



C O R T E X ²

ANNEX 1.1

Technical Description

Open Call 2:

Submission deadline: **15 August 2024, 17:00 CEST**



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement n° 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.

Disclaimer

This document may contain material that is copyright of certain CORTEX² beneficiaries and may not be reused or adapted without permission. All CORTEX² consortium partners have agreed to the full publication of this document. The document is provided with no warranties whatsoever, including any warranty of merchantability, non-infringement, fitness for any particular purpose, or any other warranty with respect to any information, result, proposal, specification or sample contained or referred to herein. Any liability, including liability for infringement of any proprietary rights, regarding the use of this document or any information contained herein is disclaimed. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by or in connection with this document. This document is subject to change without notice. CORTEX² has been financed with support from the European Commission. This document reflects only the view of the author(s) and the European Commission cannot be held responsible for any use which may be made of the information contained.

LEGAL NOTICE

The information and views set out in this document are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.



Contents

1. INTRODUCTION.....	4
2. CORTEX ² Architecture and Services.....	4
2.1. CORTEX Client components	5
2.2. Core components	7
2.3. Cortex services	8
2.4. CORTEX ² data privacy and security.....	9
2.5. Cortex Framework components available for 3 rd -party integration.....	11
2.5.1. Planned status of the Cortex Framework by end of February 2024 ...	11
2.5.2. Cortex Software Development Kits (SDKs)	12
2.5.3. How to start with developing a Cortex application or a Cortex component	13
3. CORTEX ² Pilots.....	14
3.1. Industrial maintenance	14
3.2. REMOTE TECHNICAL TRAINING	15
3.3. Business meeting.....	16
4. CORTEX2 Extended features	16
5. Concluding remarks.....	18

List of Figures

Figure 1: CORTEX ² high-level architecture	5
Figure 2: Architecture of the CORTEX ² Client.....	6
Figure 3: Architecture of the CORTEX ² Core Components	7
Figure 4: Set of the CORTEX ² Services	8
Figure 5: CORTEX ² Framework available for OC1 third party integration	11
Figure 6: CORTEX ² OC1 Co-development process	13



1. INTRODUCTION

The goal of CORTEX² Open Call 2 is to extend an opportunity to third parties to participate in the co-development of the XR and AR teleconference platform. The objective is to deliver an inclusive XR teleconference platform while involving organisations in the 'Lab-To-Market' stage that will bring new modules and features, enhancing the functionalities and opportunities CORTEX² can provide. Additionally, through open call 2 the project aims to assess and validate the social impacts associated with XR technology adoption. A total of 1 million Euro will be invested in CORTEX² Open Call 2.

This document referred to as ANNEX 1.1 Technical Description, provides a detailed description of CORTEX² architecture and services available to beneficiaries under their participation in the CORTEX² Assistance Programme for co-developers (9months duration).

The document shall be treated as an extension of Annex 1 – CORTEX² Guidelines for Applicants.

All associated Annexes must be additionally considered for the submission of a Proposal under CORTEX² Open Call 2.

2. CORTEX² Architecture and Services

CORTEX² aims to deliver a next generation telecooperation framework based on XR and AI technologies. This section provides information on the platform architecture that shows the modules, interfaces and covered functionalities that are available within CORTEX². These shall be considered from the applicants when suggesting their co-developed modules and features.

The architecture of the platform is described in three layers of Figure 1: the Cortex Client block that will work on the client side, the Cortex Core block that will work on the infrastructure or cloud side and the Third-party block that represents partner functionalities provided as a service, i.e., deployed in an external infrastructure.

The following sections provide detailed information about each of the individual components that make up the CORTEX² solution.

The description of the architecture covers modules that are currently being developed and all of them will be available only at the end of the project. Please refer to chapter 2.5 to have a clear snapshot of the available Cortex Framework when the OC2 projects are starting.



