ACCORDION

Adaptive edge/cloud compute and network continuum over a heterogeneous sparse edge infrastructure to support nextgen applications

DRAFT

Deliverable D6.1

Pilot Plans (I)

HORIZON 2020

ACCORDION receives funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 871793

DOCUMENT INFORMATION

PROJECT	
PROJECT ACRONYM	ACCORDION
PROJECT FULL NAME	Adaptive edge/cloud compute and network continuum over a heterogeneous sparse edge infrastructure to support nextgen applications
STARTING DATE	01/01/2020 (36 months)
ENDING DATE	31/12/2022
PROJECT WEBSITE	http://www.accordion-project.eu/
TOPIC	ICT-15-2019-2020 Cloud Computing
GRANT AGREEMENT N.	871793
COORDINATOR	CNR
DELIVERABLE INFORMATION	
WORKPACKAGE N. | TITLE	WP 6 Integration, Pilot Implementation & Evaluation
WORKPACKAGE LEADER	OVR
DELIVERABLE N. | TITLE	D6.1 Pilot Plans
EDITOR	Maria Pateraki (OVR)
CONTRIBUTOR(S)	Michael Dodis (OVR), Zbyszek Ledwon (ORBK), Przemysław Tarkowski (ORBK), Tomas Loven (PLEX), Emanuele Carlini (CNR), Saman Zadtootaghaj (TUB), Patrizio Dazzi (CNR)
REVIEWER	Lorenzo Blasi (HPE)
CONTRACTUAL DELIVERY DATE	Original: 31 December 2020, Amended: 31 January 2021
ACTUAL DELIVERY DATE	31 January 2021
VERSION	1.0
TYPE	Report
DISSEMINATION LEVEL	Public
TOTAL N. PAGES	46
KEYWORDS	Pilot plans, evaluation, integration strategy

EXECUTIVE SUMMARY

This deliverable provides a first release of the report on the plans to implement and evaluate the scenarios addressed for each project Use Case. The deliverable comprises and reports on tree main subtasks. The first subtask relates to a detailed description of the Use Cases and the scenarios under which evaluation which revolve. The second subtask relates to the description of the pilot prototypes, the evaluation methodology, the experimentation requirements, the modules/functionalities that will be assessed and the metrics for assessing the value of ACCORDION in terms of technology and subjective Quality of Experience. The third subtask specifies the integration plan for the components and technologies of ACCORDION, the design of the infrastructure along with the testbed combinations for pilot execution and evaluation and the execution methodology. Regarding ethical and privacy issues, all necessary measures have been considered as part of conducting the subjective Quality of experience evaluation and are reported.

DISCLAIMER

ACCORDION (871793) is a H2020 ICT project funded by the European Commission.

ACCORDION establishes an opportunistic approach in bringing together edge resource/infrastructures (public clouds, on-premise infrastructures, telco resources, even end-devices) in pools defined in terms of latency, that can support NextGen application requirements. To mitigate the expectation that these pools will be "sparse", providing low availability guarantees, ACCORDION will intelligently orchestrate the compute & network continuum formed between edge and public clouds, using the latter as a capacitor. Deployment decisions will be taken also based on privacy, security, cost, time and resource type criteria.

This document contains information on ACCORDION core activities. Any reference to content in this document should clearly indicate the authors, source, organisation and publication date.

The document has been produced with the funding of the European Commission. The content of this publication is the sole responsibility of the ACCORDION Consortium and its experts, and it cannot be considered to reflect the views of the European Commission. The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

The European Union (EU) was established in accordance with the Treaty on the European Union (Maastricht). There are currently 27 members states of the European Union. It is based on the European Communities and the member states' cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice, and the Court of Auditors (http://europa.eu.int/).

REVISION HISTORY LOG

VERSION No.	DATE	AUTHOR(S)	SUMMARY OF CHANGES
0.1	05/12/2021	Maria Pateraki (OVR)	Table of Contents
0.2	24/12/2021	Michael Dodis (OVR), Zbyszek Ledwon (ORBK), Przemysław Tarkowski (ORBK), Thomas Loven (PLEX), Emanuele Carlini (CNR), Saman Zadtootaghaj (TUB)	First round of contributions
0.21	14/01/2021	Michael Dodis (OVR), Zbyszek Ledwon (ORBK), Przemysław Tarkowski (ORBK), Thomas Loven (PLEX), Emanuele Carlini (CNR)	Second round of contributions based on updates in D2.3
0.22	26/01/2021	Zbyszek Ledwon (ORBK), Przemysław Tarkowski (ORBK), Thomas Loven (PLEX), Emanuele Carlini (CNR)	Third round of contributions
0.23	29/01/2021	Lorenzo Blasi (HPE)	Review comments received
0.3	31/01/2021	Maria Pateraki (OVR)	Consolidated version
1.0	31/01/2021	Patrizio Dazzi (CNR)	Proofreading

GLOSSARY

AR	Augmented Reality
CCU	Concurrent Users
EU	European Union
EC	European Commission
GPU	Graphics Processing Unit
H2020	Horizon 2020 EU Framework Programme for Research and Innovation
LS	Local Service
QoE	Quality of Experience
QoS	Quality of Service
RTT	Round-Trip-Time
VR	Virtual Reality
WP	Work Package

TABLE OF CONTENTS

1 Relevance to ACCORDION

1.1 Purpose of this document

This document presents the first version of the pilot plans, reporting on the intermediate outcomes of *Task 6.2 "Pilot planning".* The report provides a detailed description of the three Use Cases of the project, the rationale to support the transitioning and exploitation of the ACCORDION platform and the main challenges that are addressed for each Use Case. The report details the scenarios, under which the evaluation and pilot trials will revolve for each Use Case, along with their execution workflows. It also provides an initial description of the pilot prototypes, the evaluation methodology, the experimentation requirements and planning related aspects, the modules/functionalities that will be assessed and the metrics for assessing the value of ACCORDION in terms of technology and subjective Quality of Experience. As part of the document the integration plan of ACCORDION, the design of the infrastructure along with the testbed combinations for pilot execution and evaluation and the execution methodology are described. Regarding ethical and privacy issues, the necessary aspects as part of conducting the subjective QoE evaluation are reported.

1.2 Relevance to project objectives

This deliverable links to all the following Objectives as it addresses the evaluation of modules, technologies, overall ACCORDION platform to support the Use Case scenarios:

- Obj1: Maximize edge resource pool
- Obj2: Maximize robustness of cloud/edge compute continuum
- Obj3: Minimize overheads in migrating applications to cloud/edge federations
- Obj4: Realize NextGen application

1.3 Relation to other work packages

The deliverable consumes output from WP2 and therefore is related to all deliverables of *"WP2 Requirements & System Design"*, primarily those reflecting the User requirements (D2.1, D2.5) and Architecture design (D2.3, D2.7). The deliverable provides the basis for the upcoming deliverables of WP6 and provides output to all research Tasks of WP3-WP5 as a preparatory stage for the integration and evaluation phases.

1.4 Structure of the document

Section 2 defines the main concepts addressed in the deliverable to support the reader with a clear definition and distinction between the notion of Use Case, scenario, pilot prototype and testbed.

Section 3 provides a detailed description of the Use Case, the rational for exploiting the ACCORDION platform and the main challenges that arise. It further describes the scenarios along with their workflows.

Section 4 provides a first description of the pilot prototypes, evaluation, experimentation and planning related aspects, the modules/functionalities and a preliminary list of metrics/methods for assessing the technology and subjective QoE.

Section 5 describes the infrastructure for Use Case validation, the integration plan and the execution methodology.

2 Main concepts

Use Case: Although the term Use Case in software and systems engineering mainly refers to a list of actions typically defining the interactions between a role (actor) and a system to achieve a goal we approach the term of Use Case more from the business perspective. Thus, the Use Case justifies for pursuing a course of action in a given context to meet certain objectives or goals. In this respect, the above context addresses NextGen applications in specific domains, i.e. collaborative VR, multi-player mobile gaming and QoE optimization in content delivery, that aim to validate the ACCORDION concept. Those organizations that develop NextGen applications are considered as the prime end-users of the project.

Scenario: a description of how the system will carry out a specified process to fulfil the stated requirements. In scenarios process flow diagrams maybe used to enumerate graphically the steps that will take place in each process. The scenarios may also be considered business Use Case instances. As part of the project different functional and non-functional requirements have been initially defined in D2.1 related to the platform core functionalities and related to the different Use Cases.

Pilot prototype: a prototype is an early sample, model or release of a product built to test a concept or process. It mainly serves to verify functional aspects of the intended operation, including assumptions of the underlying architecture mechanisms to support experimentation, validation and further refinements. Pilot prototypes are instances of a system that accommodates the needs of the Use Cases and integrates platform functionalities.

Testbed: The testbed provides a realistic hardware-software environment with which to test components without having the ultimate system. The testbed provides a means to improve the understanding of the functional requirements and operational behaviour of the system. For the purposes of ACCORDION, the testbed is the environment composed of hardware and computing resources (i.e., the infrastructure) that enables experimentation.

3 The ACCORDION Use Cases

3.1 High level impact

There is a growing demand, in highly dynamic time-sensitive applications, for both real-time computation and low latency at the end device, to deliver high quality of experience. SMEs developing NextGen applications (i.e., gaming, VR/AR) face growing distribution pressures from the increased data volumes and expanding bandwidth, needed for mobile-centric, lower latency applications that could avoid the limitations imposed by remote data centers or clouds. ACCORDION supports this vision by allowing edge resources and infrastructures to contribute in the seamless experience in a device-agnostic way, across multiple sites, supporting an increasing number of Concurrent Users (CCUs). For the collaborative VR Use Case (sec. 3.3) ACCORDION leverages the compute and network stack at multiple layers -device, network and mini-cloud - to support the untethered multi-user VR experience beyond the inherent hardware limitations. In the multiplayer mobile game Use Case (sec. 3.4) the ACCORDION platform aims to support the requirements to lower latency between servers and clients and improve user experience, deploying new servers based on performance metrics and player's geographical localization. For the Use Case supporting QoE optimization for content delivery (sec. 3.5) two cloud-based products of PLEX will be integrated to exploit mini-cloud resources via the ACCORDION platform for providing personalized content to serve advertisement purposes in public spaces and augmented reality game on users mobile devices, ensuring low latency and higher performance on average network conditions.

3.2 Main actors

As part of D2.1 (User requirements) different roles were defined that relate to the individual applications that the three Use Cases bring into the project, as well as the ACCORDION platform and infrastructure owner. These roles have also been described in D2.3 (Architecture design) as part of three main classes mainly referring to the individual business entities, namely Infrastructure owner, application provider and ACCORDION Platform owner along with a human actor or an agent that acts on behalf of the business entity. Focusing primarily on the scope of this deliverable and the description of the Use Cases and scenarios of the first Cycle the roles described below focus on the application/product that OVR, ORBK, PLEX bring in and which will be supported by the ACCORDION platform. The core assumption is that the ACCORDION platform functionalities, such as infrastructure and minicloud management, indexing and discovery, creation of network paths, compute and network resource orchestration, failure detection and mitigation, etc. are agnostic to the application provider and its agents.

The three identified actors refer to both the business entity and agents (person) supporting different aspects of the application/products, their responsibilities and benefits arising from their role.

User: In the majority of the Use Case scenarios introduced in this document, the user is the consumer of the service. In the ACCORDION context, the service can be considered as a) the active participation via untethered HMDs in a collaborative VR environment for medical training; b) the downloading and playing of a game on a mobile device with minimal server response time and uninterrupted gameplay experience; and c) the optimization of the user experience for an AR game that is installed and run on a mobile device and the personalization of the content (digital signage). As part of the scenario descriptions, we do not consider the user as a producer of the actual content, but only as a consumer.

Application Developer: In this category one may consider those persons of the business entity that develop, combine, integrate and customise functions of the application/product to create the application components to be offered to the users. Developers must take into account the specificities and requirements of the ACCORDION Platform owners, such as the interfaces for communication between the ACCORDION and the application components and the packaging of the application component in a described format to allow the deployment of the application. Furthermore, to support the application model functionalities, and to facilitate the migration of the application to the ACCORDION platform. Finally, it is highlighted that the developers will take advantage of the benefits provided by the ACCORDION platform for dynamically managing application components, offloading and deploying certain components to available resources belonging to Edge miniclouds or public central clouds, thus supporting dynamic scaling of resources addressing challenges in computing capacity of end-devices, and support more users and geographies.

Application provider: We assume the business entity or company that develops latency-sensitive applications (i.e. Collaborative VR, mobile gaming, AR) that aims to take full advantage of the resources across the cloud-edge continuum.

Below these roles are further described in Table 1 in the scope of each individual Use Case.

Table 1: Defined Roles and relevant description for each Use Case.

