In addition to the immersive conference, the team will experiment additional virtual reality functionalities such as alternative appearances of equipped users, keep in touch with the real world via IoT objects that are visible in the virtual scene and the involvement of virtual assistants to enrich the discussion with additional information during the meeting. In this case, Business meeting CORTEX² application is of **type 2: "shared application"** (cf. Section 2.2.1). It means that we will have only one scene shared across VR users but as mentioned before, other users can join with audio/video via Rainbow™ videoconference solution and only presentation sharing will be allowed.

Figure 10: Pilot 3 illustration

3.3.1. Actors and roles

The three roles defined by model are present in this pilot. There will be two or more human actors participating to the meeting using virtual reality headsets among them a meeting organizer and an AI agent that play the role of a non-autonomous virtual assistant. In addition, as already noted, other human users can join the conference from the classic Rainbow™ video conferencing platform, as described by Figure 11.

- **Organizer**: initiates the meeting in the Rainbow™ bubble. The participants and the IA assistant should be already registered to this bubble as members. When the organizer starts the meeting, those using Rainbow™ platform will be notified to join the meeting. However, for the users using headsets devices, Business meeting CORTEX² application should be up and running to be able to join. The IA assistant will connect automatically after startup. This approach can be improved in future versions when users and conference rooms registration are fully resolved between CORTEX² and Rainbow™ with compatibility preserved, allowing classic users to meet virtual ones.
- **Participant**: registers as a member of a conference bubble and accepts to join the conference when started.
- **AI assistant**: is a predefined member of the bubble and starts automatically to conference when started. It is not completely autonomous but act only when an intent

was detected. Developers can adapt its functioning to provide more complexes behaviours.

- **IoT AI agent:** The CORTEX² component that receives real IoT sensor information and transmits it to the virtual environment of a specific participant, acting as the bridge between the IoT environment and the VR environment.

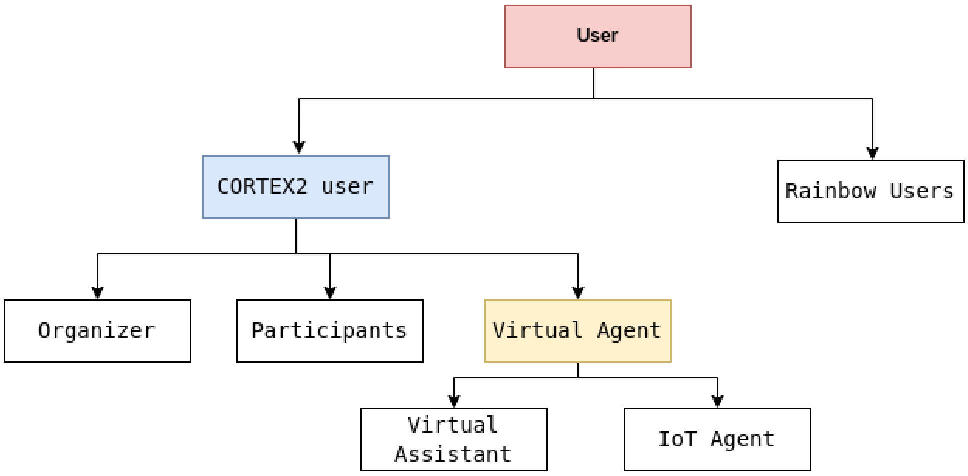

Figure 11: Business meeting actors

3.3.2. Scenes and objects

The meeting scene, cf. Figure 12, is mostly statistical. It will include static objects: a simple background and a table with monitors for remote users using Rainbow™ videoconferencing, and avatars for virtual headset users. Since avatars are dynamic objects, the presence of several actors leads to synchronization across VR participants. We're targeting head movement in particular to validate the user experience, but other movements may follow depending on the use cases of the project FSTP call.

