D2.1: Orchestration, AI techniques, End-to-end slicing and Signalling for the core enablers – design

Revision: v.1.0

Work package	WP 2
Task	Tasks 2.1, 2.2, 2.3 and 2.4
Due date	29/02/2024
Submission date	28/02/2024
Deliverable lead	TID
Version	1.0
Authors	Antti Pauanne (UOULU), Jean Cerrillo Fernandez (UOULU), Sabbir Ahmed (UOULU), Hamid Malik (UOULU), Kenichi Komatsu (UOULU), Adriana Fernández Fernández (i2CAT), Javier Fernández Hidalgo (i2CAT), Diego San Cristobal (ERI), Rocio Dominguez (ERI), Fernando Pargas Nieto (TID), David Moro Fernández (TSA), Aurora Ramos (CGE), Jose Soriano (CGE)
Reviewers	Björn Debaillie (IMEC), Olli Liinamaa (NOK)
Abstract	6G-XR project aims to build advanced experimentation platforms for validating Extended Reality (XR) and Energy Optimization services. Deliverable D2.1 provides an analysis of the required Edge and Core network enablers to fulfil the project's use cases, and the initial solution design of the identified enablers. Concepts mentioned in this document are Edge Computing, Edge Federation, Network Exposure, Network Slicing and IMS Data Channel.
Keywords	Extended Reality (XR), Holographic Communications, Digital Twin, Energy Optimization, Edge Computing, Edge Federation, Network Exposure, Network Slicing, IMS Data Channel

Co-funded by the European Union

6GSNS

WWW.6G-XR.EU

Grant Agreement No.: 101096838
Call: HORIZON-JU-SNS-2022

Topic: HORIZON-JU-SNS-2022-STREAM-C-01-01
Type of action: HORIZON-JU-RIA

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	19/05/2023	1st version of the template	Diego San-Cristobal (Ericsson)
V0.2	20/12/2023	First full draft	All
V0.3	26/01/2024	Version for review	All
V0.4	16/02/2024	Reviewers comments addressed	All
V1.0	27/02/2024	Final version	All

DISCLAIMER

Co-funded by the European Union

6G-XR (*6G eXperimental Research infrastructure to enable next-generation XR services*) project has received funding from the **Smart Networks and Services Joint Undertaking (SNS JU)** under the European Union's **Horizon Europe research and innovation programme** under Grant Agreement No 101096838.

COPYRIGHT NOTICE

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✓
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

EXECUTIVE SUMMARY

Delivering the services proposed in 6G-XR project's use cases (UC1: Resolution Adaptation or Quality on Demand, UC2: Routing to the Best Edge, UC3: Control Plane Optimizations, UC4: Collaborative 3D Digital Twin-like Environment, UC5: Energy Measurement Framework for Energy Sustainability) entail a set of challenges to the Beyond-5G networks that must be addressed for a proper service management and user experience. This document analyses those challenges and identifies some network and core enablers that could help overcome them, paving the way for 6G.

Note that within 6G-XR project, there are two different experimentation sites made available for testing and validation of the use cases. Namely, (i) the South Node, providing network and Edge computing resources distributed along Madrid (in a collaborative lab called 5Tonic) and Barcelona (in i2CAT premises), and (ii) the North Node, providing network and Edge computing resources located in Oulu (facilities also known as 5GTN). This document also details which features and requirements from the use cases are targeted at each experimentation site.

The first of the broad topics that is addressed in this document is the need for an Edge computing platform. Having an Edge computing platform allows the offloading of processing from the XR services user devices to the Edge node, resulting in compact and energy-efficient devices (for example, achieving more appealing and comfortable to wear XR goggles, which would favour the mass adoption of the technology). Extra functions can run as well on the Edge. In this project, Edge platforms located at Madrid, Barcelona and Oulu will be used.

At the South Node, the Madrid Edge platform operated by Capgemini holds computing resources for holding containerized applications and provides their Edge orchestrator to control the applications' life cycle management (LCM). They also consider AI/ML mechanisms for load optimization. They will make available a set of Northbound Interface (NBI) APIs to expose their computing resources and capabilities for third parties to interact with the network in terms of Quality on Demand, Edge Cloud Discovery or Traffic Influence (this one is about being able to use the best network path according to the location of the UE). Madrid Edge will federate with the Barcelona Edge platform operated by i2CAT, which has its own computing resources and Edge orchestrator. With this setup, it will be possible to experiment the selection of the most suitable Edge for holding the applications, as Madrid will learn the capabilities from Barcelona and will be able to use them as their own.

At the North Node, the 5GTN Edge platform operated by University of Oulu will (i) offer an adapter to translate and resend the requests towards the network, (ii) deploy enough resources to hold the 3D Digital Twin services, (iii) include a Time Sensitive Network (TSN) switch and (iv) integrate the Energy Management framework.

Another broad topic that comes up is the E2E slicing. The use of slices is crucial to provide different QoS to different services, tailored to each service specific needs.

The approach for slicing at the South Node will be to have a pre-existing fully operational 5G network provided by Ericsson, with single instances of 5G functions such as AMF or SMF where pre-provisioned slices will be sharing those functions. The slices will have different configurations to accommodate different QoS requirements. The applications will be able to interact with the network through the Network Exposure Function (NEF) by using Service Parameter, Monitoring Event, Analytics Exposure and Application Function Session with QoS APIs, to select the most appropriate UPF and the required QoS for the service.

The approach at the North Node for slicing will be to create all the functions of the 5G network when creating each of the slices, and those functions will be used exclusively by that slice. The 5G Core used is a solution provided by Cumucore. The slice management will be done via the Cumucore API set.

A separate track of network evolution is the enhancement of control plane communications to support XR. In this sense, the IMS Data Channel solution will be implemented at the South Node. This is a solution being standardized in 3GPP Release 18 to enhance the IMS architecture to expose control and media for IMS voice video and data channels to applications. The network enabler in this case is the Data Channel Server (DCS), which includes the Data Channel Signalling Function (DCSF) for the control logic, and the Media Resource Function (MRF) for forwarding media traffic.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3D	Three Dimensional
3GPP	The 3rd Generation Partnership Project
3PP	Third Parties
5G	The 5th Generation mobile network
5G PPP	The 5G Infrastructure Public Private Partnership
5GC	5G Core
5GTN	5G Test Network
6G	The 6th Generation mobile network
AF	Application Function
AMF	Access and mobility Management Function
AI/ML	Artificial Intelligence/Machine Learning
API	Application Programming Interface
AS	Application Server
BTS	Base Transceiver Station
CN	Core Network
CPF	Control Plane Function
CPU	Central Processing Unit
CSP	Communication Service Provider
DCS	Data Channel Server
DCSF	Data Channel Signalling Function
DL	Deep Learning
DNN	Data Network Name
E2E	End-to-End
EDA	Ericsson Dynamic Activation
eMBB	enhanced Mobile Broadband
ETSI	European Telecommunications Standards Institute
EWBI	East-Westbound Interface
FQDN	Fully Qualified Domain Names
gNB	Next Generation Node B
GPS	Global Positioning System
GPSI	Generic Public Subscription Identifier
GPU	Graphics Processing Unit
GSMA	Groupe Speciale Mobile Association
GUI	Graphical User Interface
IEAP	Intelligent Edge Application Platform
IMS	IP Multimedia System
IMSDC	IMS Data Channel
IP	Internet Protocol
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LCM	Life Cycle Management
MANO	Management and Orchestration
MEC	Multiaccess Edge Computing
MIG	Multi Instance GPU
ML	Machine Learning
MRF	Media Resource Function
NaaS	Network as a Service