Role/Actor	Description
User	**UC#1 (OVR):** The person to whom the service and products of OVR are offered. Beyond the single person it can also be a customer entity such as a University or other company which provides the service/ product to members of its team. The user(s) with an untethered HMD and will be able to perform in the VR environment medical training in collaboration with other users. **UC#2 (ORBK):** The person who will install and use UC2 mobile game. In order to install the UC2 mobile game one needs a modern Android mobile device with internet access. UC2 mobile game will be available in Google Play Store. Once the game is installed on the user's device it will be ready to be used without any charge. **UC#3 (PLEX):** Person on edge device range with a mobile device in which the AR game application can be installed.
Developer	**UC#1 (OVR):** The person responsible for developing the content, authoring and training module as part of the application. **UC#2 (ORBK):** The person or company that will create a Game. Game Developer must create both client part (mobile game) and server part (game server) of the Game. **UC#3 (PLEX):** The agent responsible for the development and maintenance of the services connected to ACCORDION.
Application provider	**UC#1 (OVR):** The business entity that provides the solution (products) to users/customers for medical training purposes. **UC#2 (ORBK):** The business entity that provides the game application using the ACCORDION resources to the users and is the application administrator.

	UC#3 (PLEX): The business entity that provides the solution (products) to users/customers for marketing purposes.

3.3 Use Case#1: Collaborative VR

3.3.1 Application description, rationale and challenges

OVR is a deep technology startup, innovating in the field of experiential simulation for medical training with VR. OVR provides a gamified multi-user VR platform built on Unity engine that exploits the MAGES SDK[1] on top of a networking layer by Unity[2]. The MAGES SDK handles and synchronizes in-game interactions, deformable object transformations and physics simulation, broadcasts transformation values over the network while as part of the MAGES SDK the custom Geometric Algebra interpolation engine supports interpolation of in-between positions/rotations locally at each end-device (HMD). Currently, for the case of untethered HMDs the storage, rendering, compression are all local processes as part of a single application component that is installed and run on the untethered HMD.

The transitioning to the ACCORDION platform will support the OVR business case to develop and promote collaborative cloud VR training applications specially formulated for untethered HMDs and the adaptation of OVR's networking layer to edge computing will optimize the current status of the cooperative mode, ensuring lower latency and higher performance on average network conditions and ultimately a higher number of CCUs.

In order to enable this transitioning and leverage the edge-cloud architecture for the given application, computation offloading has to be exploited to support migrating part of its computing in edge-cloud resources, requiring interactions and data exchanges between the different modules placed on device, edge and cloud, considering the benefit of remote execution, the cost of data transmission and the complexity of application partitioning. Still for remote execution and streaming has to consider the location of edge nodes in relation to the end device to avoid high latency in packets delivery between the network nodes. As part of the computation offloading paradigm, and important for collaborative multi-player environments, is a networking solution that effectively manages communication among clients and handles the game state with less dependency on the master server. Offloading this functionality from the client-host and providing it as a data service from ACCORDION to the application can be realized via a relay server that is able to handle beyond the broadcasting of messages, also host migration and solve for game-state continuity.

[1] Papagiannakis G., Zikas P., Lydatakis N., Kateros S., Kentros M., Geronikolakis E., Kamarianakis M., Kartsonaki I. and Evangelou G. MAGES 3.0: Tying the knot of medical VR. ACM SIGGRAPH 2020, August 2020, Article No: 6, Pages 1-2.

[2] https://unity.com/

The ACCORDION platform may support a) discovery of edge nodes for remote processing and streaming to the HMD, b) hosting of the relay server for message brokering, session management and host migration and c) application updates.

The main challenges, which also relate to the requirements set in D2.1 are:

- The Unity engine and the application should tolerate application partitioning, further distinguishing the modules that require specific GPU hardware from those that can accommodate any CPU configuration.
- Provide the VR service to the user utilizing the necessary capabilities of the infrastructure (GPUs, CUDA [3]acceleration) enabling secure access, connectivity and proximity to support the service requirements (e.g., latency of <20 ms)
- Manage the discovery and provisioning of the cloud-edge resources transparently to the application with no additional effort for the application developer.
- The interface to access the infrastructure should be standardized for the application.
- Connectivity towards the cloud-edge resources to be seamlessly managed.
- Ensure low-latency data exchange between multiple users through the relay server. Consider the proximity of the relay server in relation to the footprints of the users
- Ensure the commercial viability

3.3.2 Scenarios

As mentioned in the previous paragraph to enable the application to take advantage of the services provided by the ACCORDION platform the application is redesigned to enable the migration of certain functions/application components in the ACCORDION minicloud. For offloading the "full" computation offloading option is favored, where the whole computing part of the application is offloaded leaving the mobile device only responsible for UI, input/output. This approach is preferable against task offloading, namely offloading a specific task, such as rendering as the modules related to rendering are rarely independent. Inter-calls among modules from different sub-families and call frequency make the code offloading harder. Figure 1 depicts the proposed "full" offloading approach for a single user. The HMD is responsible for sending rotation/translation and events as well as performing the decoding of the rendered image before being displayed on the HMD. The computation node in the edge minicloud, also characterized as Local Service (LS) supports all related-processing from the MAGES SDK and the Unity game engine, also storing the VR scene, data assets and avatars. The LS can be further shared by multiple users, where multiple instances of rendering, encoding can be initiated supporting the different HMDs (Figure 2). Although the application on the HMD is built for an ARM-based processor (.apk) the streaming of data between application components is agnostic to the device API (i.e. Oculus SDK) eliminating dependencies on proprietary API's linked to specific HMD vendors. The LS is also able to connect to ORama Cloud Services for retaining user-

[3] CUDA® is a parallel computing platform and programming model that enables increases in computing performance by harnessing the power of the graphics processing unit (GPU). https://developer.nvidia.com/cuda-faq

specific data (analytics). Communication between the HMDs and LS can be realized via standard UDP connection. Additional modules may assess the network characteristics and dynamically provide adaptations to the send rate from the HMD(s) based on the QoE model. These modules may reside on nearby edge nodes of the minicloud or on the same edge node. Aiming also to exploit the benefit of relay severs to support multicasting, dynamic host migration and game state continuity when multiple user collaborate under the same session in the same virtual environment, the ACCORDION platform selects the appropriate resource to deploy the service based on the users locations footprints. Additional third-party software is exploited to support the re-design and validation of the scenarios as listed below.

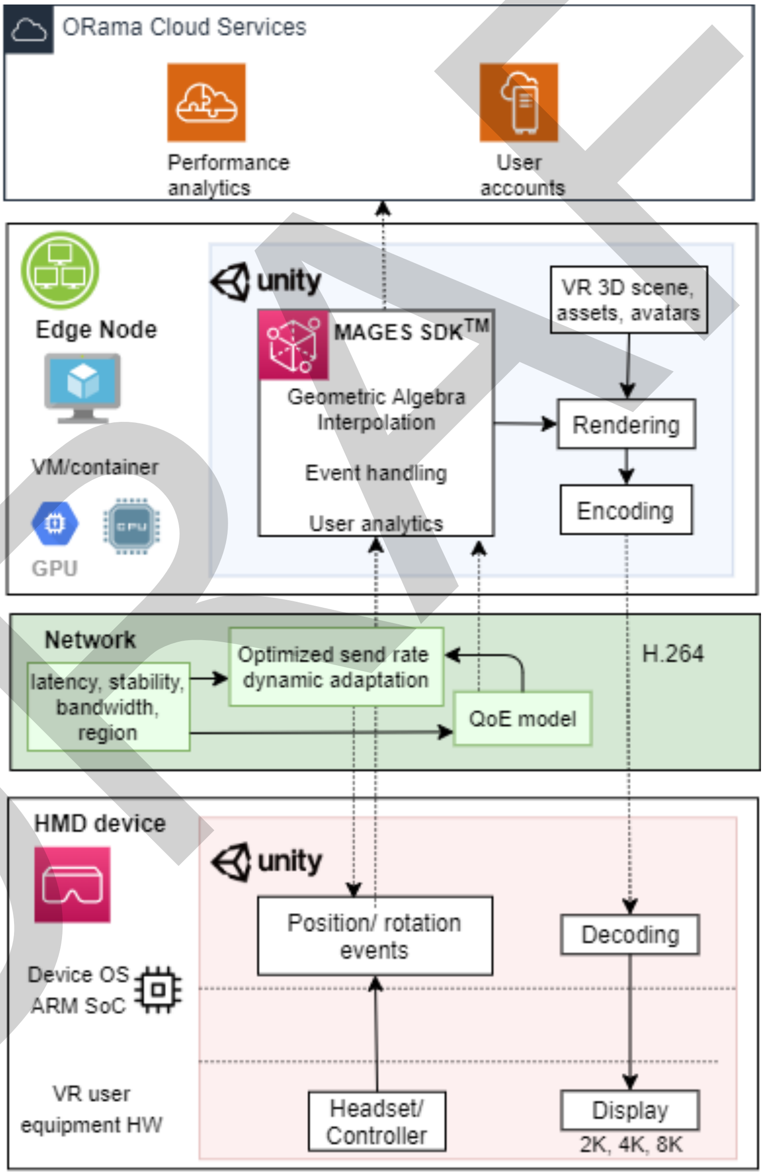

Figure 1: Full computation offloading approach for a single user in OVR application

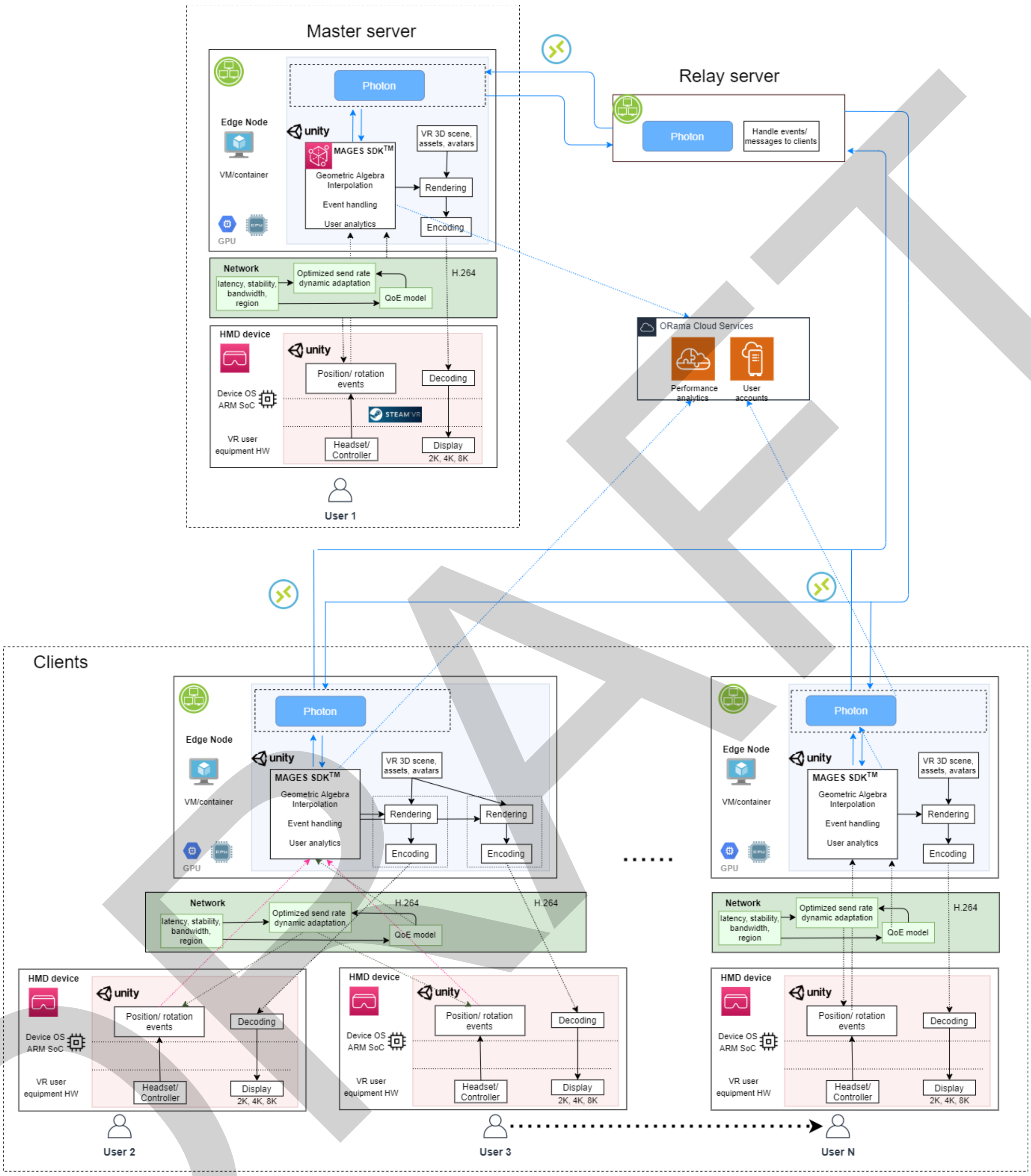

Figure 2: Full computation offloading approach in OVR application – Computation shared by different nodes

Third party software

Photon: Cross-platform networking solution used in the application, utilizing a mix of common low-latency network communication methods (peer-to-peer with mandatory relay server connection) and providing the required integration in the Unity Engine.

WebRTC: Cross-platform framework that enables real-time communication (RTC) for mobile devices and browsers. Establishment of the communication of the participants is left up to the user of the framework, via a signaling server. The participants exchange information for audio and video transportation and playback (SDP protocol), as well as the optimal communication path (using stun and turn servers). Custom data streaming and messaging through the established peer-to-peer communication is also supported. The commonly used library is provided by Google, through the Chromium project.