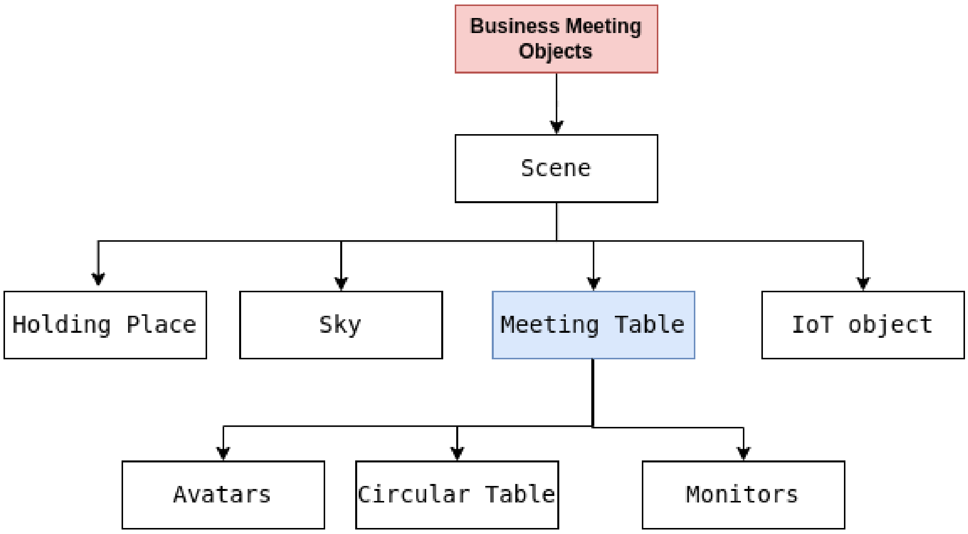

Figure 12: Business meeting scene structure

3.3.3. IoT objects

Remote conferences using virtual reality headsets can be held from the participant's home or workplace. Being in an emergent meeting mode, the person can forget about the real world for a while, and not worry about any emergencies that may arise: an alarm in the building, or a call from a child in daycare at his father's house, etc. This is only part of the reason why IoT integration is so important, but still useful for other situations. In Business meeting, a 3D dynamic object will be associated to the IoT object and displayed in the authorized user VR world. The object will change the colour depending on the events received from the IoT agent which in turn is listening to the real IoT object.

Remote conferences using virtual reality headsets can be held from the participant's home or workplace. By using IoT objects in the participant's personal environment e.g., his house, the emergent meeting mode is able to deliver notifications to the participant about any emergencies that may arise, such as an alarm in the building or a call from child daycare etc. Another reason why IoT integration is so important is the IoT objects visualization aspect, which makes the remote conferencing experience much more immersive and the collaboration between participants more effective. In the Business meeting, an IoT virtual object will be associated to the real IoT object and will be displayed in the authorized participant's VR world. The virtual IoT object will be able to change appearances (e.g., colour) depending on the events received from the IoT agent which in turn receives the relevant events from the real IoT object.

3.3.4. Intentions and behaviour

In Business meeting, participants using virtual reality headsets are represented by avatars in the scene. Thus, speech request and gaze changes will be considered in the final version of the demonstrator. Furthermore, only verbal intentions will be considered in this application: intentions that enable the virtual assistant to perform an action or trigger an event. Vocal intentions that generate user commands in the virtual space can also be integrated if a strategy is found to overcome the ambiguity between user speech commands and the discourse progress of the conference. Figure 13gives a quick view of these functionalities.

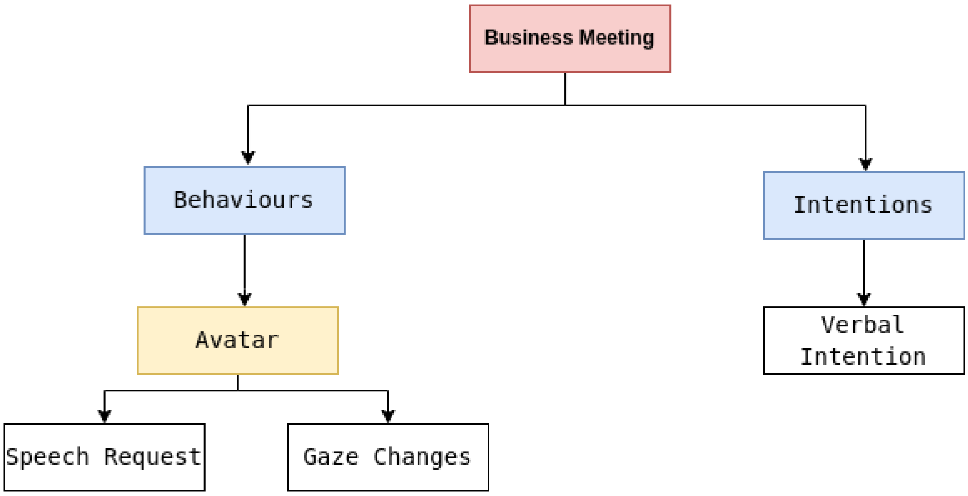

Figure 13: Business meeting behaviours and intentions

3.3.5. Execution devices

Virtual reality headsets with WebXR-compatible browsers are the preferred choice at this stage of development. The use of WebXR standards ensures that the application will run on a majority of XR devices. Particularly, Meta Quest 2[28] will be used for the experimentation of the Business Meeting application. Furthermore, the conference can be accessed from smartphones or laptops using the Rainbow™ video conferencing solution.