NBI	Northbound Interface
NEF	Network Exposure Function
NFV	Network Function Virtualization
NRF	Network Repository Function
NSA	Non-Stand Alone
NSI	Network Slice Instance
NSSF	Network Slice Selection Function
NST	Network Slice Template
NVENC	NVidia ENCoder
OAI	Open Air Interface
OSM	Open Source MANO
OP	Operator Platform
OPG	Operator Platform Group
PCF	Policy Control Function
PLMN	Public Land Mobile Network
PoP	Point of Presence
QoD	Quality on Demand
QoS	Quality of Service
RAN	Radio Access Network
REST	REpresentational State Transfer
RIC	RAN Intelligent Controller
S-NSSAI	Single Network Slice Selection Assistance Information
SA	Stand Alone
SBA	Service Based Architecture
SD	Slice Differentiator
SDR	Software Defined Radio
SDN	Software Defined Networking
SLA	Service Level Agreement
SMF	Session Management Function
SST	Slice Service Type
SW	Software
TAC	Tracking Area Code
TCP	Transmission Control Protocol
TN	Transport Network
TSN	Time Sensitive Network
UC	Use Case
UDM	Unified Data Management
UDR	Unified Data Registry
UE	User Equipment
UPF	User Plane Function
URLLC	Ultra Reliable Low Latency Communication
URSP	UE Route Selection Policy
USRP	Universal Software Radio Peripheral
VIM	Virtualized Infrastructure Manager
VPN	Virtual Private Network
VR	Virtual Reality
WP2	Work Package 2
XR	eXtended Reality

1 INTRODUCTION

1.1 OBJECTIVE OF THE DELIVERABLE

The objective of the deliverable is to provide an analysis of the requirements that the 6G-XR's use cases described in D1.1 [1] (UC1: Resolution Adaptation or Quality on Demand, UC2: Routing to the Best Edge, UC3: Control Plane Optimizations, UC4: Collaborative 3D Digital Twin-like Environment, UC5: Energy Measurement Framework for Energy Sustainability) impose to the core networks and Edge platforms of the experimentation facilities used in this project, detail the gaps that the current facilities have, identify the required network and core enablers to develop or integrate and, finally, provide an initial solution design of those enablers.

To better understand this document, note that it covers the progress made in tasks T2.1 to T2.4, which are named T2.1: Computing Tools and Capability Placement, T2.2: Smart and AI-assisted Orchestration, T2.3: Smart and AI-assisted End-to-End slice models, and T2.4: 6G Control Communications Control Plane. These tasks are devoted to identifying and implementing or developing enablers on the mentioned topics. All four tasks need to follow a process of requirement analysis, solution design and development. Thus, the common high level timeplan of T2.1 to T2.4 would be as shown in Figure 1:

2023												2024											
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24
					Analysis/Requirements					Solution design						Development							
													D2.1								D2.2		

Figure 1: High level timeplan of T2.1 to T2.4

At the moment of delivering D2.1, we are half-way to the task finalization date. We have completed the requirement analysis and made significant progress in the initial solution design, even though not all final solution decisions have been made yet. Besides, the solution could require modifications in the development phase. For those reasons, D2.1 covers the work done until the moment of producing the document. The final solution will be detailed in D2.2.

1.2 STRUCTURE OF THE DELIVERABLE

As it was described in D1.1 [1], 6G-XR deploys two different experimentation Nodes for testing and validation of the use cases. Namely, (i) the South Node, providing network and Edge computing resources distributed along Madrid (in a collaborative lab called 5Tonic) and Barcelona (in i2CAT premises), and (ii) the North Node, providing network and Edge computing resources located in Oulu (facilities also known as 5GTN). Each of the Nodes have analysed various topics that they need to address to run the use cases. D2.1 has been splitted in separate sections to gather the work around each of those topics.

Accordingly, the core sections of the document are the following. First, to cover the topic of the capabilities of each of the Edge platforms, there are sections (i) Edge Madrid capabilities, (ii) Edge Barcelona capabilities and (iii) Edge 5GTN capabilities. Then, the APIs offered by the South Node Edge in terms of application LCM are described in sections (iv) Edge NBI APIs and (v) Federation. Following up, the APIs provided by the Core network to handle slices are covered for the South Node in section (vi) CAMARA and NEF APIs for slicing, and for the North Node in (vii) 5GTN slice creation. Also, the capabilities that are going to be provided to the Energy Management Framework are covered in section

(viii) Open-source 5G solutions. Finally, section (ix) Control plane innovations, covers the network enablers needed for the control plane use case that will be experimented in the South Node.

For each of the sections, there is a first subsection called Gap Analysis and a second subsection called Initial Solution Design. The logic of the narrative in Gap Analysis subsection is to start from the requirements of the project use cases, analyse what is the state of the art of the topic and gap to be able to fulfil them in the available experimentation facilities, to get to identify which are the enablers that need to be implemented or developed by this work package. Once the enablers have been identified, Initial Solution Design subsection provides the results of the initial solution design for them.

1.3 TARGET AUDIENCE OF THE DELIVERABLE

This is a public deliverable targeting partners in the project consortium, academic and research organizations, EU commission services, and, in general, individuals with advanced knowledge and interest in the evolution of wireless networks.

2 MADRID EDGE CAPABILITIES

One of the sites of the 6G-XR project South Node is located in Madrid: 5Tonic testbed. This section provides an overview of the requirements and gap analysis done as well as first design on the edge infrastructure and orchestration capabilities needed at 5Tonic to support 6G-XR project's UC1 *Resolution Adaptation or Quality on Demand* and UC2 *Routing to the Best Edge*.

At 5Tonic testbed site, following the paradigm defined by GSMA Operator Platform Group (OPG) [2], Capgemini Engineering is providing an Edge infrastructure and a MEC orchestrator, the IEAP (Intelligent Edge Application Platform). The IEAP will be evolved within 6G-XR project based on 6G-XR requirements to provide XR enablers with means to control the resources and accelerators to run the use cases.

The IEAP also includes a set of in-house developed API to control the location of the application based in SDK developed by Capgemini and to control the application life cycle management (LCM) following GSMA Operator Group recommendations and UE to Edge interaction following GSMA OPG as well.

2.1 GAP ANALYSIS

The use case and functional requirements related to edge orchestration gathered from 6G-XR's UC1 *Resolution Adaptation or Quality* and UC2 *Routing to the Best Edge* outlined in [1] are as follows:

Table 1: 6G-XR use cases requirements relevant for Edge Capabilities [1]

Requirement No.	Requirement Description
UC2.R1	Edge onboarding capabilities, which involves being able to deploy a virtualized media function (container-based) on the Edge – renders on behalf of the user. Provide media Virtual Network Function (VNF) access to GPU.
UC2.R2	Edge discovery, which involves that the network provides a subset of Edge locations where service (is/can be made) available.
UC2.R3	Traffic Influence allows UPF (breakout) reconfiguration. UPF selection is decided by the network, but triggered by the service.
UC2.R4	Load mobility: If the service metrics degrade, then an Edge Discovery process is triggered, and a new Edge can be selected. App is re-instantiated with status migration.
UC1-3.FR17	Media encoding/decoding exploits NVIDIA GPU capabilities.
UC1-3.FR23	The XR Holo Orchestrator shall interoperate with an Edge Orchestrator to request the dynamic instantiation/termination of virtualized networked media functions on the edge and cloud based on demand.
UC1-3.FR25	The XR Holo Orchestrator shall be able to request the network to provide Quality on Demand (QoD) for a specific holographic session or to specific XR clients.