WebRTC for Unity: Implementation of the WebRTC protocol by the Unity team; supports efficient streaming of rendition results as video playback to recipients, as well as audio processing results and input (via a custom data stream).

Four main scenarios are supported:

- **Scenario 0: Submission of the application,** where the application developer enters the ACCORDION Platform portal and submits the required data to the ACCORDION platform to enable the further deployment of the application. This scenario is horizontal for all subsequent Use Cases.

- **Scenario 1.1: Start application,** where the user starts the application from the HMD, communicates to the event bus the launch command to the ACCORDION platform and the application image is deployed on the Local Service (LS). The application exploits the open source cross-platform WebRTC for Unity package for streaming between end-points. The WebRTC resides on a signaling solution involving a third-party server in addition to the two peers trying to connect to each other. Using a third-party server may seem counter-intuitive at first when dealing with peer-to-peer connection, but in general the third-party server is an easy-to-reach server which acts as a relay and enables WebRTC to discover a direct route between the two peers even in complex network scenarios (one or both peers behind a NAT) where it would otherwise be impossible for the two peers to directly discover each other.

- **Scenario 1.2: Create session**, the user accesses the interface for session list on the HMD and requests through the ACCORDION platform the Relay server address. The ACCORDION takes care of the necessary deployment steps of the Photon relay server instance on the available minicloud resource selected by the platform. Information is sent back to the LS and when connection is directly established with the relay server instance the user retrieves the list of active sessions and requests to create a new session in which other users may connect to.

- **Scenario 1.3: Runtime adaptation of the application**, QoE degradations are detected based on the QoE model of the platform and communicated to the LS to adapt the synchronization interval between the HMD device and the LS. This scenario will be supported as part of Cycle II validation phase.

Prerequisites for all scenarios:

- Communication to 5G or Wi-Fi is established via the HMD
- The channel for communication with ACCORDION platform should be known to the application running on the HMD. Following the publish subscribe model of the event bus the application components may send and receive messages to/from the event bus via subscription to specific topics.
- The channel for communication between application components, managed and deployed by the ACCORDION platform. For direct app-to-app communication the UDP protocol is used.

Scenario 0: Submission of the application

1. Application developer enters the ACCORDION Platform portal.
2. Application developer selects and enters required data on website (resource requirements, QoE requirements, lifecycle management instructions, docker image of application components).
3. Application developer sends that data to the ACCORDION platform through the ACCORDION Platform portal.
4. Application developer receives response from the ACCORDION Platform portal with deployment sta tus result.

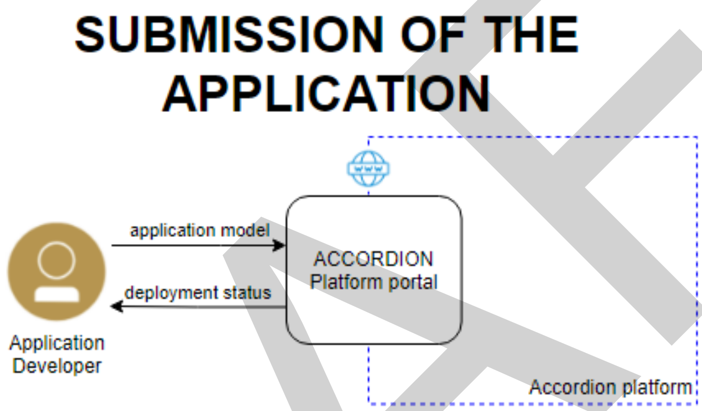

Figure 3 Submission of the application – UC#1, UC#2, UC#3

Scenario 1.1: Start application

1. The user logs in using the OVR account (i.e., authenticates in an external server – OramaVR cloud services) to access the installed applications on the HMD, selects and starts the application. A secure connection to the event bus is established and a message is sent to a specific topic with the HMD parameters and a request to start the application on the minicloud. The HMD parameters that are communicated are HMD model, screen resolution, refresh rate, IP, framerate, bitrate, memory, CPU frequency, userID.
2. The ACCORDION orchestrator is subscribed to a different number of topics from the event bus and receives the messages transmitted from one or more HMDs.
3. Upon reception of the message the orchestrator sends an application resource status message to the application status registry component (i.e. the component that keeps track about the resource in which the services of the application are deployed /running and maintains the application status).
4. The platform: a) detects a nearby to the HMD edge resource (Local Service – LS), of the minicloud creates the container of the partitioned application component on it and loads the image in the container (if it doesn't exist), thus creating an LS container instance; b) detects another resource on the minicloud to deploy the signaling server (WebRTC).
5. The signaling server IP is communicated to the LS container instance and connection is established between the LS container instance and the signaling server (app-to-app communication).
6. The signaling server IP and the LS IP are maintained in the application status registry.

7. A message is published from the ACCORDION platform via the event bus in a specific topic, to which the HMD is subscribed. The message contains the two IPs, enabling the HMD to establish connection to the signaling server, which is initialized via TCP.
8. The user accepts to connect to the signaling server and the LS and communication between the HMD and LS is established and verified (HMD-LS handshake)
9. The LS streams the rendered image to the HMD, the image is decoded and displayed on the HMD – the user enters the VR environment. UDP is used for the real-time communication.

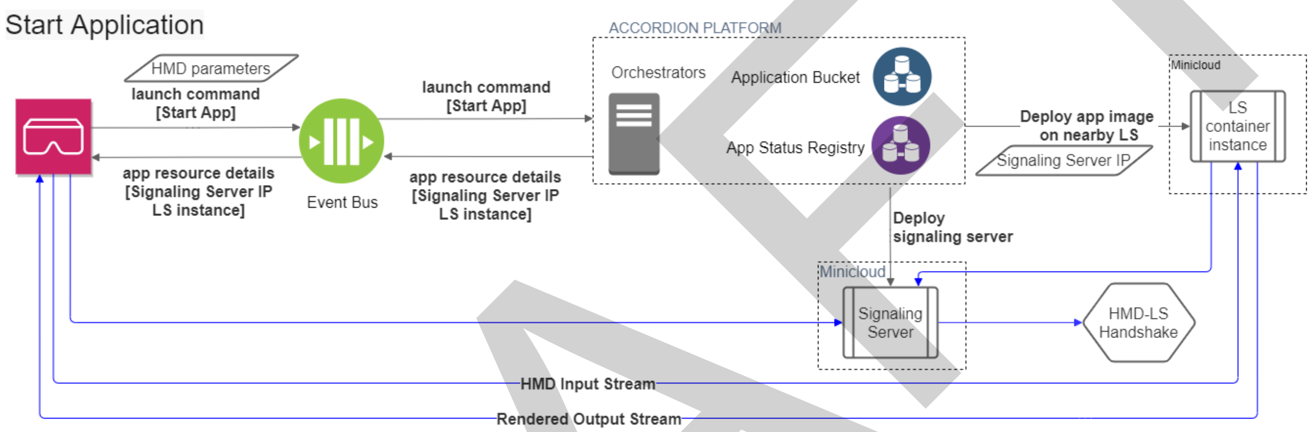

Figure 4: Scenario Start application – UC#1

Scenario 1.2: User creates a session

1. The user (#1) needs to access the interface for session list on the relay server. The LS posts a message to the event bus and sends a message with request for the Relay server address.
2. The ACCORDION platform retrieves from the specific topic the message for the relay server request.
3. Upon reception the ACCORDION orchestrator sends an application resource status message to the application status registry component.
4. The platform detects a minicloud resource, allocates and deploys the Photon relay server instance.
5. The IP of the relay server instance is maintained in the application status registry.
6. A message with the IP of the relay server is sent from the ACCORDION platform to the LS via the event bus.
7. The LS sends a request for a normal Photon connection to the relay server instance. When connection is established the LS retrieves the list with active sessions.
8. User #1 receives the information on the HMD and requests to create a new session. A new session is created and included in the list of active sessions. The list of active sessions is maintained on the relay server.
9. Another User (#2) performs steps 1-3.
10. Step 6 from the ACCORDION platform to the LS of User #2.
11. Step 7 is performed for User #2.
12. User #2 selects the session created by user#1
13. The Photon relay server registers the relevant information on user #2.

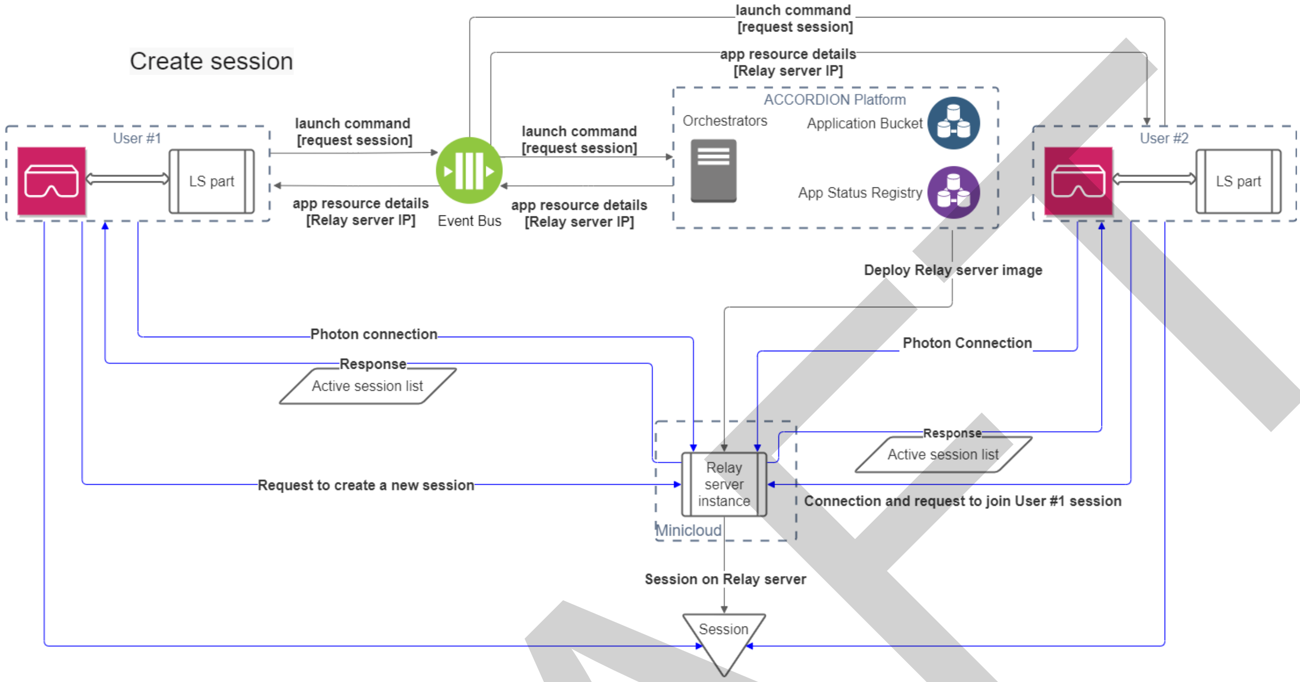

Figure 5: Scenario create session – UC#1

Scenario 1.3. Run-time adaptation of the application (Cycle II)

1. The QoE model of the ACCORDION platform estimates network degradation and posts a message to the event bus.
2. The ACCORDION platform retrieves the message and the component for dynamic send rate adaptation is invoked on the LS (Geometric Algebra interpolation) which in turn communicates necessary information to the HMD device to adapt the send rate in positions/rotations.
3. The step variable on the HMD is changed.
4. The processing is done from this point onwards based on the new rate.