3.3.6. Reusable features

Business Meting CORTEX² application has been developed with reusability in mind. The design structure is shown in Figure 12. All these objects can be replaced to give a different look to the

[28] https://www.meta.com/fr/quest/products/quest-2/

scene. Developers wishing to go further in the design are invited to contribute in-depth to the code.

Concerning the communication interface with CORTEX² Core, which is in turn managed by Rainbow, a component A-Frame is developed to facilitate its integration within 3D scene object. A-Frame, threeJS and additional libraries are open-source libraries which have proven their efficiency being part of Mozilla-Hub. In the long term, it will be very easy to connect from the virtual environment without too much development. We're planning to use a frontend project template to facilitate scene personalisation and connection to the CORTEX² Core. A guide will be written for the CORTEX² developers as a complement to the present deliverable and will be available at the end of the project. In addition, further information will be provided on how to integrate external services, such as virtual assistants. This last aspect is common to all use cases. External services will be handled in the same way for all applications, as described in Section 4.

3.3.7. Genericity of the application

The CORTEX² framework provides a set of libraries and tools to facilitate application development. In this sense, we have developed this pilot in compliance with the model established above, so that the code can be reused by other applications and for convenience, personalizing the application, I.e., design of the scene, can be done more quickly and without extra effort.

3.3.8. Scenario of use for the Business Meeting pilot

This section details a “scenario of use” for the Business Meeting pilot with a corresponding temporal sequence presented as step/action/result. In addition, at each step is associated the KPI being assessed.

The business meeting use case showcases the use of CORTEX² as a platform for a remote business meeting, at which various actors participate using different access modes (desktop PC, smartphone, VR headset). For business meetings, the CORTEX² includes a virtual assistant called COVA that assists the participants of the meeting by providing information of summaries.

In this use case, Jane is preparing for a virtual business meeting with her boss Maria and colleagues John, Liz, and Thomas. Maria and John are on a business trip, while Liz and Thomas telework from home. Using the CORTEX2 platform for the past three months, Maria sent them a meeting link. Equipped with VR glasses and hand sensors, Jane connects to the meeting. COVA, the CORTEX2 Virtual Assistant, asks for authorization to access her agenda. Liz, teleworking, also wears VR glasses and sensors, managing her phone for an important call. Thomas joins with VR glasses but without hand sensors, Maria from a hotel room, and John via smartphone. In the VR conference, avatars of the team gather around a virtual table in a 3D room. COVA's avatar aids in managing the meeting, sharing screens, and summarizing discussions. Thomas uses both his microphone and the meeting chat functionality to ask questions. Liz briefly steps out to answer her phone, while Jane efficiently shares and summarizes documents with COVA's assistance. COVA also aids in web searches and scheduling, facilitating smooth communication. At the meeting's end, Maria suggests the next meeting date, which COVA promptly arranges based on everyone's availability. Jane appreciates CORTEX2's ability to automate tasks like meeting minutes, simplifying post-meeting procedures. She finds the VR meeting environment conducive to focus, free from office distractions.

A full narrative of the scenario of use for the industrial maintenance use case if provided in section 2.3 "*VR Business Meeting*" of the Deliverable D5.1 (p. 34). There, we describe the actors, the devices and the three scenarios of the use case. In addition, we provide in the present deliverable a complete scenario of use with individual steps to perform the scenarios, corresponding actions, expected results, and corresponding KPIs[29]. The following table shows the *scenario of use* of the business meeting use case.