Based on the further elaboration of the requirements, it can be stated that UC1 and UC2 are demanding from the South Node to provide means to:

- Allocate compute resources (CPU, Mem, Storage) and acceleration into the best Edge.
- Share those resources and ensure reservation and optimal allocation.
- Easily adapt the application to the Edge infrastructure.
- Coordinate network with the Edge to provide its service.
- Coordinate several instances of Edges in Barcelona and Madrid in a federated environment to provide an inter-PLMN-like environment.
- Provide Northbound Interfaces (NBI) to easily manage those edge capabilities (e.g. list of available edge candidates, control of Quality of Service) from the application requirements perspective (Note: further details on Northbound functionalities are provided in section 5.1).

2.2 INITIAL SOLUTION DESIGN

Capgemini Engineering Edge framework IEAP will be expanded to cover the identified gaps providing capabilities grouped in the following areas:

i. Capacity Sharing and App Location and Life Cycle Management
ii. Exposure of capabilities
iii. Network Interaction
iv. Federation

Firstly, for the capacity sharing capabilities (i), IEAP orchestrator is implementing a complete Edge Orchestrator following paradigm defined by GSMA Operator Group Platform and has embedded methodology to place applications in edge cloudlets, based in a containerized infrastructure which offers out of the box sharing capabilities for CPU, memory and storage. In addition, the IEAP will be expanded to support Virtual Machines workloads in case the XR Enablers and applications follow such paradigm. And several mechanisms for GPU or Accelerators sharing will be analysed: PCI Passthru, Virtual GPU, Multi-instance GPU.

Regarding (ii), please refer to section 5.2.

The network interaction capability (iii) will be coordinated with the federation capability. To provide the network interaction capability, the IEAP Edge Orchestrator requires the interaction with the Beyond-5G network present at both 5Tonic and i2CAT labs. For that, the IEAP Edge Orchestrator will be integrated with the 3GPP NEF making use of the 3GPP NEF API: *Event_Exposure*, *AfSessionWithQoS* and *MonitoringEvents*, being under study the *Traffic Influence* integration.

The federation capability (iv) will be based in the GSMA East-Westbound Interface (EWBI) API that allows IEAP Edge orchestrator to establish a federation relationship with i2CAT Edge orchestrator so the XR Enablers will be able to deploy and manage seamlessly the infrastructure based on the location of the UE regardless of being hosted either in Barcelona Edge or in Madrid Edge. For that, via federation capabilities, IEAP will get to know Barcelona Edge capabilities and will decide where to deploy the application when applying the logic behind best edge selection (see more details in section 6 - Federation).

The diagram in Figure 2 below describes the different API that IEAP (MEC orchestrator in 5Tonic) will offer and its interaction with the Network:

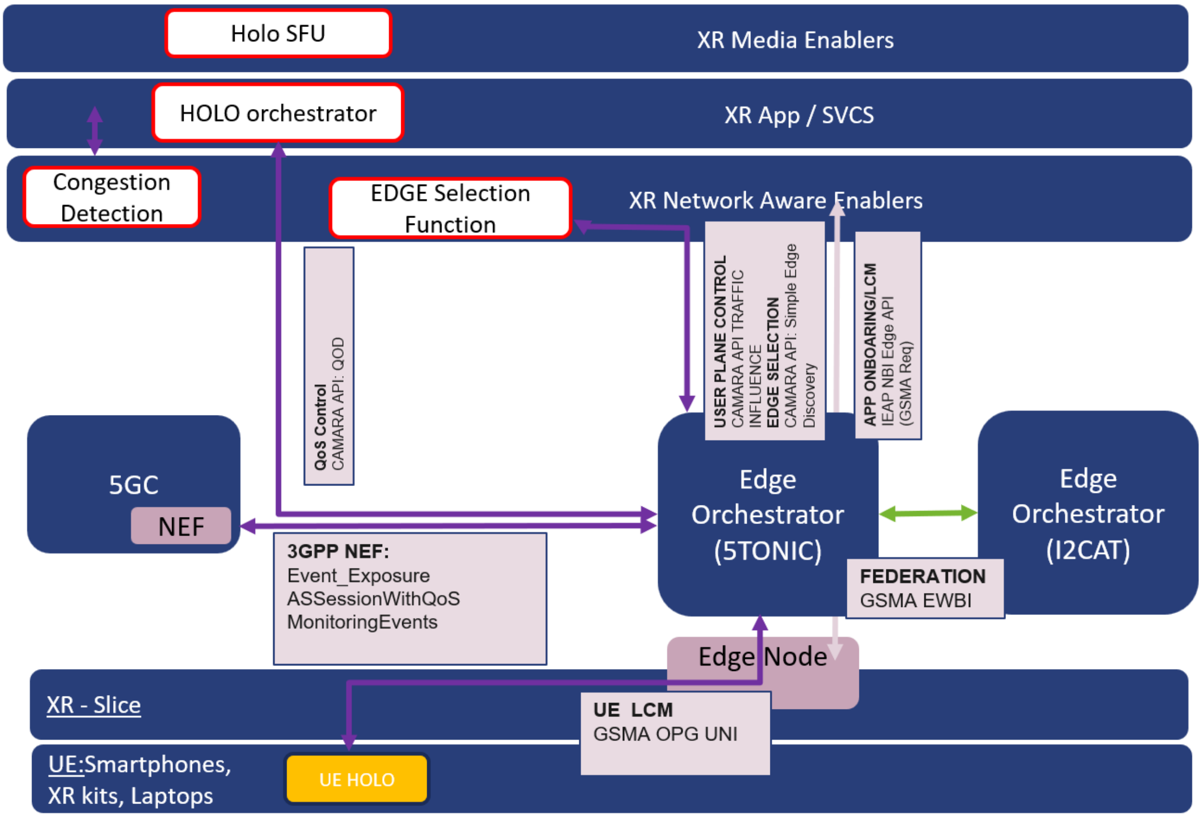

Figure 2: 5Tonic (Madrid) Edge Orchestrator interfaces

Current capacity in 5Tonic includes two servers with the characteristics collected in Table 2.

Table 2: 5Tonic Servers Characteristics

	HPE ProLiant DL360 Gen10- Controller	HPE ProLiant DL360 Gen10 - Compute
Processors	2 X Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz (10 cores, 20 threads)	2 X Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz (10 cores, 20 threads)
Memory	128GB	128GB
Networking	1 x 4 Port Broadcom NetXtreme BCM5719 Gigabit Ethernet PCIe 1 X 2 Port Intel Ethernet Controller X710 for 10GbE SFP+	1 x 4 Port Broadcom NetXtreme BCM5719 Gigabit Ethernet PCIe 1 X 2 Port Intel Ethernet Controller X710 for 10GbE SFP+
Local Storage	2 X HPE Enterprise - Hard Drive - 1.2 Tb - SAS 12GB/S Model=EG001200JWJNQ	2 X HPE Enterprise - Hard Drive - 1.2 Tb - SAS 12GB/S Model=EG001200JWJNQ

Furthermore, this capacity is planned to be expanded with two NVIDIA Tesla T4 cards (16 GB Passive, Single Slot, Full Height GPU) and at least 1 additional server with similar HW specifications as above but based in XR enablers requirements defined in WP3. Further details on the complete 5Tonic testbed will be reported later in the project in deliverable D1.3 – *Testbed Infrastructure Specification*.

3 BARCELONA EDGE CAPABILITIES

The Barcelona site has implemented an Edge platform system, designed to facilitate the deployment of diverse services near end-users. This positioning not only minimizes latency but also empowers the operator to dynamically balance loads based on operational requirements.

The illustration below (Figure 3) highlights the precise physical placement of our servers in Barcelona, strategically housed in two distinct buildings to ensure an optimal collocated infrastructure. This configuration not only enhances flexibility but also accommodates diverse operational scenarios with ease.