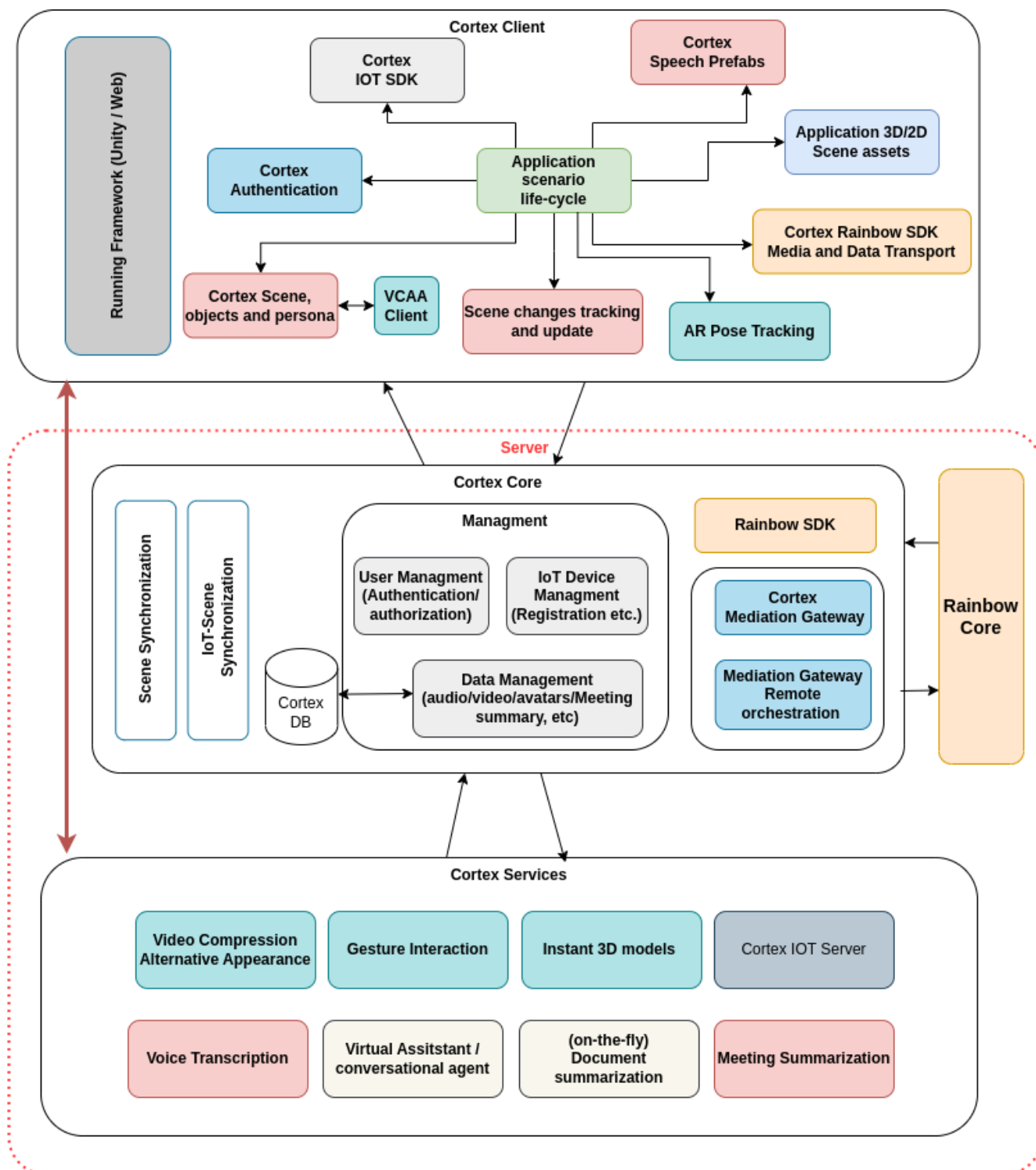


Figure 1: CORTEX² high-level architecture

2.1. CORTEX Client components

A CORTEX² application can have several views according to the role that the user can take. We can notice that especially for an industrial use case and a learning use case, different types of users are involved in the same application but have different views depending on the role they play in the cooperation.



The components of **Figure 2** will be developed to run on the XR device, laptop or smart phone in order to provide the user with the appropriate interface for his application. They are very dependent on the application domain. Due to the vast heterogeneity of possible applications, it is not planned to develop an application authoring tool to build applications. Instead, we provide interfaces for Mozilla Hubs and Unity for creating CORTEX² applications based on these specific engines. Each application may have to use only a part of these components because the user views, for the same application, could be very different, and some components may not be necessary for another type of application. A brief description of each component is given below:

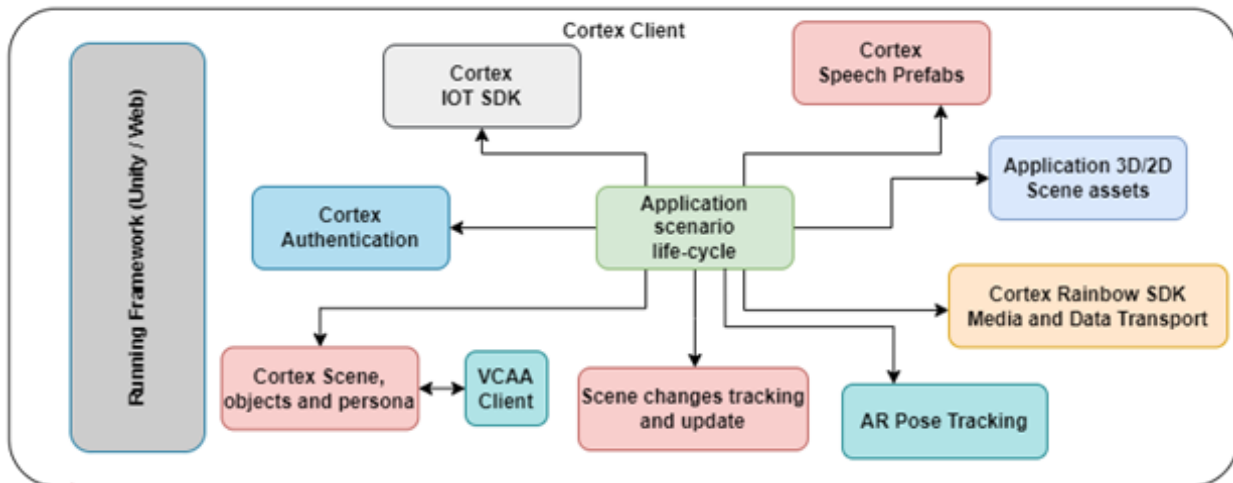


Figure 2: Architecture of the CORTEX² Client

- **Application scenario life cycle** is the heart component of CORTEX² APP. it manages the life cycle of the application by invoking the functionalities provided by other elements of the architecture.
- **Cortex authentication** is used for access management, user role and profile.
- **Cortex scene, objects and persona** is managing the different participants avatars and scene representation.
- **Scene changes tracking and update** shares authorized changes or events in a scene between the different participants, for example, if an object is moved by a user A, this might impact the rendering of user B application.
- **Application 3D/2D scene assists** are providing necessary objects needed to the construction of scene. Some predefined objects and scenes are stored in the Cortex core Database to allow quick starting of a Cortex app.
- **Cortex Rainbow SDK media and data transport** is used to connect to the core components and data transportation including the different media.
- **Cortex speech prefabs** is managing advanced conversational features and allow voice interaction to use within the rendering framework and targeted devices. For meeting summarization, users with an organisational role should authenticate to the conversational manager service provided by LINAGORA to review the auto-generated summary.

- **Cortex IoT SDK** provides the end-user applications with the libraries/mechanisms required to enable the access to and interaction with the IoT objects, either from the Cortex Core components or directly from the uiTOP platform, provided by ICOM uiTOP platform may have to integrate with the Cortex Core authentication mechanisms in order to handle users' access rights to IoT objects. The Cortex IoT SDK participates in the translation of the end user VR/AR "language" to an internal IoT format, readable by the Cortex Core or the uiTOP.

2.2. Core components

The core components (Cortex Core and Rainbow Core) are modules running on a server that can be used by any application. These modules are providing the core server functionalities that are necessary to run a telecooperation application. Two core components are identified and presented in **Figure 3**. They are detailed in the following sections.

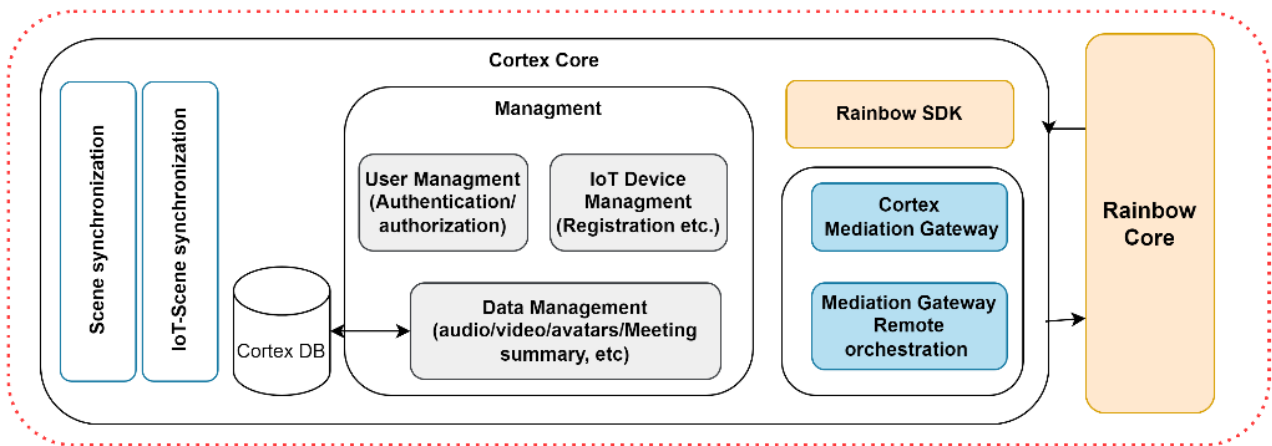


Figure 3: Architecture of the CORTEX² Core Components

1. Rainbow Core

Rainbow-core, provided by Alcatel Lucent Enterprise, is dedicated to connectivity and media transportation through its available SDK in the client side. Rainbow-core supports user authentication access using email and password, or the delegation of authority. It also includes the capability of collaboration and conferencing within “bubbles”, which are setup and monitored through the Rainbow SDK. Rainbow-core handles all communication channels for data exchange between participants in the collaborative room. It is also in charge of transporting medias, signalization and metadata to the right Mediation Gateway connected the XR services modules.

2. Cortex Core

Cortex-core brings together a set of components that are identified from the requirements and are developed within the project. They have the specificity to belong to the same deployment infrastructure, unlike some services that will be developed and deployed outside the cortex core environment.

- **Cortex-DB** is used for data persistence including 3D models, speech recognition data, users' profiles, IoT objects information, etc.

- **Management core** is made up of three components. They provide a set of utilities for user's and IoT registration, role management and authorizations. End-users' authentication is done through Rainbow SDK. It also allows the management and provision of media streams and other data in the platform. Particularly, IoT device management connects IoT world to Cortex XR world by giving access to associated sensors data or actuators. There is no generic administration GUI and it has to be considered according to pilots' usage.
- **Scene synchronization** is used to align scenes views across multiple shared virtual (or augmented) user environment. The synchronization will be achieved via data and events routing relaying on Cortex App components (Scene changes tracking and update, Cortex Rainbow SDK media and data transport) and Rainbow core.
- **IoT Scene synchronization** has the same goal as the previous element but with sharing restrictions taking into account the user's preferences and, security and privacy reasons. In case an IoT object is visible and usable by to other users, then actions or sensors status should be synchronized across users' scenes. Regarding IoT integration, uiTOP platform may be accessed either directly from the end-user application or through the Cortex core server components, also taking into account whether the information from the actuation/sensor needs to be shared or not.
- **Cortex Mediation Gateway** handles the routing of medias and metadata to the most appropriated XR and speech technologies services.
- **Rainbow-SDK** is a Rainbow Core Client simplifying the API access and management to the **Rainbow Core** APIs.
- **Mediation Gateway Remote Orchestration** is a component responsible of routing calls to external services from the Cortex Mediation Gateway.