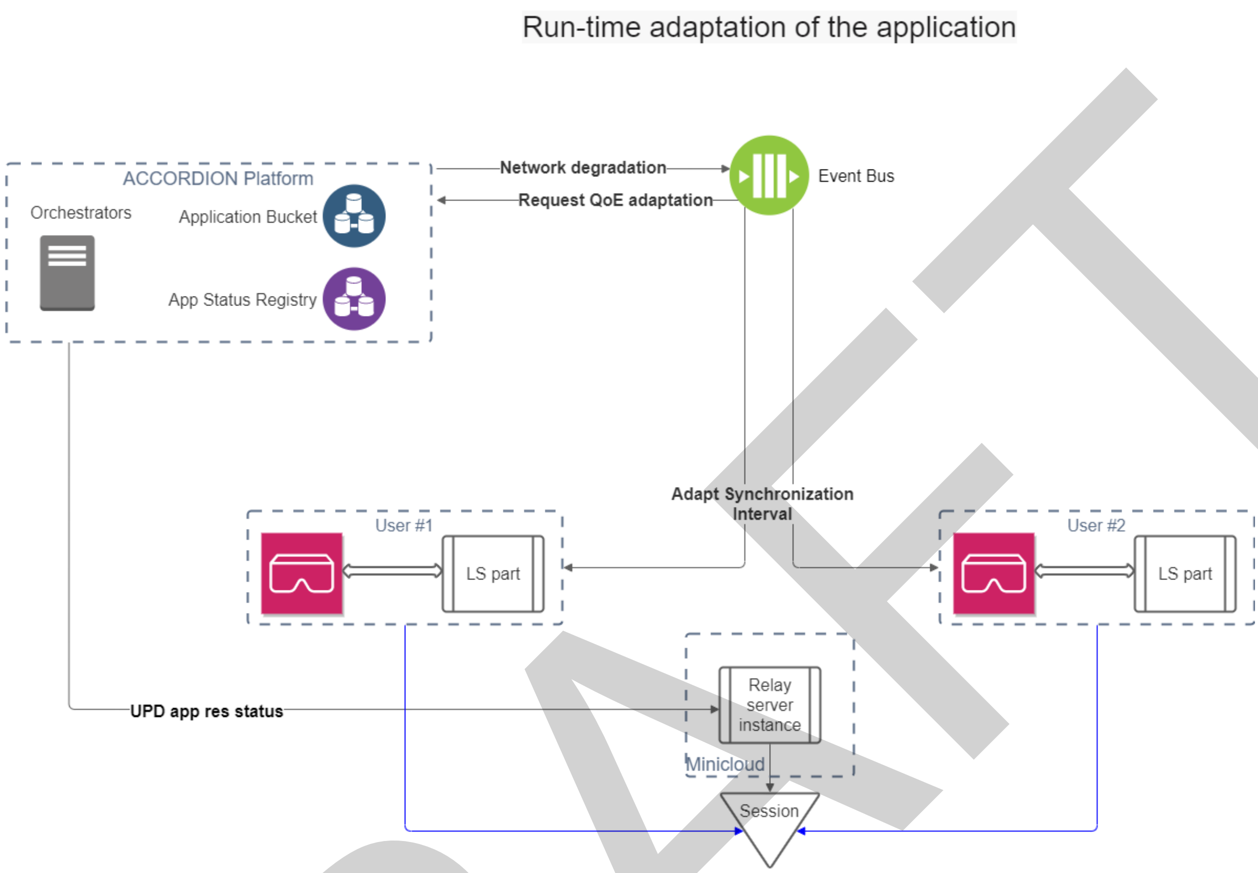

Figure 6: Scenario runtime adaptation – UC#1

3.3.3 Privacy/security aspects

Owners of OVR products are associated with user accounts, distinguishable on a role-based model, i.e. admins, supervisors, and members. The accounts are part of the internal ORamaVR cloud services. Sensitive information of user accounts is hashed using the SHA256 algorithm and traffic is encrypted using the AES-256 specification. The OVR Analytics engine uses a cloud-based user assessment service to track, monitor and present important feedback regarding each gamified operation. These data are stored in the Azure Blob storage system, utilizing its security, and are accessible in the form of json files through the analytics API. The later conforms with the REST service design.

3.4 Use Case#2: Multi-player Mobile Gaming

3.4.1 Application description and challenges

ORBK Use Case is a multiplayer mobile game. Game servers will be deployed on top of the ACCORDION system to meet the requirements of NextGen mobile gaming, which aims to lower latency between servers and clients and highly improve user experience. It will also take advantage of AI-based network orchestration to dynamically and automatically deploy new servers based on performance metrics and player's geographical localization.

The main challenges relating also to the requirements set in D2.1 are:

- Reaching low Round-trip-time (RTT) latency in order to improve user experience. The target value is to achieve average latency ≤ 30ms.
- Number of simultaneously connected players >= 100, by distributing users between multiple game servers, run on intelligently orchestrated minicloud resources.
- Providing a good gameplay experience when players have bad latency, by introducing advanced prediction mechanisms.
- Intelligent automatic game servers deployment, realized by Accordion Orchestrators components based on the IP addresses of the connected clients (geolocation)
- In the second phase of the ACCORDION project, we are aiming to introduce mechanism to allow players to dynamically move between game servers

3.4.2 Scenarios

Network latencies in turn-based multiplayer games can reach values up to 500 milliseconds and user experience will remain acceptable, whereas in real-time multiplayer games 100 milliseconds (with client-side prediction) and 10-30 milliseconds (without client-side prediction) is the upper limit for network latency. Therefore, any technology that will allow us to achieve these lower latency limits will be a milestone allowing more developers to enter the real-time multiplayer games market.

The ORBK game system will consist of two main elements: Game Server and Mobile Application. Game Servers will be deployed at the Accordion infrastructure. Mobile Application will be run by the end-users – players, on their own mobile devices. The game system must be able to handle up to 100 players, handle a huge number of in-game events while performing full simulation of the game world and generate responses with minimal possible delay. The ACCORDION platform must optimally scale its resources in order to match the number of current players. This means that adding and removing resources must be fully automatic. This aspect of the system is very important from an economic point of view.

Two main scenarios are supported aside the submission to the application scenario (Scenario 0 – sec. 3.3.2)

- **Scenario 2.1: Start application,** when Application Developer enters the ACCORDION Platform portal he is presented with the option to start all necessary processes in order to make the application up and running. The application Developer would be able to modify and save application model data before clicking the start application button.

- **Scenario 2.2: Runtime adaptation of the application**, following the application start, the ACCORDION Platform continuously monitors and manages the application through the ACCORDION Orchestrators. The platform deploys or shuts down the game server based on rules passed with application model.

Prerequisites for all scenarios:

- Communication from mobile devices to Accordion Platform and Federation is established via WiFi or 5G.
- The channel for communication with ACCORDION platform should be known to the application. Direct communication between application components is realised using UDP protocol.

- Application must be submitted to the ACCORDION Platform along with application model.

Scenario 2.1: Start application

1. Application developer enters the ACCORDION Platform portal.
2. Application developer clicks button to start the application.
3. Request is sent to the ACCORDION platform.
4. ACCORDION platform manages that request and then deploys the new game server on minicloud.
5. Application start status is returned to the Application developer

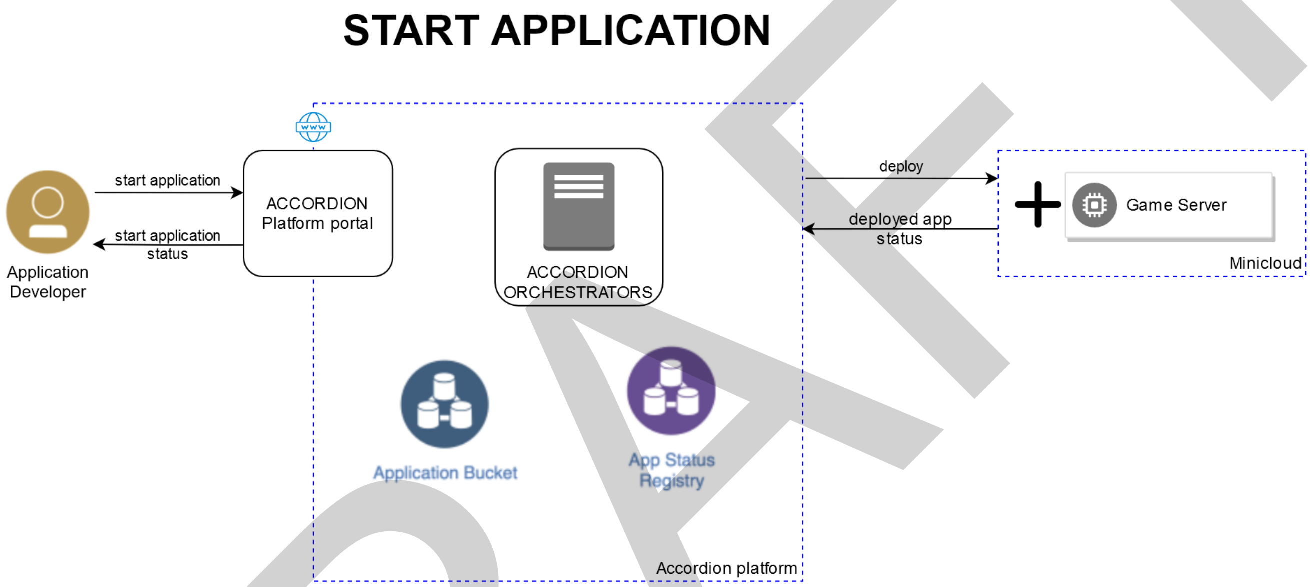

Figure 7: Scenario start application – UC#2

Scenario 2.2: Runtime adaptation

1. ACCORDION Platform receives the game servers' status from the game servers' status DB.
2. ACCORDION Platform gathers QoE data and app status data from its components.
3. Based on rules passed with application model in **Scenario 0: Submission of the application** the platform decides whether (and where) to deploy a new game server, shutdown existing one or do nothing.
4. ACCORDION Platform deploys or shuts down the game server.

The game server status DB contains game servers status. The game server status is a status of a game server, therefore more game specific containing the following information about game server:

- server IP and port
- server name (probably string based on geolocation)
- maximum available players,
- currently connected players amount
- game server start datetime
- game server status update date, time

- information about all connected players (information about each player: player id, player ip, server joined date time)

The game server status data (together with application status data) is used by ACCORDION Orchestrators to make orchestration decisions.

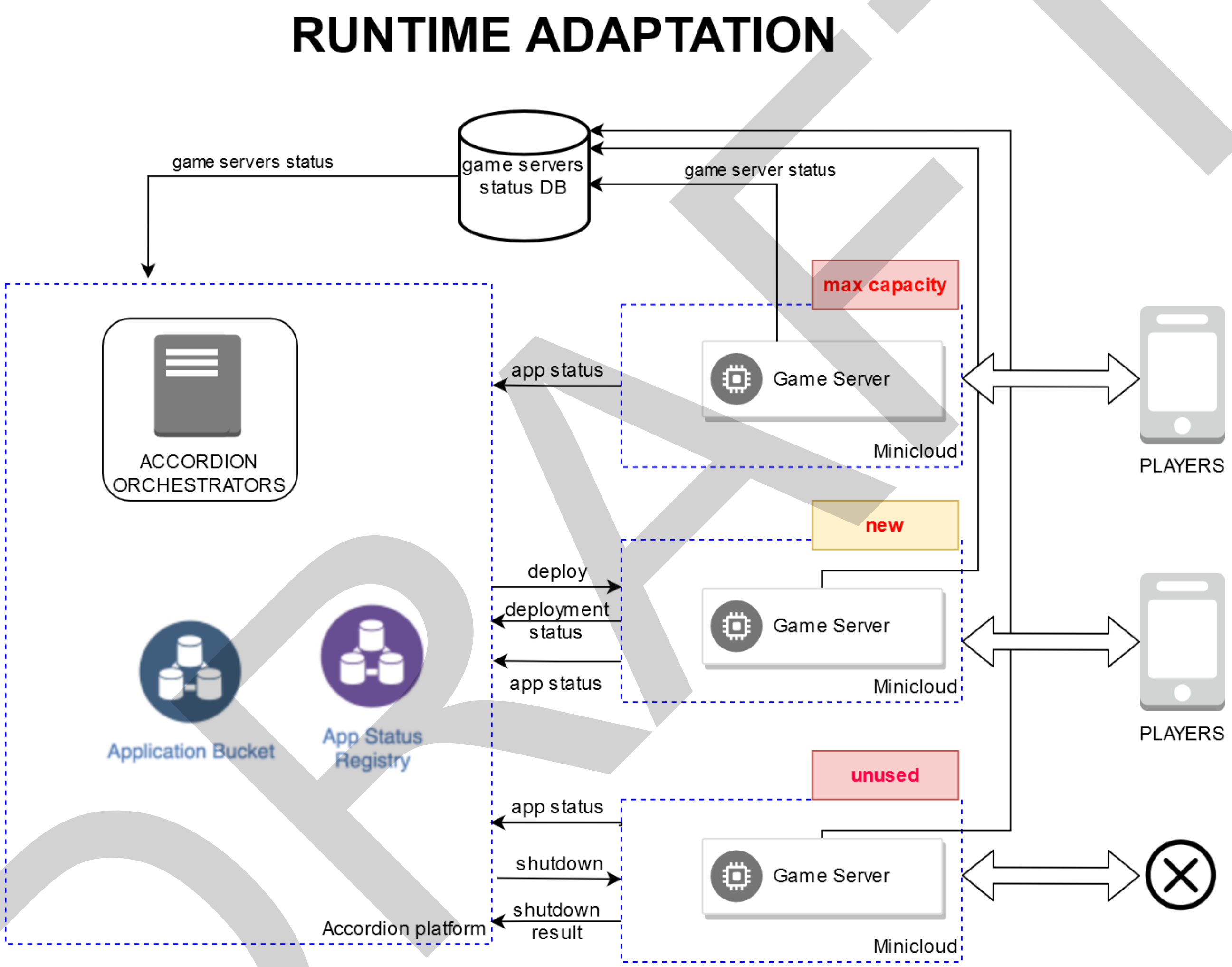

Figure 8: Scenario runtime adaptation – UC#2

3.4.3 Privacy/security aspects

All users will be logged in with email and password and their password will be stored encrypted. Users' data stored outside of ACCORDION will be stored in AWS DynamoDB and performance measurement data will be stored as log files in AWS S3.

3.5 Use Case#3: QoE optimization in content delivery

3.5.1 Application description and challenges

Plexus is a technology company. Two of our key products are Traqus and Anblick[4]. Traqus is a geo-localization platform which processes and analyses the information obtained from mobile devices detected by access points within a Wi-Fi network. Anblick provides digital signage and works as a content manager where content can be programmed to reproduce on screens, for example in a hospital, train station or mall. Currently these cloud solutions are independent, and information is not exchanged between them.