Table 3: Scenario of use for the Business Meeting pilot

Scenario BM1				
Jane is at her office starting her workday. She has a business meeting with her boss Maria and her colleagues John, Liz, and Thomas. Maria and John are out of town on a business trip, and Liz and Thomas are at home teleworking. Maria sent them a link to connect to the meeting yesterday.				
Nr	Step	Action	Result	Related KPIs

[29] The list of KPIs if provided in deliverable D1.1, section 6.2

1	Maria registered as a conference admin using the Business Meeting application.	User registers using his email and bubble name.	User receives confirmation and is admin on the given conference bubble	Number of simultaneou s conferences KPI 5.20
2	Maria invites Jane, Thomas, Liz and John by email; to where they can get access credentials.	User provides participants emails.	Each participant receives his/her access credentials	Number of users by Conference KPI 5.21
3	Two more hidden participants are registered automatically to provide additional features for the meeting, namely: IoT Agent and CoVA (virtual assistant)	CoVA and IoT agent instances are created	CoVA and IoT agent are member of the conference bubble	Number of languages, CoVA supports / Number of IoT devices to link to VR KPI 3.14 KPI 3.10
4	All participants can now attend the conference and Maria can start the meeting.	Participants login to the application	The meeting is ready to start	Number of CoVA instances working synchronous ly with different conferences KPI 5.23
Scenario BM2				
Everybody is at the VR conference now. Different avatars sit around the table. Jane feels now in a 3D space. She is in a virtual room.				
Nr	Step	Action	Result	Related KPIs
5	Laptop and Mobile users (Maria and John) appear in the conference with real face or using VCAA (video compression and alternative appearance)	User activates VCAA	The appearance of the user is different and follow the user head movements	Number of VCAA to choose KPI 3.5 KPI 3.6 KPI 3.7
6	VR users (Jane, Liz and Thomas) appear as 3D avatars	User selects his/her avatar	User appears like an avatar sitting around the meeting Table	Number of Avatars can participant choose

				KPI 5.17 KPI 5.18
Scenario BM3				
Also present in the virtual meeting room is COVA's avatar, a simple icon-like representation acting as CORTEX2 Virtual Assistant. Upon connection, Maria activates CoVA which in turn starts listening and requests each participant authorization access to the agenda.				
Nr	Step	Action	Result	Related KPIs
7	Maria activates CoVA, the virtual assistant.	CoVA started	CoVA is listening. [User/Conference CoVA modes allow detection of the speaker for answering to all participants or to a given participant]	Number of languages, CoVA supports KPI 3.14
8	CoVA request access to each participant agenda.	Each participant answers CoVA question [Textual/Voice]	CoVA has access to authorized agenda	Speech recognition capability (TTS/STT) KPI 3.13
Scenario BM4				
Liz wears also her VR glasses and hand sensors from home. Apart from the meeting, she must pay attention to her cell phone because she is expecting an important call from one of her customers.				
Nr	Step	Action	Result	Related KPIs
9	IoT device (including mobile) is registered with the appropriate authorizations (shared among all conference participants or visible to owner only)	Liz registers her IoT device	IoT device is linked to user VR environment	Number of devices registered per user KPI 3.10
10	IoT device appears to the owner environment or to all users depending on the authorization.	Application creates IoT device object in the VR environment of Liz	IoT device is only visible to Liz	Number of devices displayed in VR KPI 5.22
11	IoT device starts sending data	Application receives IoT device data	IoT device status is shown in VR IoT device object.	Type of IoT devices to consider KPI 5.24
Scenario BM5				

Maria shares her screen to show the slides that will guide the meeting. All can see the presentation on the VR screen. The meeting went well. All the conversation is in English.				
Nr	Step	Action	Result	Related KPIs
12	Maria shares the slides	User uses sharing screen button	The slides are visible to everyone and appear also in VR monitor object	KPI 5.17
13	Maria asks CoVA to start recording the meeting. The file will be used for meeting summarization	User activates “User mode” and talks to CoVA	The command “recording” is detected by CoVA and “conference mode” is activated	Number of CoVA supports commands KPI 3.16
Scenario BM6				
On some occasions the microphone of Thomas does not work very well. He writes down in the chat his question and a conversational assistant pops up and tells us what Thomas is writing.				
Nr	Step	Action	Result	Related KPIs
14	Thomas writes a chat message	User writes a chat message	The message arrives to CoVA	Number of CoVA supports commands KPI 3.16
15	CoVA detects the Thomas command and synthesizes the messages	CoVA understands that he is the recipient of the message	CoVA uses Text-to-Speech and read the message	Number of intents supported, Quality of search KPI 3.11 KPI 3.12
Scenario BM7				
At a certain moment, there is a sign on the big screen for Liz, her cell phone is ringing. She leaves the meeting for some minutes and then she comes back.				
Nr	Step	Action	Result	Related KPIs
16	Liz receives a notification indicates phone ringing	VR IoT device object shows “Phone incoming call”	User status changes to “Absent”	Number of IoT devices and types KPI 5.24
17	Liz come back	User changes him/her presence state	User status is now “online”	Number of IoT devices and types KPI 5.24