2.3. Cortex services

Cortex services are modules provided as services and deployed outside the platform to facilitate integration. Their use is optional and dependent on the application. Incoming or outgoing media data, to these services, is driven by the Cortex Mediation Gateway via the Rainbow Core media server.

Identified services from the requirements are represented in **Figure 4** and are described in the following section.

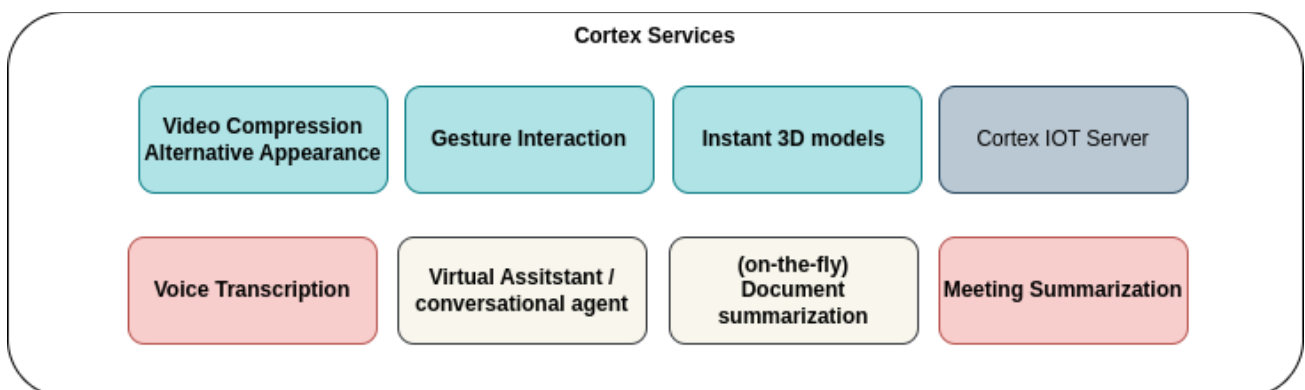


Figure 4: Set of the CORTEX² Services



- **Video Compression/Alternative Appearance service** is a module with two functionalities. On the one hand it can be used to save bandwidth when multiple users participate to the virtual cooperation environment. Instead of the transmitting 2D video, only metadata, low-dimension representation, is transmitted, and the video is reconstructed on the client side. On the other hand, the reconstructed video can be based on an alternative appearance, changing the physical appearance in the video of the user. DFKI will provide the encoder/decoder component.
- **Gesture Interaction service** reproduces hand poses from real user gestures and makes them available for a virtual interaction with real objects.
- **Instant 3D models service** implements and provides necessary APIs for AR scene analysis and objects recognition.
- **Voice Transcription service** provides live speech to text technology for different languages.
- **Virtual Assistant/Conversational Agent service** provides design patterns to handle live interaction between users and the Cortex app or users and the virtual assistant in cooperation environment.
- **Document Summarization service** is used to extract summaries from written documents, saved as files or stored in databases.
- **Meeting Summarization service** implements and provides discourse summarization API. In addition, a conversational tool will be made available to extend the generated summaries with annotation and NLP based tasks.
- **Cortex IoT server** is based on ICOM's uiTOP platform to implement and provide the necessary APIs for the management of IoT objects and their data including IoT objects registration/connectivity, monitoring/measurements, as well as actuation. The relevant flow goes through the Cortex core components.

2.4. CORTEX² data privacy and security

Security and privacy are two essential pillars in software development.

The following section gives a broader context of the concepts of data security, integrity, sovereignty, and privacy.

Data security is the procedure in which it is made sure that data is protected from being accessed, manipulated, or corrupted by unauthorized personnel or applications during its span of life. It includes activities such as data encryption and hashing, multi-factor authentication, access control, breach response, network security or activity monitoring.

Data integrity or often also called data quality, indicates how consistent and untampered a set of data is regardless of where and how it is stored. It ensures that data is accurate, reliable and available to authorized parties.

Data sovereignty makes sure that your data is always subject only to the laws of the country which it is located in.

Data privacy is concerned with proper handling, processing, storage and usage of personal data. According to the law, personal data means any information relating to an identified or



identifiable individual; an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number (e.g., social security number) or one or more factors specific to his physical, physiological, mental, economic, cultural or social identity (e.g., name and first name, date of birth, biometrics data, fingerprints, DNA...) (CNIL definition). Ensuring data privacy requires scouting and applying regulation, deploying policies and practices, governing data and third-parties.