The transitioning to the ACCORDION platform will support the Plexus business case to provide personalized content by integrating both products incorporating edge computing to ensure lower latency and higher performance on average network conditions.

Our approach towards ACCORDION raises the development of a brand-new complete platform, based on a PaaS model integrating edge computing and derived services. A multi-level, microservices-based scenario platform that calculates varied information from different locations and creates real-time analytics. Analytics and derived knowledge elements that provide advantages to all parts of a new value chain based on information processing. The platform will allow for processing of device information on the edge, reducing latency and allow for advanced calculations to be performed in the cloud. Based on the results of these calculations adapted content will be streamed to the different screens within the network.

The continuous processing of information will be managed from different levelled nodes. The integration of all the information will lead to an expert system whose parts are mutually feeding each other. As a result, content delivery to different devices and the measurement of their response will be done in a consistent way across peripheral devices and connections.

This Use Case consists of two sub Use Cases, UC#3.1 and UC#3.2. For UC#3.1 and UC#3.2 two and four scenarios are supported respectively.

UC#3.1 Client identification and adapted content displayed on screens (digital signage): Based on the characteristics obtained from the mobile device information from the clients in a certain space like a mall, content reproduced on the information screens within that space is adapted (Figure 9). For example, in case there are more men than woman, content can be adapted to their interest.

Figure 9: Digital signage output– UC#3

[4] https://www.plexus.es/hacemos/

UC#3.2 Mobile Augmented Reality game: The game is an augmented reality customer loyalty game where points can be won based on clients playing against each other. A 3D object needs to be captured in order to gain points. Depending on the interaction between the user and the 3D object considering the user geolocation the game scenario changes. The following figure (Figure 10) shows the 3d object to be captured by the user. The 3d object will be based on the logo of the client (hospital, mall, train station, etc.). In this case the 3d object is based on the logo of Plexus.

Figure 10: Mobile Augmented Reality game - UC#3

The main challenges are:

- Autonomous content decisions based on the information from the probe requests (i.e frames or message types) processed within the edge and cloud environment.
- Optimization of the user experience (QoE) of the AR game. In mobile games and in applications based on spatial location, which depend largely on speed and latency to offer adequate QoE. current technologies require a minimum DSL connection type (1-100Mbps) to facilitate constant high-speed Internet connection for a seamless experience. Even in this situation, the user experiences QoE issues due to buffering or latency. An edge computing-based strategy will try to reduce processing time, for smart space applications, from the current processing time of 10 minutes (the current process includes email registration, verification, content adaption) to less than 60 seconds and with 30 seconds as an optimal figure when processed on the edge.
- Channel all data and processing orchestration within the ACCORDION system. To enable a decrease of the latency between the different level nodes and the sensors involved. Increasing the communication quality and speed between the first level nodes and the final nodes.
- Develop an autonomous system capable of deciding when to process data locally or download it from the cloud considering the user experience. The system is enabled to decide autonomously to reap the benefits of edge processing or cloud processing. This helps ensure an optimal user experience and avoids delays that dedicated apps or gamers are highly sensitive to.
- To learn from high-volume data scenarios such as online video streaming, mobile gaming / cloud gaming / data collection and processing applying Machine Learning, Big Data, neural network and deep neural network (DNN) techniques.

3.5.2 Scenarios

To enable the application to take advantage of the ACCORDION platform the applications are redesigned to enable the migration of certain functions / application components in the Edge minicloud or in central clouds. The following figure, illustrates the proposed approach. The ACCORDION edge device is responsible for detecting and identifying probe requests from mobile devices. Basic calculations are performed on the edge and the data is shared with the ACCORDION mini-cloud for further processing and more advanced calculations. Content streaming is done based on the decision making in the mini-cloud. For the first sub Use Case content is streamed to the generic screens and for the second sub Use Case content is streamed to the mobile game on the phone of the user.

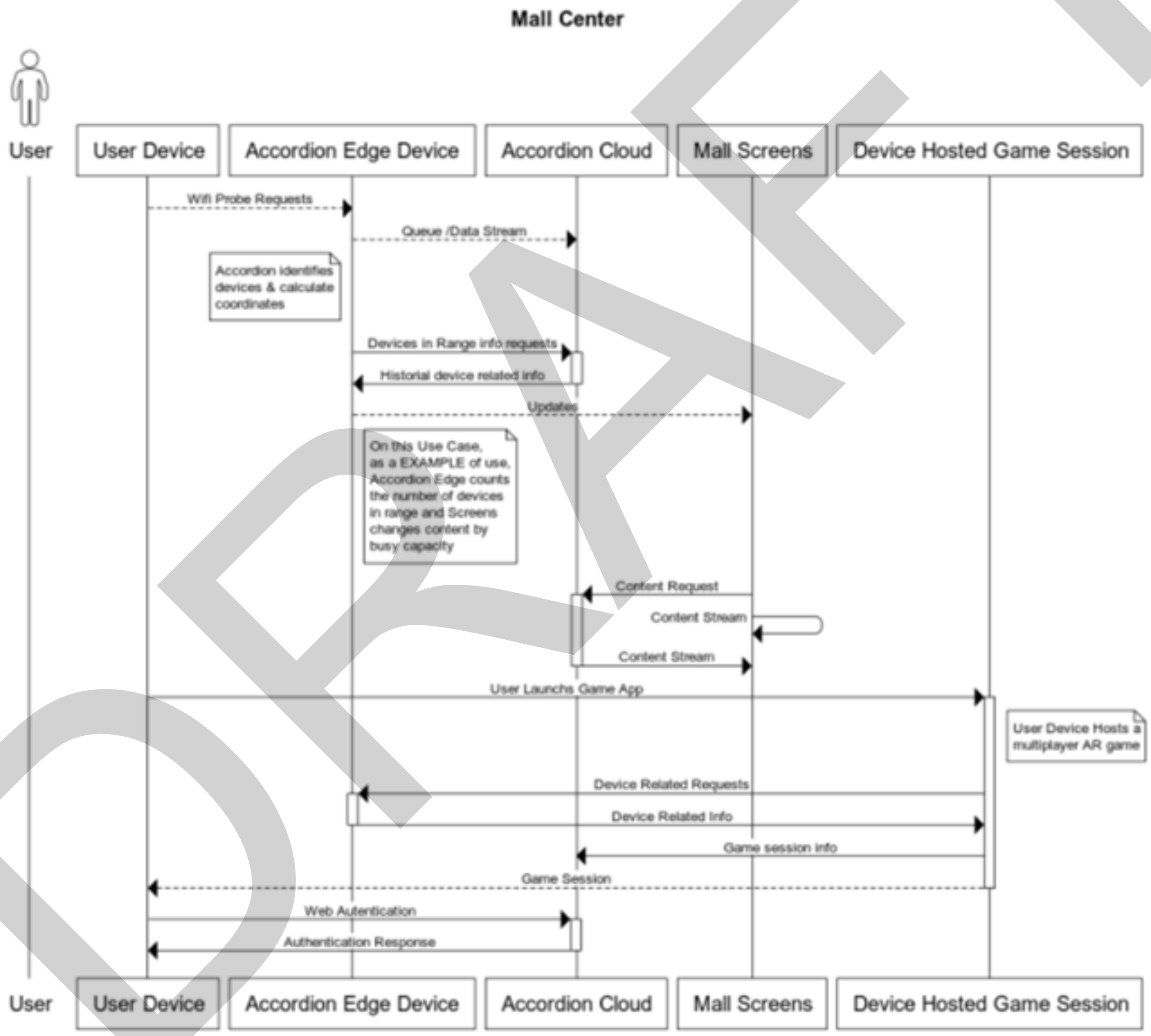

Figure 11: Sequence diagram – UC#3

UC#3.1: Client identification and adapted content displayed on screens (digital signage)

Two main scenarios are supported aside the submission of the application scenario (Scenario 0 – sec. 3.3.2):

- **Scenario 3.1.1: Start Edge Service Application**, automatic activation of the required processes in the edge device, which is a dedicated physical device. This edge device can interact with user mobile devices, with microservices hosted in the mini-cloud and in the cloud, managed by the ACCORDION platform.
- **Scenario 3.1.2: Run time adaptation**, communication between the end user device, edge device (LS) and a remote service (RS), hosted in the cloud, and the process that allows the information to be exchanged. The LS can perform tasks offline when not connected to RS.

Prerequisites for the scenarios of UC#3.1:

- Deployment pipeline (deployment environment) enabled and running.
- Edge device (Raspberry Pi) connected to pipeline.
- Remote service cloud connected to pipeline.
- LS running the dedicated code developed to perform specific tasks.
- User mobile device Wi-Fi antenna enabled.

Scenario 3.1.1 – Start Edge Service Application

1. The developer requests the deployment of the application component and the containerization and deployment on an edge device (LS) is executed by the ACCORDION platform. Remote Service (RS) in the cloud maintains the parameters for this LS via message queue and REST API. On LS boot, it spawns containerized services and monitoring connections.
2. The ACCORDION platform checks the deployment coverage criteria.

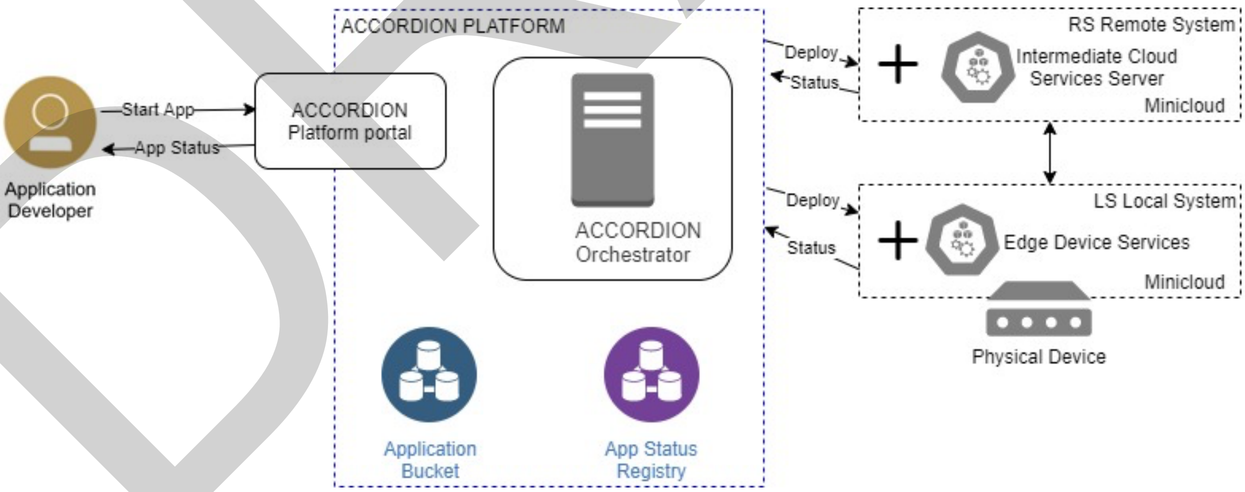

Figure 12: Scenario start edge service application– UC#3.1

Scenario 3.1.2 – Run time adaptation

User enters the Wi-Fi range area with personal device Wi-Fi antenna turned on.

1. The probe request from the mobile device of the user is detected by the LS, LS identifies hardware specs and devices relative position.
2. LS shares device info with RS via REST API, hardware specs and devices relative position.
3. LS communicates with Smart Screen (content management application) via API REST to exchange status changes and content tags, broadcasted content is adapted based on LS/RS instructions
4. In case LS instance services reach their limit, meaning that a maximum number of mobile devices is being processed, multiple LS instances should be spawned by the ACCORDION platform.