It is important to note that although the edge is collocated in two buildings, it functions as a cohesive unit. The RAN portion is situated in the Nexus building, together with all latency-sensitive computers/servers, while the remaining servers operate from the Omega building, ensuring operational efficiency.

Figure 3: Edge platform in Barcelona

3.1 GAP ANALYSIS

After evaluating UC1 and UC2, it is apparent that the infrastructure must be equipped with computing resources to facilitate the instantiation of diverse applications. To accomplish this, it is imperative to ensure that the Edge is furnished with resources capable of virtual application instantiation, including essential computing components such as CPU, RAM, and Disk. Additionally, the presence of a local orchestrator is indispensable for efficiently managing the instantiation of applications and overseeing their life cycle.

The South node will use CAMARA APIs across various scenarios, with a deliberate decision to restrict their availability exclusively to the primary site in Madrid. However, in certain scenarios, the vertical must have the possibility to select resources from the Barcelona site. This capability will be made possible through the Edge Federation established between the two sites, specifically in the South Node.

The Edge Platform in Barcelona should be equipped with the necessary East-Westbound Interface (EWBI) APIs, ensuring compliance with the standards set forth by GSMA in the Operator Platform Group. These EWBI APIs play a crucial role in maintaining interoperability, fostering new collaboration opportunities between the operator in the primary site in Madrid and the ones in the interconnected Barcelona site.

Considering that the edge will be hosting Extended Reality (XR) applications, the availability of a GPU is paramount, ideally one that can be shared among different applications requiring it. This ensures optimal performance for graphics-intensive tasks associated with XR applications.

Moreover, establishing a seamless communication channel is crucial for cloud-based applications to access the edge in Barcelona and gain awareness of its resources. This connectivity is vital for fostering a cohesive integration between cloud-based applications and the Edge infrastructure, facilitating efficient resource utilization and management.

3.2 INITIAL SOLUTION DESIGN

At the Barcelona Edge, key features include seamless integration with Kubernetes, leveraging a private repository for charts and application images (such as ChartMuseum or Harbor). Beyond these aspects, the edge cloud provides comprehensive information on availability zones, artifact management, application profile administration, and control over the application life cycle.

Additionally, the system offers detailed insights into application specifics, encompassing essential details such as IP addresses and Fully Qualified Domain Names (FQDN).

In terms of resources the edge platform in Barcelona provides vCPUs, RAM, and Disk, with a notable emphasis on GPU availability, which is particularly crucial for certain use cases (Table 3). Specifically, Barcelona site opted for the NVIDIA Tesla T4 GPU due to the vertical's requirements, especially for the Remote Renderer, where NVENC is a mandatory prerequisite for the application.

Table 3: Resources in Barcelona Edge

	Processor	CPUs	RAM	Disk	GPU
Edge1	Intel Xeon 6230 (2.1GHz)	2 x 20 C x 2 T = 80 Cores	256 GB	9 TB	NVIDIA Tesla T4 16 GB
Edge2	Intel Xeon Gold 5218R (2.1 GHz)	2 x 20 C x 2 T = 80 Cores	256 GB	9 TB	NVIDIA Tesla T4 16 GB

For efficient GPU resource allocation, NVIDIA MIG (Multi Instance GPU) will be employed, offering optimal virtualization for flexible assignment to processes as needed. However, it is worth noting that GPUs supporting MIG lack NVENC capabilities. Consequently, the choice of the NVIDIA Tesla T4, that supports NVENC capabilities, but at the cost of not supporting virtualization. To address this challenge, the project is exploring the implementation of GPU time-slicing within Kubernetes.

GPU time-slicing becomes possible thanks to the NVIDIA GPU Operator, which permits GPU oversubscription. This technique allows workloads to be scheduled on available slots of oversubscribed GPUs. By enabling GPU time-slicing in Kubernetes, the system can create replicas of the GPU, independently assigning them to pods for running specific workloads.

4 5GTN EDGE CAPABILITIES

In this chapter requirements from 6G-XR Use Cases and related technologies towards 6G-XR North Node, that includes two 5GTN RI facilities, University of Oulu and VTT, are handled. Use cases and Work Packages are gone through, current state and target state is described, and gaps are identified and listed. Areas that are covered here are the following: 3D Digital Twin Use Case, North Node Adapter, End-to-End Slicing and Energy Management Framework. These areas are gone through only from the perspective that they have on WP2 implementation, 5GTN networks of the University of Oulu or VTT or the core enablers.

4.1 GAP ANALYSIS

4.1.1 Facility Adapter Gap Analysis

North Node Adapter API will be implemented in T4.4 Deployment of trial controller, APIs and novel orchestration. As the component is from the Southbound API perspective interfacing with 5GTN Facility functions and by that also has requirements towards WP2 5GTN implementation it is relevant to do the gap analysis for it in this document also.

The 5GTN Facility Adapter, that was created in 5G!Drones project [3], is a crucial REST API that plays a vital role in the setup and management of network slices and KPIs Measurement Jobs within the 5GTN facility at the University of Oulu. It acts as a communication bridge between the 5G!Drones Trial Controller and underlying 5GTN resources and management technologies like Open Source MANO (OSM) to create several network slice instances. The 5GTN Facility Adapter API has also an interface to interact with Qosium Storage database for KPI collection. The 5GTN Facility Adapter API implementation is based on Python as a programming language and Django as a REST framework.

According to the 6G-XR architecture 5GTN Facility Adapter API will be basis for the implementation of the North Node Adapter API. The North Node Adapter API will used in the 6G-XR project to handle two key functions that are based on requests from the 6G-XR Trial Controller and the underlying 5GTN facility technologies (e.g., Cumucore and Qosium QoS measurement system).

Function1: The North Node Adapter API is responsible for managing the life cycle of network slices in the 5GTN facility. This includes tasks such as creating, updating, listing, and deleting network slices according to the requests received from the 6G-XR Trial Controller that are defined by the experimenter according to the network slicing capabilities in the 5GTN facility. Particularly, after receiving a network slice creation request from the 6G-XR Trial Controller, the North Node Adapter parses the received Network Slice Template (NST) and sends it in the right format to the specific orchestrator or management interfaces in the 5GTN facility. The definition of the specific parameters of the NST that is created in the Trial Controller depends on network slicing capabilities of the orchestrator or management interfaces. Importantly the NST is usually a Java Script Object Notation (JSON)-based input parameter to be provided from the 6G-XR Trial Controller to the North Node Adapter API to create a network slice at the underlying 5GTN facility. A similar process without NST is carried out when a requests to list, update or delete a network slice is received. Lastly, the creation process of a network slice is initiated by the trial execution by the 6G-XR Trial Controller. Likewise, the deletion process of the network slice is triggered by the trial finalization and decommissioning.

Function2: The North Node Adapter API oversees the life cycle of KPI measurement jobs, sequentially after the creation of a trial in the 6G-XR Trial Controller and network slice at the 5GTN facility. In this sense, it utilizes input parameters such as Trial ID, Slice ID, Facility, and KPI list provided by the 6G-XR

Trial Controller to create, list, update, and delete the KPI measurements in the 5GTN facility. Additionally, the North Node Adapter also facilitates communication with the Qosium QoS measurement system (e.g., Qosium Storage and Qosium Scope components) to automatically collect the requested network KPIs (e.g., throughput, delay, jitter, packet loss) in the 5GTN facility.

By effectively carrying out these above-mentioned actions, the North Node Adapter API ensures seamless integration and coordination between the 6G-XR Trial Controller and underlying 5GTN facility technologies, contributing to the smooth and dynamic operation, management, and monitoring of network slices and KPI collection within the 5GTN facility and 6G-XR project.