Scenario BM8				
Jane raises her hand and shares her screen to show some administrative documents related to 50 pages report. COVA detect the intention and quickly summarizes the document to a half page size and makes the summary available in the chat.				
Nr	Step	Action	Result	Related KPIs
18	Jane raises her hand [her avatar hand is visible to VR users and speech request icon indicates Jane]	VR User [avatar] raises hand	A speech request is noticed by all users	Number of gestures KPI 3.9
19	Jane starts discussing about the document information	CoVA is listening	CoVA detects the intention and provides the summary in the chat.	Number of intents supported KPI 3.11
Scenario BM9				
John asks if this customer also has activities in the US. COVA hears this question and searches the internet. As there is a short silence, it speaks out the answer it found.				
Nr	Step	Action	Result	Related KPIs
20	John activates his microphone and starts discussing.	User mode is activated	CoVA instance is listening to the user	Number of CoVA instances working synchronously with different conferences KPI 5.23
21	CoVA detects John question and makes an internet search followed by a voice response	CoVA is detecting intention	A "search" intent is detected, and a response is provided [textual/voice]	Number of intents supported KPI 3.16
Scenario BM10				
At the end of the meeting, Maria suggested finding a date for the next meeting during the next week. COVA speaks out to propose a date and time compatible with all participants' agendas. As nobody objects, Maria says that it's OK and COVA sends an invitation.				
Nr	Step	Action	Result	Related KPIs
22	Maria discusses the next meeting date	CoVA is listening	CoVA detects appointment organization	Number of intents supported KPI 3.11

23	CoVA detects the intent and propose a date	CoVA accesses participants agendas and proposes a date	A date for the next meeting is proposed	Number of CoVA instances working synchronously with different conferences KPI 5.23
24	Maria validates the date with every one	CoVA is listening	CoVA is processing the stream	Number of intents supported KPI 3.11
25	CoVA sends the invitation for the participants	Detecting the agreement about the date by CoVA	Invitation is sent to participants	Genericity of the API for external services integration
Scenario BM11				
Jane is happy because she does not have to write the minutes of the meeting. CORTEX2 does that for her. They can download the summary of the meeting directly from the platform containing the link to slides in SharePoint.				
Nr	Step	Action	Result	Related KPIs
26	Maria [organizer] is redirected to Linto Studio where she can fine-tune the meeting summary and add additional remarks.	User updates the summary	Summary is ready	Quality of summarization KPI 3.15
27	Maria validates the summary which will be sent to the participants.	User sends the meeting summary	Meeting summary is received by the participants	Quality of summarization KPI 3.15

4. External services

CORTEX² has been conceived as a modular platform and allows for the addition of external services. Some of these services are developed by the consortium partners in dedicated tasks (e.g., Video Compression and Alternative Appearance, or 3D reconstruction of the environment), others will be added thanks to the participation of third parties to the project, and possible extended external services will be added to the platform in the exploitation phase after the end of the project. All of these external services need to be compliant with the abstract models defined in this deliverable in order to ensure a seamless integration of the services in the existing framework. In this section, we list the criteria the external services need to follow in order to be integrated easily in CORTEX².

4.1. Input and output

An external service needs to define a clear input and output based on the information presented in Table 4.

Table 4: External service meta-data

Information	Examples
Data type	Text, image, video sequence, 3D model
Encapsulation format	Json file, video stream, audio stream
Service synchronicity	Asynchronous (delivery of output after processing with dedicated channel), synchronous (instant delivery of output)
Service scalability	1 request at a time, 100 simultaneous requests

4.2. Service processing location

Each service can run locally at the client side, on a server related to CORTEX² core, or on the cloud (distant server). The external services must declare where the processing will take place.

4.3. Hardware compatibility

A service might necessitate specific hardware for data acquisition, or for data presentation. For example, eye-tracking might be available only on a limited number of AR devices. An external service must precise the list of compatible hardware with following information: List of currently supported hardware, requirements for support of future hardware

4.4. Integration in behavioural model

An external service might influence a CORTEX² application behaviour. For example, if a service loads a new object in the scene at runtime, this object should include its behaviour or intention if it is a dynamic object, thus the scene could manage it correctly. Therefore, it is important and recommended that any external service used within CORTEX² should describe the intended behaviour of its output, and the technical details on how to access and react upon this behaviour. To this aim, the external services have to provide a documentation and a clear interface on how the CORTEX² framework can use the result of the service and adapt to the intended behaviour.