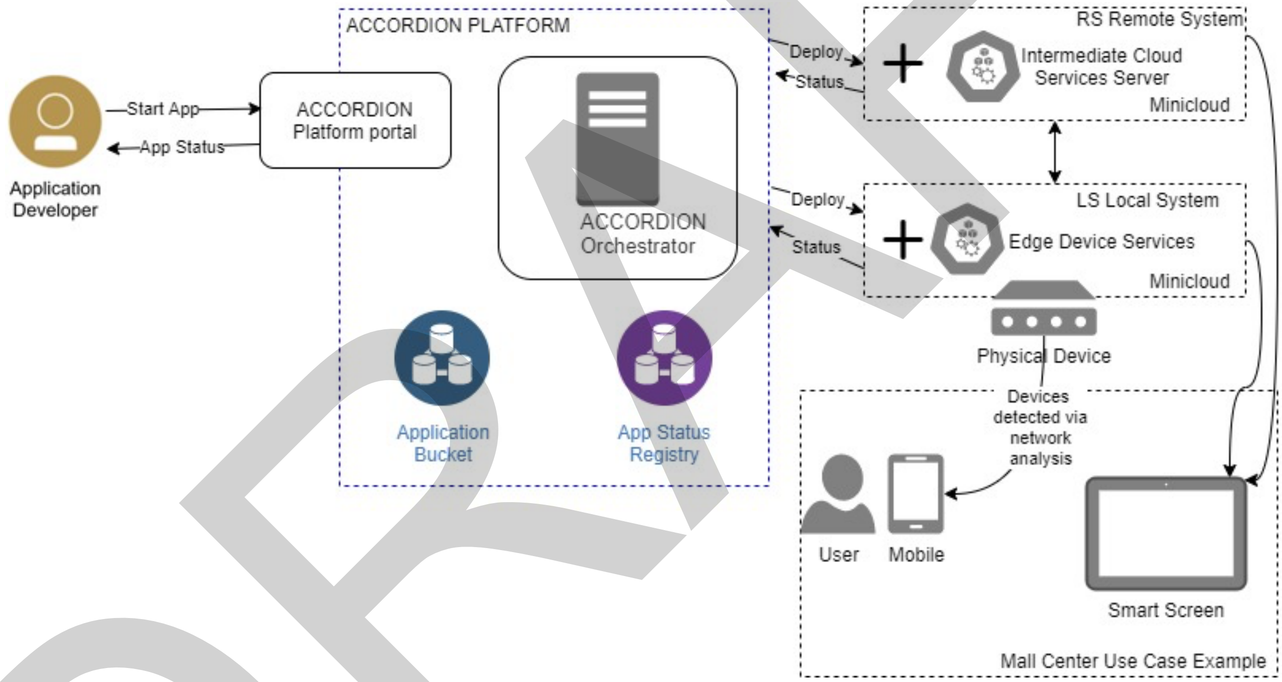

Figure 13: Scenario run time adaptation– UC#3.1

UC#3.2: Mobile augmented reality game

Four main scenarios are supported aside the submission to the application (Scenario 0 – sec. 3.3.2):

- **Scenario 3.2.1: User launches mobile Application**, the scenario involves the user launching the native mobile app from his/her mobile device and all necessary processes are started to make the platform up and running.

- **Scenario 3.2.2: Multiplayer AR Game - Runtime Adaptation of Application,** this scenario describes the communication between a mobile device, a dedicated physical edge device (LS) and a service (RS), hosted in the cloud, and the process that allows the game to be executed. The mobile device will be connected through API to the LS and this will be connected via message queue to the RS.

- **Scenario 3.2.3: User Device Hosts Game Session,** after the application is started, the user can choose whether to participate in an existing game room as a client or to start a new game and become the host of the session. A second mobile device can join the game room via IP.

- **Scenario 3.2.4: Multiplayer Session**, within a multiplayer game room the relative position of the participating devices is shared by REST API with the LS to manage their position.

Use Case#3.1 already shows the complete architecture including the relation between the ACCORDION platform and the LS and RS. Hence, for Use Case#3.2 this part has been omitted in the Figures.

Prerequisites for the scenarios of sub Use Case#3.2:

- AR Game APK installed on user mobile device.
- Dedicated physical edge device on range (Raspberry Pi).
- User mobile device Wi-Fi antenna activated.
- Edge Device Services On.
- User mobile device on Edge Device antenna range.
- Cloud related developed services activated

Scenario 3.2.1 - User launches mobile Application

1. User presses game icon and launches game Application.
2. Game Application is running on device and sends game info to minicloud (LS) that shares with RS.
3. Game Application running on device, mobile device is within range of Edge Device (LS).
4. Application requests device location info to the minicloud
5. Game Application starts AR Game Session with relative location elements

Note: AR Game Session can run with a complete experience without cloud data connection, no intermediate, managed or ACCORDION dependent, but existence of RS resource improves gameplay, records, interactions, etc.

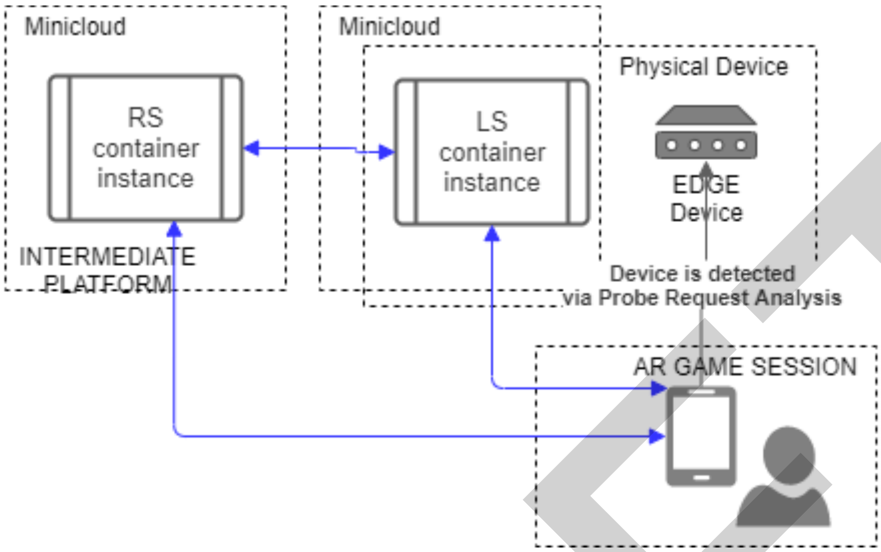

Figure 14: Scenario multiplayer AR game-user start application– UC#3.2

Scenario 3.2.2 - Multiplayer AR Game - Runtime Adaptation of Application

1. User starts AR mobile device game.
2. AR game session shares data with LS.
3. If mobile device is in the LS range, the AR experience can start, AR session game is hosted by user device.
4. Another user in LS range could participate in hosted session or start they own host session.
5. LS system improves the AR multiplayer experience adding information about their relative positions to the game elements.
6. Session game data is shared with LS, that shares it with the RS, to improve the game loop.

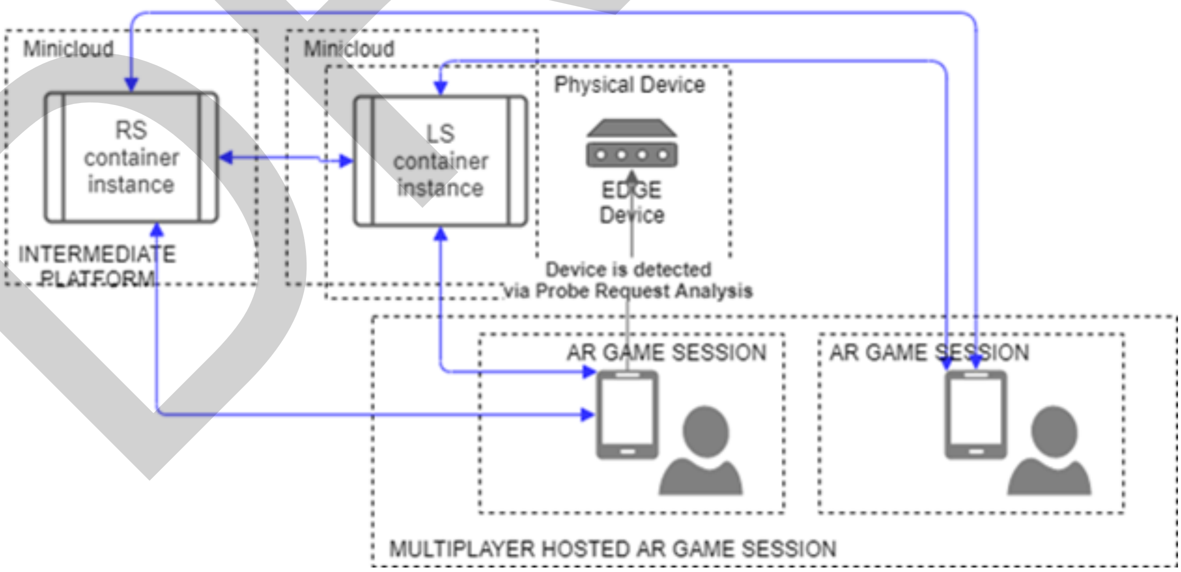

Figure 15: Scenario multiplayer AR game session– UC#3.2

Scenario 3.2.3 - User Device Hosts Game Session

1. AR Game Session allows user to become host of a multiplayer Game Session.
2. User selects to become host of a multiplayer Game Session.
3. One of the mobile devices is assigned as host (host information is not registered anywhere but in the user device)

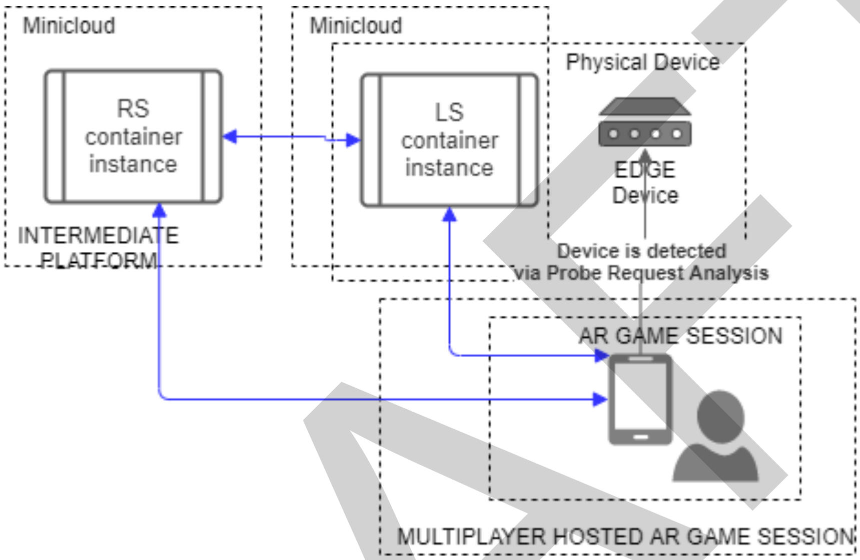

Figure 16: Scenario multiplayer AR game-user device hosts game session- UC#3.2

Scenario 3.2.4 - Multiplayer Session.

1. User presses game icon and launches Application.
2. Application running on device, application shares game info to cloud (RS), if any.
3. Application running on device, Application starts Edge Device (LS) in range discovery.
4. LS discovery, Application requests device location info.
5. Application starts AR Game Session with relative location elements.
6. AR Game Session allows user to become host of a multiplayer Game Session, join on an existent multiplayer Game Session or stand in an alone Game Session.
7. User selects join on an existent multiplayer Game Session
8. Devices relative position info is requested to devices for the LS to manage a multiplayer AR game

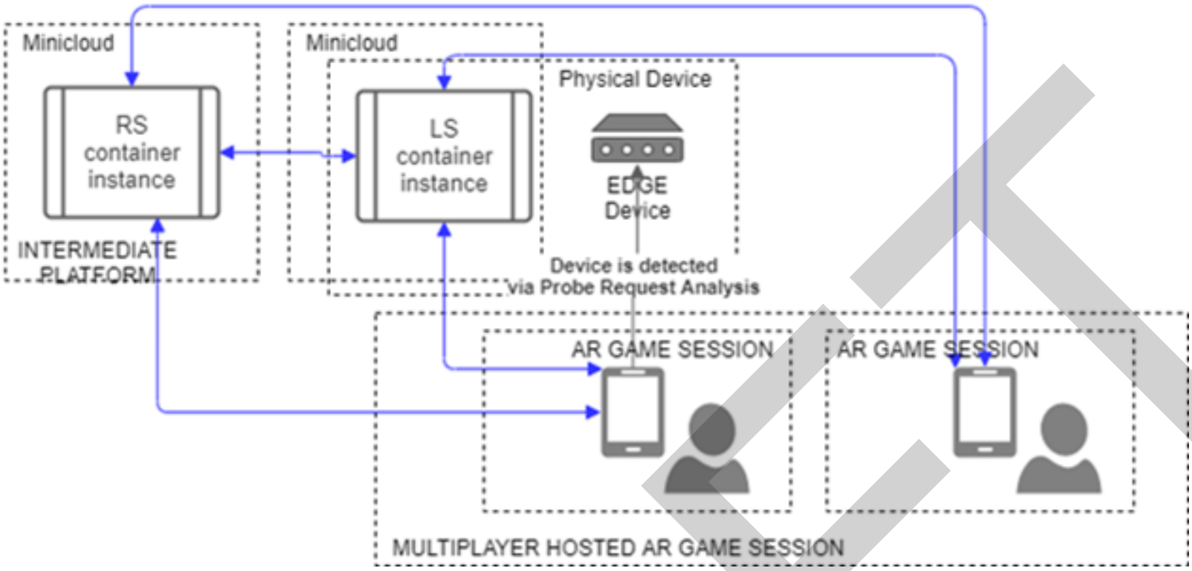

Figure 17: Scenario multiplayer AR game-multiplayer session- UC#3.2

3.5.3 Privacy/security aspects

- Raspberry PI identifies anonymized IPs, these are not stored locally to avoid data leakages.
- For data transfer between Raspberry PI and Cloud encrypted queue messaging is used.

In order to ensure data ACORDION compliancy key concepts regarding GDPR have been described and the user's rights are safeguarded we follow in all stages of development of the Use Case the ´**Data protection by Design**´. Recital 78[5], making available to users quick and easy access to manage their personal data, Article 15, GDPR and to all the complementary information, as the time during which the information is kept, the use that is being made of the data. Such access would not be complete without incorporating data rectification, Article 16, or data deletion. Provide options to modify authorizations for data processing responds to the respect of the article 18 GDPR. Finally, the system will also include options for the export of personal data. Article 20 GDPR. A detailed privacy and ethical analysis has been realized and is available upon request.