Current State:

- 5GTN Facility Adapter, that has been developed by University of Oulu as part of 5G!Drones project, already exists. It has been integrated and tested in the 5G!Drones, it has been deployed in the 5GTN production environment and it is currently functional in the 5GTN network. Software (SW) code is available.
- The 5GTN Facility Adapter can communicate the 5G!Drones Trial Controller's requests to create, list, update and delete network slices using the OSM technology at 5GTN facility. The OSM implementation enables creating, listing, updating, and deleting network slices based on Open5GS core instances deployed in the OpenStack in conjunction with Microstack [4]. The OSM Northbound RESTful Interface is following ETSI SOL005 standard. The 5GTN Facility Adapter API talks directly to the OSM Northbound Interface to create a network slice.
- The 5GTN Facility Adapter can communicate the 5G!Drones Trial Controller's requests to create, list, update and delete measurement jobs at the 5GTN facility. The measurement job triggers a background automatic process to collect and publish the requested KPI data from the Qosium Storage database for the KPI collection.
- Once a KPI measurement job request arrives, the 5GTN Facility Adapter runs a routine module to automatically publish specific requested KPIs to the KPI Monitoring Component API placed in Frequentis cloud (this handles centralized monitoring of KPIs).
- The API is documented as part of several 5G!Drones deliverables (D2.3, D3.3, D4.3, D4.4) and available at the project's deliverables website [5].
- The API uses secure authentication and authorization for communication from 5G!Drones Trial Controller and with the OSM technology.
- The API is available to operate within 5GTN facility.
- The API communicates with the 5G!Drones Trial Controller of other facilities (e.g., Aalto University) by means of an Abstraction Layer and a Virtual Private Network (VPN) connection between the University of Oulu and Aalto University.
- The API runs in a Docker container in the production environment of the 5GTN facility.
- The API software code is privately available on [6].

Target State:

- The North Node Adapter API is to be based mostly on implementation, structure, operation, and workflow of the 5GTN Facility Adapter API implemented in the 5G!Drones project.
- The API needs to support life cycle management of network slices with the underlying 5GTN facility technologies (Cumucore 5G Core solution and potentially also with Open-Air Interface-based RAN slicing capabilities).
- The API is to support communication with Cumucore's solution for network slices life cycle management at the 5GTN facility.

- The API needs to support communication with Qosium QoS measurement system to automate the KPI measurement job. For this, measurement automation using Qosium system that includes Qosium Scope controller and Qosium Storage database is to be available.
- The API must support secure communication with the 6G-XR Trial Controller through the Unified Abstraction Layer (API gateway) and using secure authentication and authorization.
- The API needs to support secure communication with the interfaces of the Cumucore's solution by using secure authentication and authorization as well.
- Initially, it is considered that the North Node Adapter API will not contain a routine module to send KPIs to one more component for centralized KPI monitoring but rather the KPI data is to be stored and monitored automatically using the Qosium system in the 5GTN facility.
- The API is to be run in a Docker container in test/production environments of the 5GTN facility.
- The API is to be integration tested to guarantee intra and inter- operability.

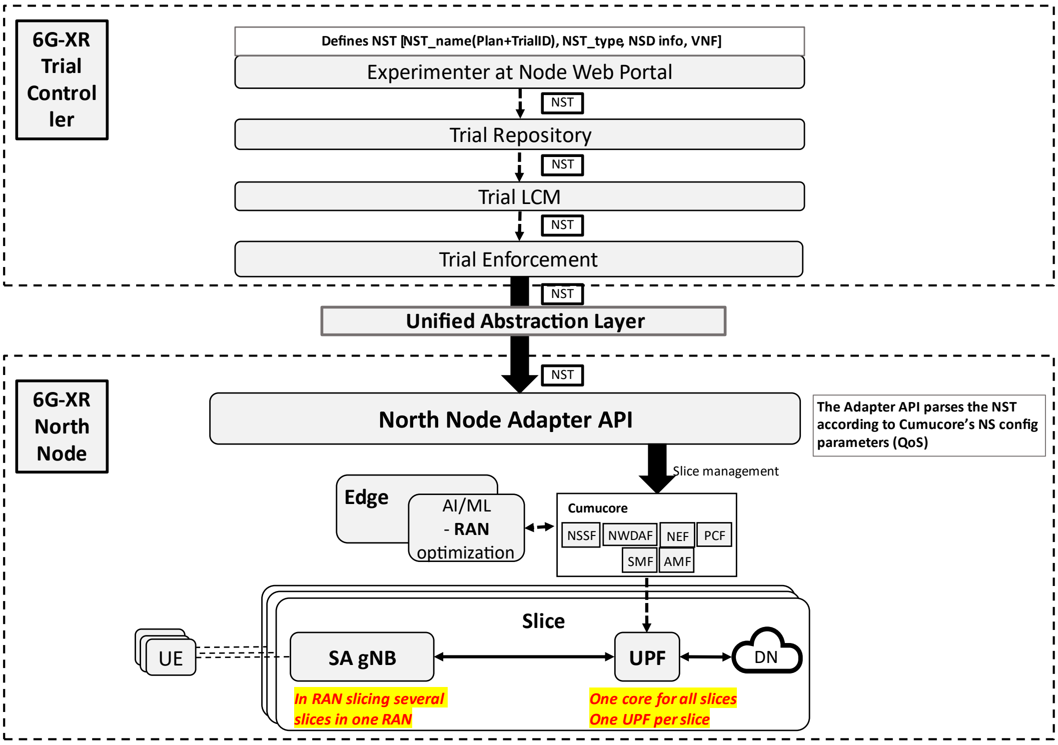

Figure 4: 6G-XR North Node Architecture with Trial Controller

Gap Analysis Summary:

Table 4 includes relevant requirements, current state of the Facility Adapter as well as the target state of the North Node Adapter API. It also includes the possible identified gap between the current and target states.

Table 4: North Node Adapter API Gap summary

Requirement	Current State (5GTN Facility Adapter API)	Target State (North Node Adapter API)	Gap
Network Slices requesting handling	Cannot handle network slicing-related requests from the 6G-XR Trial Controller and communicate them to the underlying 5GTN facility technologies	Must handle network slicing-related requests from the 6G-XR Trial Controller and communicate them the underlying 5GTN facility technologies	Implement network slicing-related requests handling capabilities by exposing interfaces to create, list, update and delete network slices according to underlying 5GTN facility technologies
Measurement jobs requesting handling	Cannot handle KPI measurement jobs-related requests from the 6G-XR Trial Controller and provide them to the underlying 5GTN facility technologies (e.g., Qosium system)	Must handle KPI measurement jobs-related requests from the 6G-XR Trial Controller and provide them to the underlying 5GTN facility technologies (e.g., Qosium system)	Implement KPI measurement jobs-related request handling capabilities by exposing interfaces to create, list, update and delete KPI measurement jobs in the 5GTN
KPI data publishing routine	Centralized KPI monitoring	Each facility must monitor and store automatically its own KPIs using Qosium system	Create a mechanism to provide the required information to the Qosium system for the automation of the KPI monitoring in the 5GTN facility
Containerized API	One container deployment available of the 5GTN Facility Adapter API from 5G!Drones	North Node Adapter API must run in a Docker container in a test and then production environment	Create a Docker container for the functional North Node Adapter API and deploy in a test and then production environment at the 5GTN facility
Secure communication with 5GTN technologies components	No communication with underlying 5GTN facility technologies	Secure authentication and authorization with underlying 5GTN facility technologies	Implement, verify and test secure authentication and authorization with underlying 5GTN facility technologies
Testing	5GTN Facility adapter has been integration tested in 5G!Drones project	Perform integration testing for Node North Adapter API	Conduct integration testing
Documentation	Basic documentation	Comprehensive documentation	Create comprehensive documentation

4.1.2 3D Digital Twin Gap Analysis

3D digital twin is an advanced system that synchronizes physical and digital assets within a virtual online space. This virtual environment includes highly detailed 3D representations of rooms, human avatars, and machines. The creation of these immersive 3D Virtual Reality (VR) scenes is powered by high-performance Graphics Processing Unit (GPU) computing.