CORTEX² has a number of built-in security and privacy mechanisms:

- Users: CORTEX² handles access control to ensure that each user could only access to the data he is authorized to.
- Data governance: The user produces personal data that is transmitted and stored in CORTEX². CORTEX² handles this private data according to European regulation and policies.
- Streams: CORTEX² ensures the cyphering of media and data streams.

Security and privacy development recommendations:

New CORTEX² components or use-cases: the new CORTEX² components will be directly impacted by data privacy and data security when interacting and exchanging data with other components. Third parties must make sure that the new services:

- do not impair GDPR compliancy
- do not impair CORTEX² access control mechanisms
- ensure integrity and security of all data generated and stored either in CORTEX² framework or outside of CORTEX² framework.



2.5. Cortex Framework components available for 3rd-party integration

2.5.1. Planned status of the Cortex Framework by end of February 2024

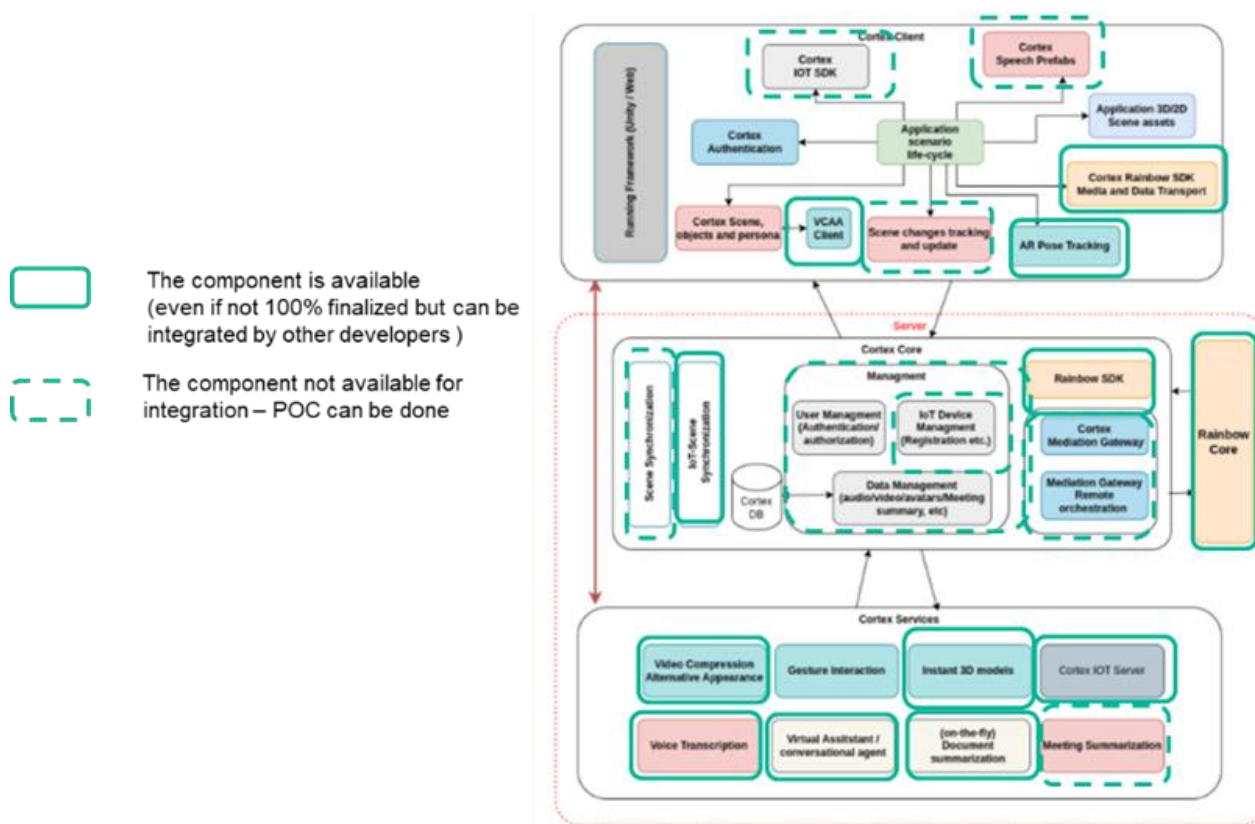


Figure 5: CORTEX² Framework available for OC1 third party integration

- Components and functions
 - Scene synchronisation: use of the Rainbow WebRTC Data channels available in Unity.
 - Rainbow Mediation Gateway: first iteration of orchestration with 3rd-party services. (see below Cortex SDKs)
 - User role management and IoT device management: A keycloak server instance is deployed in Cortex core that can be used by 3rd-parties
 - IoT Scene synchronisation server: This is an interactive bot with uiTOP and it drives the actuation and sensing subscription. It will use keycloak for device management and configuration of access rights to IoT devices
 - VCAA: The component supports client emitter side and client receiver side, available in Unity as prefabs
 - Voice Transcription - CoVA (Cortex Voice Assistant) - partially available: The component supports Speech to Text, Text to Speech, and Intention recognition + document search within a local repository and summarization
 - Meeting Summarization: Partially available for POC.
 - Gesture interaction: Not available (development will start after the OC1).
 - Instant 3D scene reconstruction: A client server architecture will be available with a Unity module as client, and a server service that reconstructs 3D scenes.