5. Ethical and social compliance

Social and psychological factors, together with biological factors are those whose particular qualities determine a specific lifestyle for each individual. These factors play a significant role in whether a user decides to use new technologies or not. CORTEX² is aware of the importance of psychological and social factors and ethical issues in the development of its collaborative model and in the development and testing of its applications. Because of that a whole WP is devoted to this: WP4 "**Ethical, legal and social aspects**". This WP has been in close collaboration with WP2, WP3, and WP5 to determine the psychological and social variables and the ethical issues relevant for CORTEX². The results of the analysis and studies carried out in WP4 are included in D4.1 - Ethical and legal inventory and D4.4 - analysis of psychological factors to optimize CORTEX². In this section we will present a summary. A more detailed information is included in D4.1 and D4.4.

The CORTEX² abstract model takes into consideration the user experience with rigour and detail. The model restricts CORTEX² domain of applications to collaboration and cooperation environments using XR technologies where the users play different roles depending on their involvement and use different devices to cooperate with other participants. Moreover, additional features can be brought in to facilitate collaboration, such as 3D object scanning, augmented reality annotation and automatic notetaking. In addition to voice and video communication, there's a highlighting of the subjects discussed during cooperation through the manipulation of 3D objects and immersive interaction. The representation of the users in the

VR environment is also complex and carefully addressed, with a variety of representation, from images to VR avatars.

During the work done in WP2 and WP5 (requirements), a thorough analysis of social and psychological factors was conducted. Based on this data, points that could potentially impact CORTEX² development were selected, leading to a set of specifications to be considered. Those factors are related to the concept of perceived control and are usability, techno-fatigue, and technostress. In the requirements we include these factors as well as the mitigation actions to optimize or reduce them in the future CORTEX² applications.

Another factor that is very important in CORTEX² is the use of avatars to represent the users. Incorporating user avatars in a VR environment can lead to potential challenges, affecting the overall experience. On one hand, using avatars that closely resemble reality can trigger feelings of insecurity while using the tool (Thaler et al., 2018). On the other hand, allowing avatar customization may lead to altered behaviour (Yee et al., 2009) or the development of psychological conditions like body dysmorphia (Raman, 2016). To explore the advantages of user acceptability and technostress caused by the difference in avatar quality, a study was conducted with 42 participants. This study analysed three types of avatars: hyper realistic, non-realistic, and the use of avatars belonging to others. The goal was to understand the benefits of each type and gather user preferences in different contexts. The details of this study are provided in the deliverable D4.4.

The study revealed valuable insights regarding the use of avatars in different settings, including work and training. The results showed that users have a highly positive intention to use avatars in the work and education contexts and technostress related to avatar use was low. The results confirmed that the sense of presence is crucial. The presence factor was inversely related to technostress caused by avatar appearance. As the sense of presence increases, users perceive the avatars as more useful and express a higher intention to use them. To maximize the benefits of avatars, it is essential to focus on increasing the sense of ownership and minimizing the perception of body changes in the avatar. Regarding privacy fears, the results showed that in general, users have low privacy concerns when using avatars. However, this fear increased when users were informed that someone else could use their avatar. To address this, transparency about data usage and protection measures is crucial. Based on the importance of presence and the results related to avatar types, it was concluded that the best type of avatar to enhance presence is a realistic representation of the user with good image quality. This aligns with user preferences in work and education settings, where users prefer avatars that allow them to be presented in the best possible quality. Finally, the study highlighted the significance of users' prior experience with VR. Experience in VR positively correlated with increased presence, perceived usefulness, and intention to use avatars. This emphasizes the need for user-friendly manuals, excellent customer service, and comprehensive tutorials to help users adapt to the avatar system.

6. Conclusion

This deliverable completes deliverable D5.1 “CORTEX² requirements and deployment architecture”, which identified the requirements and proposed an architecture for CORTEX² framework. It focuses on the client side of the framework, i.e., the CORTEX² application. The goal is to give application developers a comprehensive information about the concept of a CORTEX² application, in order to allow them to develop applications using the tools and/or code libraries that will be made available. The document offers an intuitive UML abstract model of extended cooperation use cases. It also defines a set of concepts to facilitate interoperability and understanding during developers' exchanges.