6G-XR 3D digital twin use case is composed of 3D digital twin environment of FabLab at 5GTN Edge computer. VR scenes are produced by Unity 3D engine located in a high-performance GPU processing enabled Edge server. In the 3D digital Twin use case the VR host creates a virtual Fab Lab room and spawns avatars (remote user and instructor) into it. The VR host is located at the Edge server of 5GTN and VR user logs in as a client. To ensure secure and efficient communication within 6G-XR 3D Digital Twin environment, the system employs Stand Alone (SA) 5G network in 5GTN. For industrial class robot arm control that is utilized in the use case, Time Sensitive Networking (TSN) is integrated into 5GTN. TSN technology guarantees real-time data transfer with deterministic communication, that is essential for maintaining a seamless and interactive experience synchronizing the physical and digital assets. Required KPI measurement is automated in each function of the environment to verify the network and use case performance.

Current state:

- Fab Lab 3D digital twin does not exist.
- SA 5G network is not available at the University of Oulu Fab Lab premises.
- Unity 3D engine is not integrated and available at 5GTN.
- No GPU powered Edge computing available in 5GTN.
- TSN switch not integrated and available in 5GTN.
- There is no WiFi TSN implemented in 5GTN.
- Automated KPI Measurement system is not available for 3D Digital Twin use case.

Target state:

- 3D digital twin of the University of Oulu Fab Lab is created.
- SA 5G network is available at University of Oulu Fab Lab.
- Unity 3D engine is integrated into the 5GTN Edge node.
- GPU powered Edge computing (PC or server) is available in 5GTN.
- TSN ready network exists in 5GTN.
- WiFi TSN is integrated with robotics arm robot to 5GTN and made available in the University of Oulu FabLab.
- Automated 3D digital twin KPI measurement system is available.

Gap analysis summary:

Table 5: 3D Digital Twin Gap Analysis Summary

Requirement	Current State (5GTN Facility)	Target State (North Node)	Gap
Fab Lab 3D digital twin	Does not exist	Integrated and available in 5GTN	Integrated Fab Lab digital twin environment

SA 5G availability	Not available in the University of Oulu Fab Lab	5G SA connectivity available in Fab Lab	Make SA 5G available in the University of Oulu Fab Lab
Unity 3D engine	Does not currently exist	Fully integrated into an 5GTN Edge node	Integrated Unity engine available at 5GTN Edge node
GPU powered computing	No GPU enabled computing node available in 5GTN Edge	Edge node with GPU enhancement available in 5GTN	Edge node with enough GPU performance to fulfil 3D digital twin KPI requirements
TSN switch	Does not currently exist	TSN enabled in 5GTN	TSN switch is integrated into 5GTN
WiFi TSN	Not available in 5GTN	TSN integrated to 5GTN	WiFi TSN is integrated into 5GTN and available in FabLab to be used with robotic arm
KPI measurement	Automated test system does not cover 3D digital twin	Automated test system exists	Automated test system enabled for 3D digital twin KPI measurements

4.1.3 Energy Management Framework Gap Analysis

The North Node (UOULU and VTT) 5GTN is providing an energy measurement framework to develop and deploy End-to-End (E2E) energy management and energy conservation technologies to transform the Next Generation Node B (gNB) site towards off-grid powering systems using local energy-weather forecast, local batteries, local renewable energy sources, and power saving measures.

Current state:

The current energy measurement framework for the North Node 5GTN site is basic, only capable of tracking overall power consumption across the site using an API monitoring system. It lacks detailed data on energy consumption per individual component, and it does not use machine learning (ML) or deep learning (DL) algorithms to optimize energy usage. Additionally, the network components, such as gNBs are outdated, and there are no edge servers, Universal Software Radio Peripheral (USRP), or an OAI environment that could aid in developing and implementing energy-efficient algorithms.

Target state:

The goal for the energy measurement framework is to collect detailed data on energy consumption for each network component and then utilize ML/DL algorithms to optimize energy usage. The 5GTN will be upgraded with the latest gNBs and with the latest software upgrades, new system modules and radio heads. OAIBOX, Open Air Interface (OAI) RAN and Open 5GS core will be used to deploy the OAI environment. Additionally, edge servers and USRP will be deployed, enabling the development of more sophisticated and energy-efficient energy management strategies.

Gap Analysis Summary:

Table 6: Energy Management Framework Gap Summary

Requirement	Current state	Target state	Gap
Outdated gNBs	Network components, such as gNBs are outdated	Upgraded with the latest gNBs with the latest software upgrades, new system modules and radio heads	5GTN architecture upgrade for gNBs, relevant software features and modules
Lack of edge servers and USRPs	No edge servers	Deploy edge servers	Deploy edge servers to enable the development of more sophisticated and energy-efficient energy management strategies
Lack of OAI environment	There is no OAI environment	OAIBOX, OAI RAN and Open 5GS core will be used to deploy the OAI environment	Make OAI environment available to enable implementation of energy measurement framework using OAIBOX, OAI RAN, 5GS Core and USRP radios

4.1.4 End-to-End Slicing Gap Analysis

E2E network slicing encompasses RAN slicing, Transport Network (TN) slicing and Core Network (CN) slicing. To make the dynamic E2E network slicing environment, together with these three basic communication network components, some other network slicing elements are also crucial. These elements are Network Slice Instance (NSI), Service Level Agreement (SLA), service orchestration, resource allocation, isolation mechanism, etc. Before mentioned technology areas are all to be sliced or used in the creation of slices other than Transport Network slicing. It is not seen necessary for the 6G-XR use cases as the needed network QoS is easy to guarantee in a private 5G network like 5GTN. In 6G-XR project, there will be two different E2E network slices implemented in the 5GTN facility, namely Ultra Reliable Low Latency Communication (URLLC) and enhanced Mobile Broadband (eMBB). UE's that will used needs to be slicing compatible.

RAN Slicing: This involves creating virtual slices of the RAN to provide dedicated resources and configurations for different use cases. RAN slicing relies on technologies like Network Function Virtualization (NFV) and Software Defined Networking (SDN) to create virtualized radio access nodes (V-gNBs) that can be dynamically configured and provisioned.

Core Network Slicing: The core network includes the elements responsible for processing and transporting user data. It consists of functions such as the UPF, several Control Plane Functions (CPFs), and other network functions that collectively enable communication within the slice. The key component in the CN for slicing is called Network Slice Selection Function (NSSF) which contains one or more Single Network Slice Selection Assisted Information (S-NSSAI). An S-NSSAI has two parts: Slice Service Type (SST) which is a mandatory filed and Slice Differentiator (SD) which is an optional part. Network slicing parameters are configured inside the Access and Mobility Management Function (AMF) and NSSF components of the CN.

Service Orchestration: It is responsible for managing the life cycle of slices, including creation, provisioning, activation, and deactivation. ETSI-OSM is a collaborative open-source project to develop NFV Management and Orchestration stack aligned with ETSI NFV Information Models and APIs. It is a set of tools and APIs that allow to manage and orchestrate the NFV infrastructures. It can handle the creation and management of multiple virtual network slices on top of a single physical infrastructure by supporting LCM, resource allocation, slice isolation mechanisms, and policy management.