2.5.2. Cortex Software Development Kits (SDKs)

CORTEX² does not provide a single Software Development Kit (SDK), but rather a collection of SDKs that can be used for different purposes in the framework. The CORTEX² framework bases on the Rainbow Videoconferencing system by Alcatel Lucent Enterprise, which has a specific SDK called Rainbow SDK. CORTEX² is providing the developers with a set of SDKs going beyond only Rainbow SDK

For the Rainbow SDK please refer to <https://developers.openrainbow.com/>. Please note that the WebRTC data channels and the Mediation Gateway (MGW) may not be fully described in this documentation as their development is not completed when issuing the OC2 but it could already provide a lot of information regarding available features and functionalities. For the documentation of the WebRTC data channels and the MGW, you will get support from the CORTEX2 consortium.

Notice, we are targeting 2 different development environments: Unity 3D or Mozilla Hubs. We will rely in the capabilities of each of those environments that are not described here.

The following presents the different SDKs made available to the developers for the use of the CORTEX² framework:

- **Client-Side SDK**
 - Web SDK:
 - Audio, video conferencing ...
 - C# SDK:
 - Unity package and sample Unity project showing how to set up the Rainbow Controller aiming at easing the developers' integration of Rainbow.
 - IoT SDK : an abstract layer on top of Rainbow SDK , an implementation of Cortex IoT SDK data models. The structure of the messages exchanged will be described.
 - Tracking SDK: A Unity module and accompanying dll library for 3D object recognition and tracking which allows to track an object in the received remote video
 - VCAA service SDK: integration of the VCAA module for being able to change the video transmitted in the Rainbow conversation
- **Server-Side SDK**
 - Rainbow Node SDK (Nodejs – Javascript): it is used for instance for IoT scene synchronisation bot using Instant Messaging to exchanged structured messages (IoT SDK)
 - Cortex² Virtual Assistant (Rasa SDK): CoVA is based on Rasa Open Source (<https://rasa.com/docs/rasa/>) and will later on be completed with a dedicated generative AI-based module. Currently, writing a CoVA assistant consists in implementing a Rasa bot by writing dedicated files as described in <https://rasa.com/docs/rasa/training-data-format>. This is done in a clone of a git repository that will be made available. The agent is then built and deployed using either Docker Compose or Kubernetes.



- **External services integration through Mediation Gateway**

- The MGW component will be partially available by the end of Feb 2024 and will be able to route flows. The XR Cortex service uses the Rainbow C# SDK to receive and exchange WebRTC flows.
- The MGW acts as an orchestrator and, on Service activation request, (Voice Assistant), or according to the scenario, the MGW scripting part will be triggered to activate or route the several WebRTC and chat flows to the right component. Basic scripting capabilities are available and integration use cases can be discussed with applicants.

2.5.3. How to start with developing a Cortex application or a Cortex component

The 3 pilots have instantiated the abstract model described in deliverable D2.1. The Virtual Business meeting use Web technologies and is based on Mozilla Hubs while the 2 other pilots are based on Unity 3D and so use C# programming.

For integrating the contributions developed in the frame of the Open Call 2 the third parties are expected to use or extend the abstract model, and in this case, the 3rd-party should follow the general guidelines of CORTEX² framework development. This will be provided by mentors to guide the 3rd-party partner during the implementation of the project.

How to start developing with and for CORTEX²

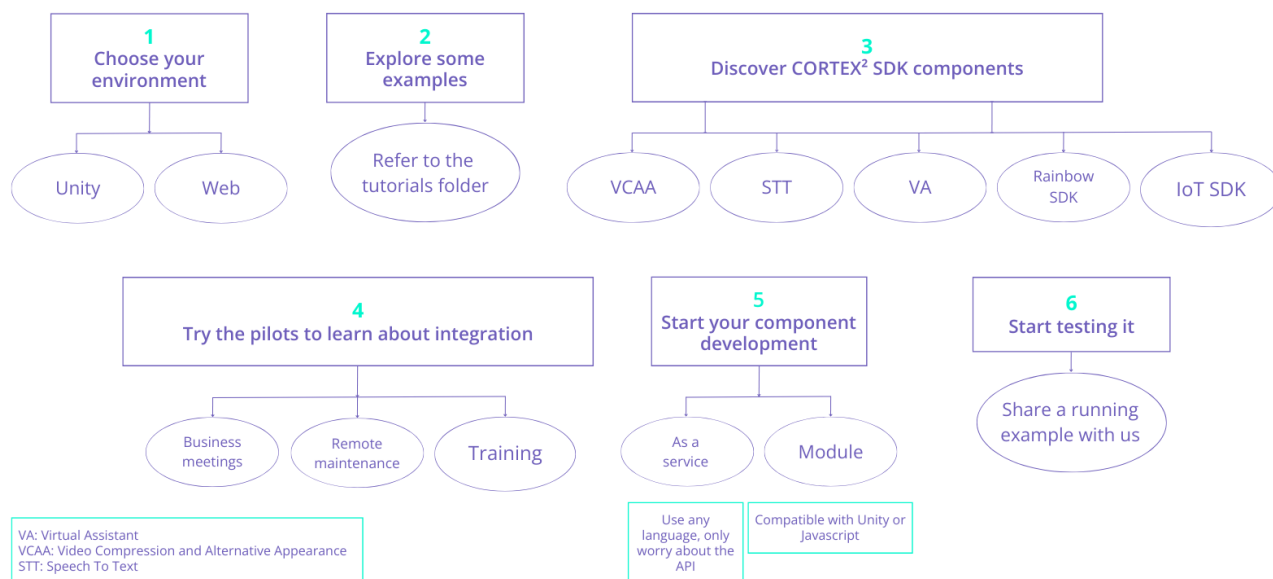


Figure 6: CORTEX² OC1 Co-development process

Documented project samples: To help the 3rd-party partners, several samples highlighting how the consortium integrated the CORTEX² Framework components by using the different SDKs available will be provided.