[5] https://www.privacy-regulation.eu/en/r78.htm

4 Pilot plans and evaluation

4.1 Pilot prototypes

The project will run test and evaluations to assess the platform from different perspectives with respect to technology and QoE aspects. For each Use Case certain applications/modules provided by the end-users of the project (i.e. OVR, ORBK and PLEX) are selected for the experiments to be conducted in the evaluation phase considering the scenarios in sec. 3.3.2, 3.4.2, 3.5.2. These will focus on verifying the functional aspects of the intended operation following the requirements described in D2.1 for each Use Case and which define what the system must accomplish or must be able to do. In parallel the non-functional requirements which address different attributes of system will be addressed.

The pilot prototype to support the evaluation and trial of Use Case#1 will utilize a specific module from the suite of different products/modules of OVR that simulate in VR different medical training operations. For the purposes of the project the CVRSB (COVID-19: VR Strikes Back) module is selected as the application module "in test" for experimentation with the ACCORDION platform and to support the required modifications in the application (i.e., computation offloading mechanisms, app to platform and app-to-app communications, integration of third-party tools) needed to exploit the services of ACCORDION. The CVRSB supports a **collaborative VR training** mode where multiple users can join in the same environment to perform the covid-19 swab testing. A gamified mechanic is included to demonstrate the proper hand disinfection according to the World Health Organization. Screenshots from the VR simulation module are shown in Figure 18. Each end-user will be using an untethered HMD (e.g. Oculus Quest).

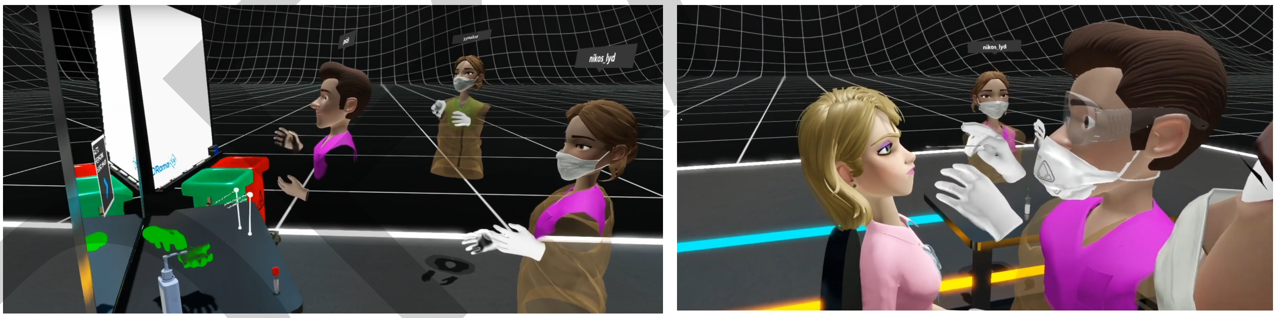

Figure 18: OVR application module CVRSB

The pilot prototype for **the mobile multiplayer game** will use a specific game for mobile devices by ORBK. The game is about moving your character on an open level along with other players. Players can see each other's characters, obstacles and other enemy items. Players will be able to shoot. The goals of gameplay are not yet decided. The architecture of the game is designed in a way allowing to add relatively easily new functionalities and change/adjust gameplay rules. The whole system consists of Game Servers and Game Client. Game Servers will be deployed and run in the ACCORDION minicloud environment, while Game Clients will be running at the end-user's mobile devices. The Game Client will be built for the Android OS. We are aiming to achieve up to 100 players per each instance of the Game Server while maintaining seamless gameplay experience. We are currently focusing on the multiplayer mechanisms, efficient network communication, lowering RTT latency and client-side movement prediction. This is to maximize the QoE of the Game. Relevant screenshots of the mobile game are illustrated in Figure 19. The end users will be using a mobile phone with Android OS and at least 2GB RAM.

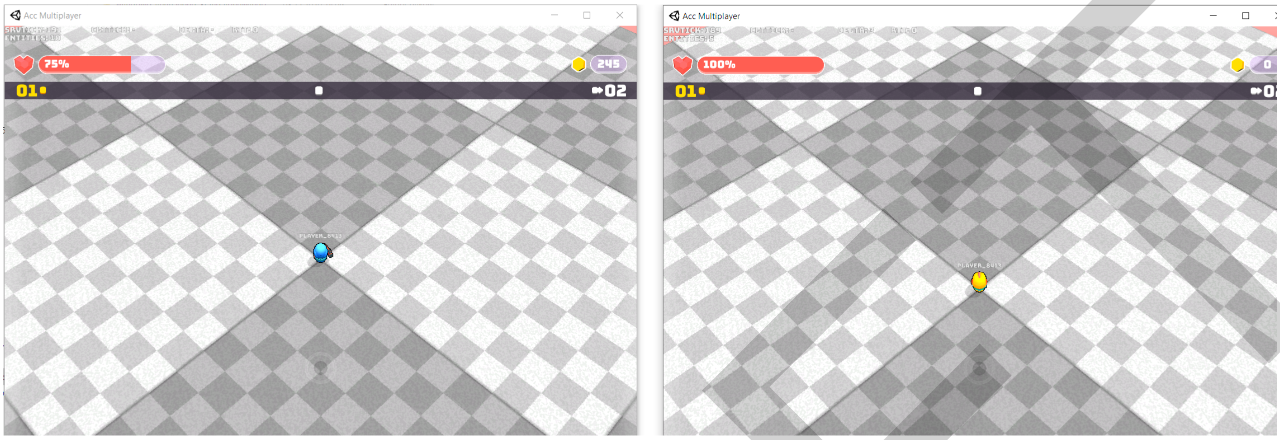

Figure 19: ORBK mobile game

The pilot prototype for **QoE optimization on content delivery** will reside on a new platform that integrates the Traqus and Anblick products of PLEX (sec. 3.5.1). The platform aims to manage both a real-time analytics complete model that evaluates any information received by the system and to provide the users with content-experience related with the evaluation made by the analytics model, with special interest in an AR gaming and a screen-location-wide marketing content experience. For the analytics part, we are aiming to achieve a low latency, efficient network communication, decreasing the current analytic experience processing calculation from minutes to lower that 90 seconds. In the case of the screen-content, we are aiming to achieve a management of up to 50 screens per edge node, with near-real time change of content (less than 2 seconds) and with the lowest possible screen update time between the different screens (milliseconds). The AR gaming is related to the recovery of objects in a real scenario, with different players (up to 100) interacting at the same time with the objects. Players will see the objects and try to interact with them, in direct competition with the other players. Relevant screenshot of the mobile game was depicted in Figure 10 (sec. 3.5.2).

All pilot prototypes will evolve during the first and second cycle to support the aforementioned scenarios. The implementation details of the different releases of the pilot prototypes will be reported in D6.4 [M17] and D6.9 [M32]. The following Table (Table 2) provides the list of scenarios attributed to the three pilot prototypes of the individual Use Cases of the project that will be assessed during the first and second cycle evaluation phase.

Table 2: Pilot prototypes and scenarios of the first and second cycle

Pilot prototype	**Scenario**	**Supported in the first cycle**	**Supported in the second cycle**
Collaborative VR module (CVRSB)	Scenario 0: Submission of the application	✓	✓
	Scenario 1.1: Start application	✓	✓
	Scenario 1.2: User creates a session	✓	✓

	Scenario 1.3. Run-time adaptation of the application (Cycle II)		✓
Mobile multiplayer game by ORBK	Scenario 0: Submission of the application	✓	✓
	Scenario 2.1: Start application	✓	✓
	Scenario 2.2: Runtime adaptation		✓
QoE optimization on content delivery integrating Traqus /Anblick products	Scenario 0: Submission of the application	✓	✓
	Scenario 3.1.1: Start Edge Service Application	✓	✓
	Scenario 3.1.2: Run time adaptation		✓
	Scenario 3.2.1: User launches mobile Application	✓	✓
	Scenario 3.2.2: Multiplayer AR Game - Runtime Adaptation of Application		✓
	Scenario 3.2.3: User Device Hosts Game Session		✓
	Scenario 3.2.4: Multiplayer Session		✓

4.2 Evaluation methodology

In general, software validation is the process of developing a “level of confidence” that the system meets all requirements and functionalities as set out during the preliminary phase of the project. It is a critical process used to assure the quality of its components and as well as the overall platform or system. It allows for improving/refining the end platform. The challenge is to develop appropriate measurable criteria along with an evaluation methodology including targeted groups, procedures, tools to collect data and expected results. The results have to stress both negotiation and consensus concerning the evaluation procedures, and the conclusions received.

Different quality models for software have been proposed in the literature, each one addressing different quality attributes that allow evaluating the developed software. Though applying a selected model to a given system is not a straightforward process and there are no automated means for testing software against each of the characteristics defined by each model. ISO/IEC 25010: 2011 model [1], also known as SQuaRE, provides the leading models for assessing software product and follows a simplified analysis from its predecessor, ISO 9126 [2]. The product quality model categorizes product quality properties into eight characteristics (functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability and portability). Each characteristic is composed of a set of related subcharacteristics. Furthermore, the model defines three types of metrics:

- **Internal metrics** associated with static internal properties of a system such as number of function calls, number of rules.
- **External metrics** associated with dynamic external properties. These are metrics that are observable when the user interacts with the system (i.e. the user performs a task/function/operation and observes the response in the sense of time required, results obtained etc.).

- **Quality-in-use metrics**, which refer to metrics that evaluate the extent to which a system meets the needs of the user.

Aside the definition of the appropriate metrics, the necessary tools need to be utilized to gather the relevant evaluation data. Some or all of the following tools will be utilized to gather the necessary evaluation data:

- **Logs analysis**. Individual partners may have to instantiate specific tools that will give them the opportunity of monitoring the component's behaviour and analyse the logs during the pilot execution.
- **Questionnaires.** Evaluation questionnaires will be created individually or collaboratively by the platform developers and the Use Case providers to derive conclusions on the technical characteristics and the functionalities of the platform. This activity will be led by OVR as WP6 leader. The questionnaires will be used where needed to assess individual functionalities and aspects (subtopics) (see **Error! Reference source not found.**).
- **Usability and user acceptance testing.** It will be conducted by different partners of the consortium with the use of supportive tools for the assessment of the platform's performance, usability and acceptability. OVR will ask from the contributing partners to the related subtopics (see **Error! Reference source not found.**). to define the usability tests to be conducted to evaluate the individual component functionalities of the platform. OVR together with CNR will define the usability and acceptability tests for the platform at a whole.

For the purposes of ACCORDION all three types of metrics will be used, though not all of the characteristics of the model will be applicable. Considering the ongoing work as part of the integration, the final list of criteria and metrics will be reported as part of the respective deliverables summarizing the outcomes from the pilot's evaluation in the first and second cycle. Though in sec. 4.4 and **Error! Reference source not found.** we provide a list with identified subtopics referring to different aspects/functionalities of the platform to be evaluated along with corresponding metrics. These will be used as the basis for further analysis and refinements addressing in parallel the characteristics of the aforesaid ISO/IEC 25010: 2011 model.

4.3 Experimentation requirements and planning

The overall purpose of the of the pilot's evaluation is to test the platform functionalities both on component level and as a whole. Furthermore, it is intended not only as a means to evaluate the overall progress of the technical work so far, but even as an early feedback to the project development itself.

As already mentioned, the ACCORDION consortium will execute two evaluation sessions or pilots' trials. The project team expects to take advantage of the validation results from the first cycle and subsequently adjust the development work accordingly for the ACCORDION system and the pilots' implementation in the second cycle. The latter will be utilized in the 2nd pilot execution and evaluation.

For each pilot iteration, a series of activities will be followed with the ultimate goal to lead to a successful pilots' evaluation. The piloting activities and the actors involved in each one of them are described in the following sections.

4.3.1 Monitoring team

For planning and monitoring purposes of the piloting activities specific roles are assigned to members of the project team. Namely, OVR as Use Case and pilot execution leader will set up, prepare, maintain the pilot system and monitor the activities during pilot's execution phase. In more details the piloting execution leader will a) coordinate the preparation and set-up activities , b) execute pre-validation testing and c) monitor the experiments execution and progress. In parallel the Technical Manager of the project (CNR) will monitor the technical aspects related to the platform performance and support the piloting activities from the technical point of view. The Technical Manager will: a) perform pre- tests in order to ensure the proper operation of the platform before the execution of the pilots' experiments, b) in parallel monitor the execution of the experiments and interfere if the stability of the system is affected.

4.3.2 Set-up requirements

To run the pilots, the following setup steps need to be realized:

- Setting up the testbed infrastructure that will be aligned with the integration plan
- Installing and setting up ACCORDION platform in a bug-free, up and running state, integrating all the components and releasing it with supporting material.
- Setting up the methodological framework for capturing the data that will be used in order to conduct the pilots' experiments and evaluate the platform.
- Running the pilot and reporting the results within a one-month period.

The steps preceding the evaluation phase are detailed in sec. 5.