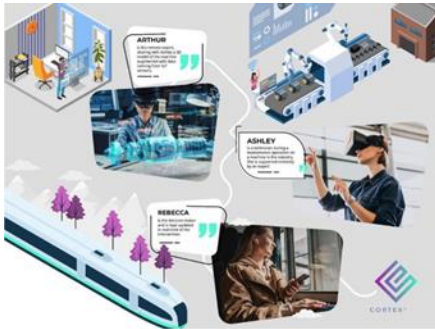
In addition, **3 Pilots MVP** will present what can be achieved thanks to the CORTEX² framework.

3. CORTEX² Pilots

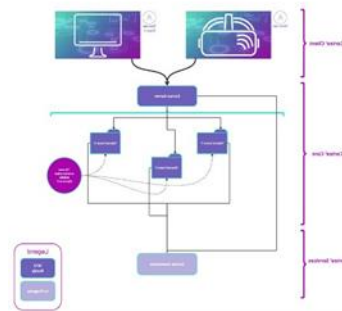
Three use cases, called pilots, are being developed as part of the project. They present the first adopters of CORTEX2 framework, presented above.

3.1. Industrial maintenance

We created the first MVP of a remote maintenance solution involving collaboration through VR and AR technologies.



(a) Scenario of the industrial remote cooperation use case



(b) Use diagram of the industrial remote cooperation use case

Industrial remote cooperation use case

The industrial remote cooperation use-case pilot utilizes Rainbow functionality and the Unity engine to connect different clients.

In a session, the client supports three different personas:

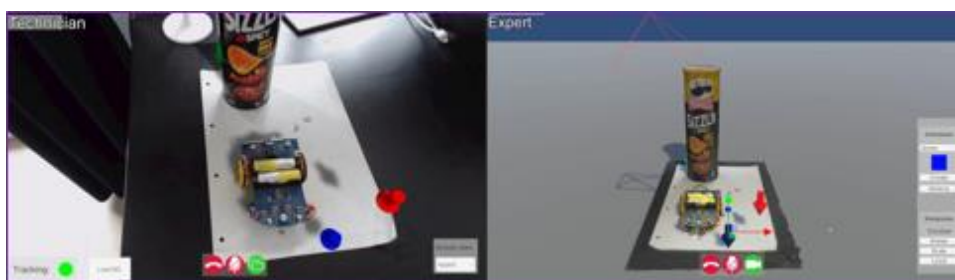
- The technician
 - Requires remote help for complex machinery
 - Uses a device with a camera or an XR headset
- The expert
 - Joins technician and receives 3D model and tracking information
 - Places annotations to aid understanding of tasks
- The Observer
 - Can join a session

As the process of joining and leaving is completely asynchronous, due to the way that bubbles operate and other factors, such as network availability, the pilot utilizes the Rainbow data channel for a variety of communication tasks between the clients.

The technician uses an internal tracking library from the DFKI based on scans of 3D objects. With this, objects can be tracked with a common webcam. The known model can also be used for additional functionality. The technician will see the 3D annotations created by the expert in



the camera image. Due to the knowledge of the 3D model, these annotations can be accurately occluded by the tracked model in the camera image, as well as cast shadow on each other. Additional background 3D objects can be scanned and added to further recreate the environment. This enhances the spatial understanding of the user. The camera feed of the expert can be shown as an augmentation in the camera image or as a small window. Additionally, observers are shown with either their avatars or their camera feed. The expert receives tracking information from the technician and shows both the 3D models as well as the position and orientation of the technician. 3D annotations can be placed and manipulated. If shared, the augmented view of the technician can be shown as a video. Observers are shown the same way as on the technician side. Observers just see roles, participants and shared videos of the expert and technician.



3.2. REMOTE TECHNICAL TRAINING

The Training Platform is developed in Unity and uses OpenXR, an open standard for accessing VR and AR platforms. It consists of a trainee section and a trainer section.

For trainees, the platform is tailored for VR headsets. Initially, trainees enter a solo lobby where they select the course they wish to participate in. Upon selection, they are seamlessly transported into the designated solo experience assigned by their trainer. Within this immersive environment, trainees can communicate orally with all course participants and interact with objects, such as machine models, in real-time. Their actions and interactions are transmitted to the trainer instantly through Rainbow, ensuring seamless communication and feedback.

Trainees also have the flexibility to switch to the multiplayer lobby, where they can interact with their co-trainees represented as basic avatars. This interaction is made possible through Mirror Networking, a networking solution within Unity that facilitates real-time communication and synchronization between multiple instances of the application running on different devices. In the lobby, there is a screen where the trainer can share their screen or camera.

On the trainer side, the platform allows the trainer to remotely manage the training session from their PC. They can communicate with all trainees in their course and have access to a grid-like view of what each trainee sees. This real-time insight into trainee perspectives enables effective guidance and instruction. Moreover, trainers can dynamically modify the solo experiences assigned to each trainee as needed to tailor the training to individual learning objectives and progress.



3.3. Business meeting

Business meeting (BM) is a video conferencing application with extend reality. It is made so that users without VR devices can also join the conference and remain interoperable with large consumer solutions. Currently, users can join the VR conference from the classic Rainbow video-conferencing solution or from Business meeting. People joining in video can choose an alternative appearance, while XR users appear as an avatar in the environment.

CORTEX² provides several features that are partially integrated into the BM, such as the virtual assistant who listens and intervenes in the meeting on various topics, with document and meeting summarization capabilities to preserve conference time. In addition, BM supports interaction with real-world IOT objects to alert the user about an urgent situation or to enable faster actions. The application is also aimed at detecting VR avatar gestures, so they can be perceived by both VR and non-VR participants. Other components are currently being developed in CORTEX, with the aim of making the most of these advances.

4. CORTEX2 Extended features

Through its Open Calls, CORTEX² funds third-party projects to further co-develop and demonstrate CORTEX2 functionalities by integrating additional new use cases. Open Call 1 selected 10 co-development and 10 use-case projects. The current Open Call 2 for co-developers aims to expand CORTEX² functionalities with innovative modules and features not addressed in the first call and to enhance the proposed application areas. Below is a list of topics and applications covered by Open Call 1.

Table 1 List of selected projects under OC1 Track 1 Co-development

#	Selected Project	Brief description
1	Enabling support for externalising models in XR collaboration	Advancing and extending the proprietary markerless motion capture (MoCap) technology of MOV, focusing on supporting the fluent, interactive, and immersive remote collaboration between teams. Apart from the capturing step, the envisioned technology will provide identifiable body representations (i.e. personalised avatars) aiming to foster a sense of presence and to provide a more authentic experience, while facilitating dynamic collaboration through an engaging virtual environment.
2	ARY the AR Media	ARY - an AR media will offer the capability to anchor 3D objects, video, picture, double numeric, pdf file, into indoor environment and make those virtual elements available to anyone using a smartphone or other device.
3	Automatic Minute Taking with	At the core of this project are advanced speech-to-text capabilities and Large Language Models (LLMs). In collaboration with ZA Cloud, the project will leverage knowledge and resources to develop a



	Artificial Intelligence	system that simplifies the meeting process, provides customized and contextually relevant summarization and ensures adaptability across various languages and strict compliance with data security standards.
4	The Infinity Palette	Aims to enrich the Cortex platform with an innovative 2D/3D asset library, optimized for Unity and Mozilla Hubs. A plan to create immersive and adaptable learning environments including a traditional classroom, a group study room, and a library for individual learning, alongside interactive spaces for virtual concerts and cultural exhibitions. These environments, comprising a blend of static and dynamic assets, are designed to be customizable to user needs.
5	Revolutionizing Virtual Spaces: SENSO3D's Comprehensive 3D Object Library	The project envisions the creation of detailed and accurate 3D models for extended reality (XR) applications, with a particular focus on areas such as elder care, language learning, and interactive education. By converting 2D images into immersive 3D environments, SENSO3D enhances visualization and interaction, offering substantial benefits to users, including those with special needs.
6	Co-development of a Dynamic library of personalized gestures	Creating a module capable of accurately capturing and interpreting a wide range of hand gestures. These gestures, once recorded, can be associated with specific semantic interpretations and actions to be used across various AR/VR applications in the CORTEX2 platform.
7	Enhancing Videoconferences with Real-Time IoT Data in agrifood sector	The project targets the integration of advanced IoT technologies within the CORTEX2 framework, focusing on the agri-food industry. Goal is to advance the operational efficiency and user experience in XR applications within this sector through transforming the current fragmented landscape into a more cohesive, efficient, and technologically advanced system.
8	Multiplayer Haptic interactions	Embarking on the development of a multiplayer toolkit, designed to empower XR developers in seamlessly creating interactive virtual environments featuring haptic gloves and hand tracking.
9	Virtualization Service for Object Reconstruction	Proposes a web service that will take images or a video stream of a small object and generate a digital-twin as a triangular mesh, that can be used by all current XR applications and game engines and



		be visualised on any device, enabling easy sharing of the 3D model across multiple stakeholders and environments.
10	Realistic Avatars for XR	Developing a scalable, automatic, integrated tool for realistic, customisable, interoperable, multimodal Avatars that will be integrated with the CORTEX ² technological ecosystem to extend its capabilities by covering User Representation as well as User Avatar Customization.

5. Concluding remarks

In conclusion, this document serves as an informative resource detailing the architecture of the CORTEX² teleconference platform. It complements Annex 1 CORTEX² Guidelines for applicants by offering a comprehensive understanding of the fundamental platform functionalities. It is important to note that this document is a complementary resource, enhancing the understanding provided by Annex 1 rather than existing as a standalone document.