4.4 Experiments and evaluation

Evaluation revolves around the core technical objectives of the project as well as their sub objectives, thus:

Objective 1: Maximize edge resource pool size	Resource monitoring & characterization
	Resource indexing & discovery
	Hybrid elasticity
Objective 2: Maximize robustness of cloud/edge compute continuum	Intelligent, adaptive resource orchestration
	AI-based network orchestration
	Resilience across the dynamic continuum
	Security across the dynamic continuum
	Privacy-preservation across the dynamic continuum
Objective 3: Minimize overheads in migrating applications to cloud/edge federations	Application data services model
	QoE assessment models
	DevOps to orchestrate network paths at the edge
	DevOps to support application deployment
Objective 4: Realize NextGen applications	Collaborative VR prototype
	Multiplayer mobile gaming
	QoE optimization on content delivery

In relation to the above stated objectives appropriate metrics need to be defined to be able to evaluate the platform and reflect the characteristics that they represent. They also need to allow for appropriate measurements to be obtained, either through quantitative methods (e.g. by software/modules tests/simulations, usability tests) or qualitative methods (e.g. through user observations) using the appropriate tools to collect the necessary data (sec. 4.2).

Error! Reference source not found. performs a full -breath association of: a) the core functionalities and aspects of the platform that would be evaluated as part of individual subtopics; b) high-level objectives of the project; c) the prevailing technical/scientific, functional and non-functional requirements from D2.1 along with their individual IDs; d) the partners involved in the individual subtopics for supporting the definition of metrics, evaluation procedure and assessing experimentation outcomes; and e) a preliminary list on the type of evaluation metrics/procedure for each subtopic. The exact metrics and types of experiments to be used during the pilots' evaluation are subject to further refinements and the final list will be reported in the upcoming deliverables (D6.5, D6.10).

Table 3 Association of evaluation subtopics, objectives, requirements, partner involvement and experiments/metrics

Subtopic description	ACCORDION Objectives	Related requirements (ID)	Partner	Metric/evaluation
Changes and complexity when migrating the application components to ACCORDION	Objective 3 Objective 4	R_TS_01, R_TS_06, R_TS_07	HUA/ AALTO	User acceptance tests
Automated lifecycle management Service monitoring Fault tolerance	Objective 2	R_TS_02, R_TS_04, F_UC1_07, F_UC1_17, NF_UC1_11, NF_UC1_12, F_UC2_17, F_UC3_11, F_UC3_12, F_UC3_25, F_UC3_60, NF_UC3_04	HUA/ AALTO/ HPE	Latency and RTT Traceability of components through-out their lifecycle. Detailed measurements by the platform
Resource discovery provision and service elasticity	Objective 1	R_TS_09, R_TS_11, R_TS_12, R_TS_16, F_UC1_03, F_UC1_07, F_UC1_08, F_UC1_09, F_UC1_12, F_UC1_13, F_UC1_14, F_UC1_19, NF_UC1_10, F_UC1_14, F_UC2_26, NF_UC2_09, NF_UC3_07, NF_UC3_08	HUA/ AALTO/ CNR	Measurements on types of resources discovered and migration times Detailed measurements by the platform
Deployment mechanism	Objective 3	R_TS_13, R_TS_14 R_TS_15, NF_UC1_15, F_UC2_08 , F_UC2_09 , F_UC2_11 , F_UC2_13 , F_UC2_22 , NF_UC2_06, NF_UC3_03	BSOFT	User acceptance and detailed measurements by the platform
Definition of accepted QoE levels	Objective 3	R_TS_03	HUA/ TUB	QoE model based on standards Subjective tests
Security and privacy	Objective 2	R_TS_05, NF_UC1_09, NF_UC1_13, NF_UC1_18, F_UC2_04, NF_UC2_02, NF_UC2_03 , F_UC3_04, F_UC3_09, F_UC3_17, NF_UC3_05	HUA/ HPE	Security Policy (ISO 27001) Compliance/Adherence and detailed measurements by the platform
Latency, data rate, number of users	Objective 4	F_UC1_01, F_UC1_02, F_UC1_10, F_UC1_11, F_UC1_15, NF_UC1_01, NF_UC1_02, NF_UC1_04, NF_UC1_05, NF_UC1_06, NF_UC1_07, NF_UC1_16, NF_UC1_17, NF_UC1_19, NF_UC1_20, NF_UC2_10 , NF_UC2_11 , F_UC3_59	OVR, ORBK, PLEX	Measured in an end-to-end manner following different metrics

ACCORDION receives funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 871793

4.4.1 Technology

As mentioned in the previous paragraphs different experiments will run to assess the components/functionalities of the platform. The majority of experiments (Table 3) have a technical value for the project as they allow to profile the technical performance and support further developments and refinements. For these experiments different QoS criteria will be used as part of the metrics in Table 3, considering QoS as the standard for measuring network performance and which embodies the notion that the hardware characteristics (e.g., the storage capacity and the number of processors in the servers) and software characteristics (e.g., the interface development) can be measured, improved, and guaranteed. An indicative list is provided in Table 4. The limitation of QoS is that it doesn't account for the relationship between the end user and the technology. QoE looks at the impact of the network behavior on the end user, a fuzzier domain where certain network imperfections go unnoticed but others may render an application essentially useless. The combination of QoS and QoE takes a much more holistic look at network performance and the end user, considered focal when multimedia services are involved (sec. 4.4.2).

Table 4: QoS Evaluation criteria

Measurable Criteria	**Description**
Response delay (RD)	i) Processing delay (PD): the time required for the server to receive and process a user's command, and to encode and transmit the corresponding data to that client; ii) Playout delay (OD): the time required for the client to receive, decode, and render a frame on the display; iii) Network delay (ND): the time required for a round of data exchange between the server and client.
Network loads	Throughput, packet loss rate, frame rate, traffic shape control
Energy consumption	watts/hour

4.4.2 Subjective Quality of experience

In parallel to the evaluation of the technical issues as described in the previous paragraph, subjective experiments will be carried out as they form the basis of the quality assessment of any multimedia service. A standardized subjective methodology for assessing the experience will be adopted based on ITU-T Rec. P.809 [3] for gaming applications as well as ITU-T P.919 [4] for virtual reality video streaming services. The subjective test is designed in Task 6.3, which is the result reported in D6.2. D6.2. gives an overview on

ACCORDION receives funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 871793

requirements for the development of the Quality of Experience (QoE) model for ACCORDION uses cases. The document targets the development of a parametric-based monitoring model that can predict the mean opinion scores (MOS) on a 5-point ACR scale based on the impact of impairments introduced by typical networks and compression on the quality experienced by users using the OVR, PLEX, and ORBK applications.

D6.2 describe the necessary steps before the development of QoE models including defining the scope of the models, range of the parameters, planning subjective test, and developing the structure of the model.

The subjective experiment has multiple steps, which typically start with an instruction session and training session to explain the subjective test structure to participants as well as training the participant how to interact with the Use Case. Prior to the test, demographic information is collected through a pre-test questionnaire, which depends on the Use Case; some additional information about the experience of the participant with the Use Case will be collected. The core part of the subjective test is then started where participants will interact with the Use Case under different levels of network and compression settings. After each stimulus, the participants will rate the quality based on the post-condition questionnaire. Finally, the test is ended with a post-test questionnaire to get an insight about the possible issues during the test as well as about which quality aspects were considered for the overall QoE judgment. While the general structure of the test remains the same, the details of the experiment, such as the selected questionnaire, duration of the stimulus, might be different depends on the Use Case, which is partially described in D6.2.

4.4.3 Privacy and Ethical issues

We adhere to all known means to avoid any ethical or legal concerns by participants of the experiments. The participants will be informed about the purpose of the test before the experiment begins and are informed, that they may choose to quit participating at any time during the session without specifying reasons. They will also be offered to be informed about the use of the empirical data and about conclusions, which are drawn from them. This inclusion has proven successful and valuable in earlier studies conducted at TU Berlin. Nevertheless, information, which may be used to identify an individual test participant, will be removed at the earliest possible point of time, and caution will be taken to avoid recording such identifying information at all, if not necessary for the purpose of the test.

The data collection and analysis efforts, as well as the experiments with human participants in the lab will be checked by the Ethics Committee of Faculty IV of TUB against the TUB, national and international requirements (process started). This process has been done with other projects of similar type in the past, and it has in all cases led to the approval of the experiments by the Ethics Committee of TU Berlin.

5 ACCORDION integration plan and infrastructure for Use Case validation

5.1 Design goals

The idea of the ACCORDION platform is to provide an integrated collection of services that support the execution of ACCORDION application on top of a federated pool of computational resources. These services are designed and developed independently by the tasks in the WP4 and WP5. The core design goal of the integration at this current stage of the project (M12) is to bring together these services to provide a first comprehensive version of the platform. This would serve to drive the further design of the platform's services on one hand and tuning of the first prototype of the applications on the other hand.

Further goals:

- Identification of the components (intended as deployable units)
- Selection of proper containerization mechanisms for both the ACCORDION services and the applications (including VMs and Unikernels)
- Decision about the orchestration platform for the platform infrastructure (K8s)
- Definition of images and code repositories
- Identification of connection mechanisms

5.2 Infrastructure for Use Case Validation

The Use Case validation and test integration will run on top of a hardware testbed infrastructure that is composed by computational resources provided by the partners and, eventually, public resources. The first version of the infrastructure at the time of this deliverable is composed by an x86 cluster composed by a large main node (2x Intel Xeon E5630 2.53Ghz 4 cores, 112GB RAM) and 4 smaller secondary nodes (E5630 2.53Ghz 4 cores, 16GB RAM). The whole cluster can be used as a private cloud node. Alternatively, the main node can be used as a cloud node and the secondary nodes can be a minicloud.

Further computational resources to expand the infrastructure will be added by the partners at a later stage. The target is to reach an infrastructure with a high degree of heterogeneity, like the one presented in Table 5Table 5 below.

Table 5: Resources and testbed configurations

Resources	Testbed configurations
public cloud	GCP, AWS, Azure
private clouds	VMWare, OpenStack
on-premise clusters	HPC cluster, SBC cluster
Network testbed	MEC, Mobile
end-devices	HMD, Mobile Devices

5.3 Integration plan

Currently, the diverse services of ACCORDION are implemented and tested in private infrastructures of the relative partners. The plan is to exploit the hardware resources defined in the previous section and install them with the ACCORDION platform software. The next steps toward the integration are the following:

- **Finalize the hardware platform**. The hardware resources defined in the Table 5 above to be ready and installed with the ACCORDION platform software. This includes all the services with basic functionalities that are equipped inside the miniclouds and cloud clusters. These components are developed in the context of WP3 and WP4, which will be delivered by the end of M14.
- **Develop the service components.** This step will support the development of those platform components that are specific to the service / are not open sourced, onboard them to the project repository (i.e. GitLab) and perform pre-requisite actions for initial component functionality testing.
- **Integration test definition.** The test shall cover the features requested by the Use Cases. Such testing case are created by considering the scenarios proposed in this deliverable. Also, this work will go in parallel with the definition of interfaces between the various components of the ACCORDION platform and the application to platform communication, according to the work of WP3 and WP4.
- **Test execution and evaluation**. The integration tests will be a core element in evaluating the coverage of Use Case requirements. We will first instantiate the service in a local testing environment and confirm the operational conditions before instantiating the service on the environment (described in sec. 5.2) in an automated manner.

The above process will allow the successful deployment of the ACCORDION platform and the exploitation of the provided services according to the scenarios described in sec. 3. We expect to gather data and information that will be used to organize the proper correction both for the finalization of the development, integration and deployment of phase one (M18) and towards the start of phase two.

5.4 Execution Methodology

The execution methodologies are tailored by taking into consideration the ACCORDION SDK which is designed and implemented in the context of WP5. The SDK gives the possibility to run battery tests when the latest version of an application is submitted to ACCORDION. The testing code will be securely stored in a Git repository behind Https. The plan is to use this feature of the SD to run integration tests (after and in addition to application specific tests).

The evaluation and review of the integration tests is planned to be double. The first analysis is performed by automatic logging tools (e.g., greylog[6]). The second analysis is a manual review that will check more potential problems and evaluate corrective actions.

[6] https://www.graylog.org/

6 Conclusions

This document is the first outcome of WP6 and sets the basis for upcoming output of WP6 Tasks and strongly links to the work that is - in parallel - conducted in other technical WPs, including WP2. It depicts a first version of the scenarios and workflows which will be subject to further refinements during the course of the project. Considering the scenarios, the pilot prototypes, the ACCORDION technologies and the infrastructure for use case validation tightly linked, the description of the pilot trials and evaluation is subject to frequent updates. With this in mind the document provides an overall view of the evaluation methodology along with the dominant modules and functionalities of the platform to be put in test considering the high-level goals of the project. In parallel to the aspects related to evaluation, the integration roadmap is drawn.

References

[1] ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

[2] ISO/IEC 9126. Software engineering -- Product quality.

[3] Recommendation ITU-T P.809 (2018), Subjective evaluation methods for gaming quality, Geneva, Switzerland: International Telecommunication Union.

[4] Recommendation ITU-T P.919 (2020), Subjective test methodologies for 360º video on head-mounted displays, Geneva, Switzerland: International Telecommunication Union.