Resource Optimization: After the sliced 5G network has been set up and is operational with different slices, there is a need to re-allocate slice resources dynamically. The intention is to allocate the sliced 5G radio interface resources to achieve an optimum configuration based on the needs of the service or application. As an option there is also the possibility to adjust, for example, the energy parameters of the radio resources, such as cell transmit power or other parameters.

Current state:

- **RAN Slicing:** 5GTN has both commercial and open-source (Open Air Interface: OAI) RAN solutions to be used. Currently in 5GTN, the network cannot dynamically allocate the radio resources to the slices as per the SLA. Based on the pre-configured slice, it uses all the available resources from the network and that has been practiced in the previous 5G!Drones project.
- **CN Slicing:** The open-source CN used in 5GTN facility is called Open5GS 5G Core (5GC). 5GTN site has the capability to implement CN slicing using the 3GPP defined parameters.
- **Service Orchestration:** For orchestration management purposes, ETSI OSM together with MicroStack [4] version of OpenStack as a Virtualized Infrastructure Manager (VIM) have been used in the 5GTN infrastructure. OSM handles creating, modifying, deleting, and listing network slices together with MicroStack in the 5GTN facility.
- **Resource Optimization:** Currently there exists no means to adjust the resource allocation of a sliced 5G network.

Target State:

- **RAN Slicing:** Dynamic RAN slicing is to be implemented in the 5GTN facility. Different slices will be allocated with the required radio resources according to the given SLA. Here one gNB aims to support several slices at the same time. Resource allocation in the RAN needs to be enabled and the resources are to be dynamically reconfigurable.
- **CN Slicing:** Control and user plane functions will be implemented in the 5G Test Network for different slices to fulfil the requirements for multiple simultaneous slices to be supported by the 5G core.
- **Service Orchestration:** Service orchestration for full dynamic slice life cycle management is to be enabled in the 5GTN. Slices are to be dynamically created, re-configured, listed and terminated. or Service Orchestration, North Node Adapter API will be communicating with the Cumucore 5GC solutions for creating, activating, modifying, deleting slices as well as the slices life cycle management.
- **Resource Optimization:** AI/ML based algorithm will be used to optimize the radio interface resources in the sliced 5G network and to re-allocate them in an optimal way.

Gap Analysis summary:

Table 7: End-to-End Slicing Gap Summary

Requirements	Current State	Target State	Gap
RAN Slicing	Cannot handle RAN resource allocation dynamically to the target slices	Must handle dynamic resource allocation for RAN slicing	Implementation of RAN slicing resource allocation mechanisms in the facility
CN Slicing	No capability to support slicing	Slicing support to be enabled in the 5G Core	Need for a 5G Core supporting slicing in the 5GTN
Network Slicing Resource Allocation	Current OSM setup cannot handle resource allocation in different slices	Resource allocation must be supported by the 5G Core for slicing	Ability to allocate resources efficiently for different slices is missing
Slice Isolation	Slice isolation does not exist	Must enable slice isolation capability in the 5GTN	Slice isolation is missing
Policy Management	Slice policy management is missing	Policy management available	Implementation of slicing policies in the 5GTN
Resource Optimization	No method to adjust sliced 5G network radio interface resources	AI/ML controlled optimization of the radio interface resources	Implementation of AI/ML algorithm to re-allocate the sliced 5G network radio interface resources in an optimal way

4.2 INITIAL SOLUTION DESIGN

This chapter includes the functional description for all the enablers defined in chapter 4.1 Gap Analysis. For some of the enablers, where it is needed, also high-level architecture definition is included.

4.2.1 North Node Adapter API Functional Description

Based on the provided gap analysis in chapter 4.1, the solution design for the North Node Adapter API should focus on addressing the identified gaps to support 6G-XR Trial Controller requests:

The solution design consists basically of the following plan:

1. Implement network slicing-related requests handling capabilities by exposing 5G core interfaces to create, list, update and delete 5G network slices according to available 5GTN facility technologies:
2. Implement KPI measurement jobs-related request handling capabilities by exposing automated Qosium based test system interfaces, capabilities and data formats to create, list, update and delete KPI measurement jobs in the 5GTN facility.
3. Create a mechanism to provide the required information to the Qosium system for the automation of the KPI monitoring at the 5GTN facility:
    a. Define the information required by the Qosium system to automate the KPI monitoring.

b. Define the target API endpoints from the Qosium system to provide the requested information from 6G-XR Trial Controller and the required input parameters to talk with.

4. Create a Docker container for the functional North Node Adapter API and deploy in a test and then production environment of the 5GTN facility:
    a. Deploy the API as a Docker container in the 5GTN facility.
5. Implement, verify, and test secure authentication and authorization with underlying 5GTN facility technologies:
    a. Implement communication between the North Node Adapter API and the underlying 5GTN facility technologies (Cumucore and Qosium).
    b. Integrate the required authentication and access control mechanisms.
    c. Test and validate the secure communication.
6. Conduct integration testing:
    a. Perform integration testing to assess the API's ability to handle the requests from the 6G-XR Trial Controller to underlying 5GTN facility technologies.
    b. Conduct testing to identify potential errors.

4.2.2 3D Digital Twin Enablers Functional Description and Architecture

Below in Figure 5 3D Digital Twin high level architecture is presented. It has three main building blocks: 5GTN, remote location and Fab Lab. Remote place presents the location where the remote user is connecting wirelessly to 5GTN wearing VR glasses. 5GTN has the needed 5G network and Edge computer where the 3D engine running the Digital Twin is located. Fab Lab then presents the real-world Fab Lab, 3D Printer, cameras etc devices. Also, the Instructor is physically present in the Fab Lab.

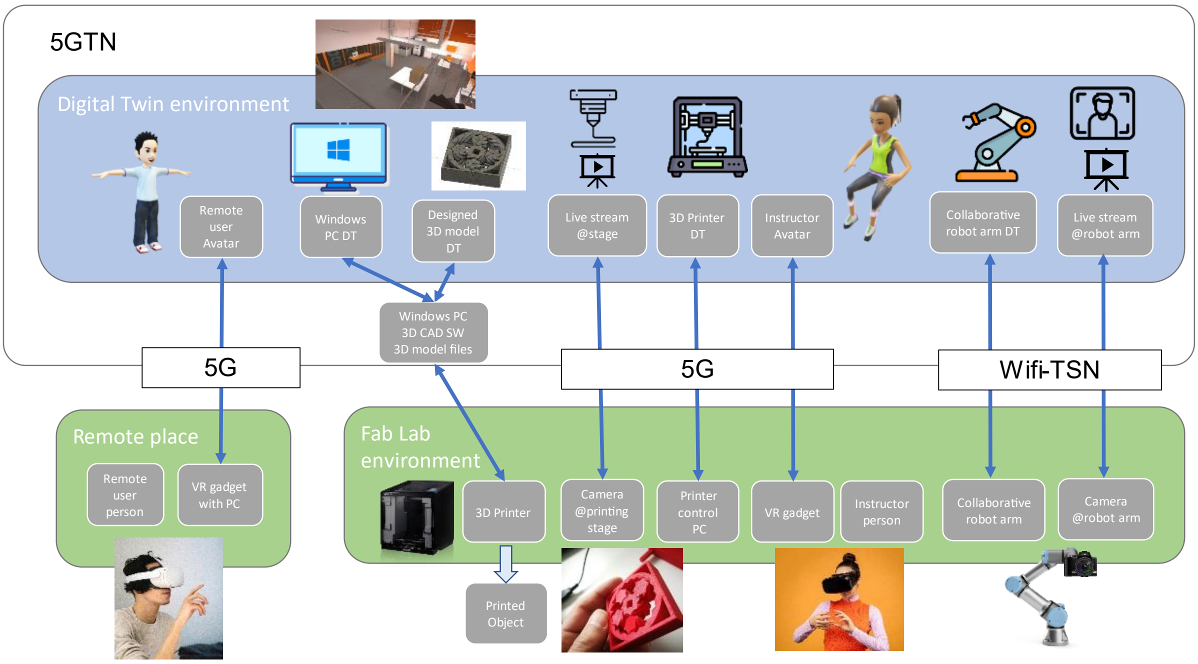

Figure 5: 3D Digital Twin Architecture

The solution design consists basically of the following plan:

1. 3D Digital Twin of the University of Oulu FabLab is made available in the Unity 3D engine as VR collaborative workspace.
    a. Fab Lab is available as a VR room.
    b. Digital mirror of real-world 3D printer and control computer is created into the virtual FabLab.
    c. VR multi-play and user interaction functions are available to manipulate object to be printed and also to manage the real-world 3D printer.
2. SA 5G availability.
    a. 5GTN SA 5G is to be made available in the FabLab.
3. Unity 3D engine application.
    a. 5GTN Edge node is made available so that Unity 3D engine host application can be installed on it.
4. GPU powered computing.
    a. GPU card that fulfils the 3D Digital Twin KPI requirements is available in the 5GTN Edge node where Unity engine will be installed.
5. TSN switch that provides Time Sensitive Networking functionality, to minimize the jitter by deterministic communication with accurate timing.
    a. TSN ethernet switch is to be integrated into 5GTN to provide the needed TSN functions
    b. Accurate clock timing source (Global Positioning System (GPS) or atomic clock) is made available for the TSN switch.
    c. Time synchronization and traffic shaping of TSN functions (IEEE 802.1AS, IEEE 802.1Qbv) is available in the TSN switch and in Wifi-TSN (machine end TSN terminal).
6. Wi-Fi TSN.
    a. Wi-Fi TSN is integrated with the controller of robotic arm in Fab Lab.
    b. Wi-Fi TSN can be configured with Time synchronization and traffic shaping of TSN functions (IEEE 802.1AS, IEEE 802.1Qbv).
7. KPI measurement.
    a. Qosium probes are to be set at 5GTN node to fulfil the measurement points of KPI requirements.
    b. Qosium automated measurement system is to be enabled for 3D Digital Twin with relevant measurement parameters that fulfil KPI requirements.

4.2.3 Energy Management Framework Enablers Functional Description and Architecture

Based on the above gap analysis summary provided in Chapter 4.1, the solution design for implementing energy measurement framework for the North Node 5GTN site should focus on addressing the following identified gaps. Figure 6 below shows the target architecture of the 6G-XR Energy Management Framework System.

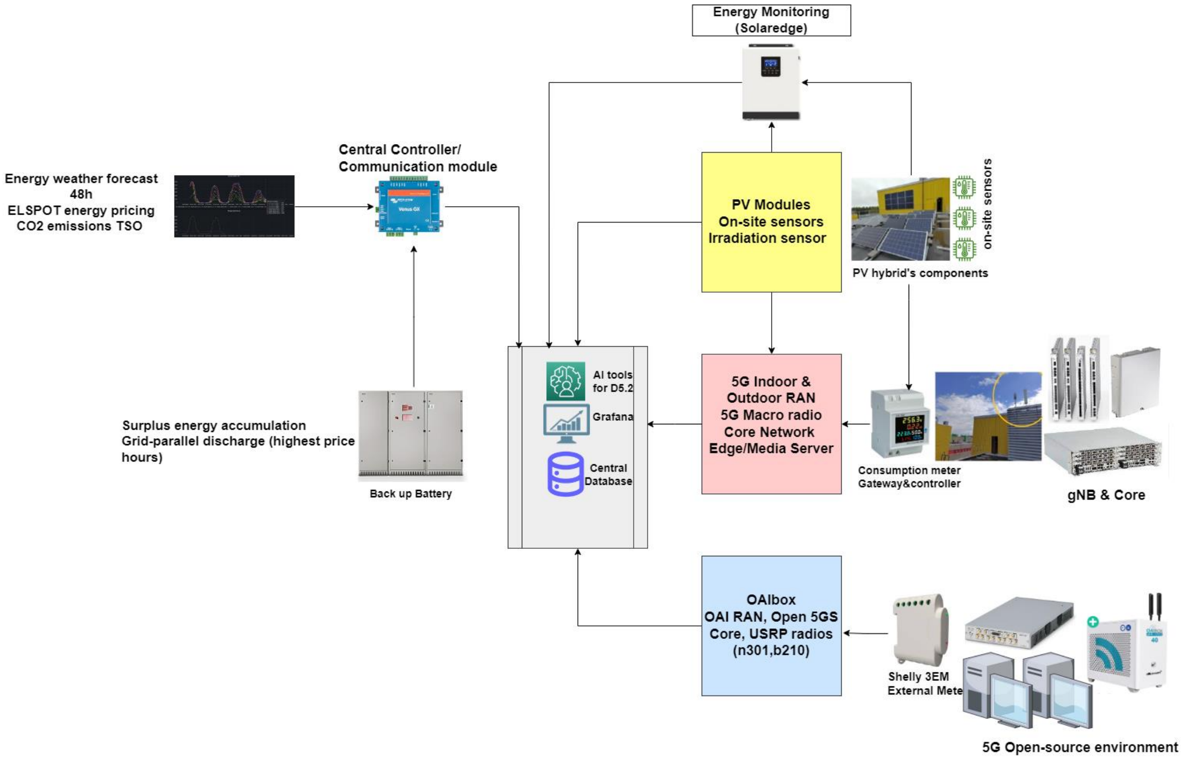

Figure 6: Energy Management Framework System

The solution design is:

- The gap analysis summary will be addressed in future 6G-XR's D5.1 comprehensively while addressing each gap in detail.
- Integrate Edge servers into the 5GTN network.
- Configure, Integrate and test OAIBOX as part of the 5GTN with needed USRP radio. For details see Chapter 9.
- Update outdated gNB's in the 5GTN with new ones.
- Install relevant software features for all the above-mentioned devices/modules.

4.2.4 End-to-End Slicing Enablers Functional Description and Architecture

Solution to enable End-to-End slicing support in 5G Test Network of the University of Oulu is to introduce 3GPP compliant 5G core that can support dynamic slicing with different suppliers gNB's. For this Cumucore 5G Core will be taken into use in 5GTN during 1Q/2024. Cumucore supports the slice life cycle management, RAN slicing capability and core network slicing requirements. These requirements are supported through thee Cumucore API that is to be integrated together with the North Node Adapter API.

- RAN slicing will be implemented to enable dynamic resource allocation to the slices. Two slices (eMBB and URLLC) will be configured in the RAN, and resources will be allocated for them according to the SLA.
- CN slicing will be offered by the introduction of the Cumucore in the 5GTN facility. Two slices will be configured in the core network. Each slice will have a dedicated UPF and common CPF of the 5GC network.

- Slice level resource allocation will be implemented using possibilities offered through Cumucore API.
- Slice isolation will be designed and implemented to ensure the confidentiality, integrity, and availability of traffic.
- Slicing policies such as QoS, security etc. are to be defined and implemented in the E2E slicing using the Cumucore APIs.
- AI/ML algorithms will be implemented to optimize the sliced 5G network radio interface resources in an optimal way. SW can be implemented as an independent application, or it can be implemented as an xApp to utilize xApps technology running in Near-Real-Time (Near-RT) RAN Intelligent Controller (RIC).

Figure 7 below describes the high-level architecture for the End-to-End slicing introduced in the North Node.

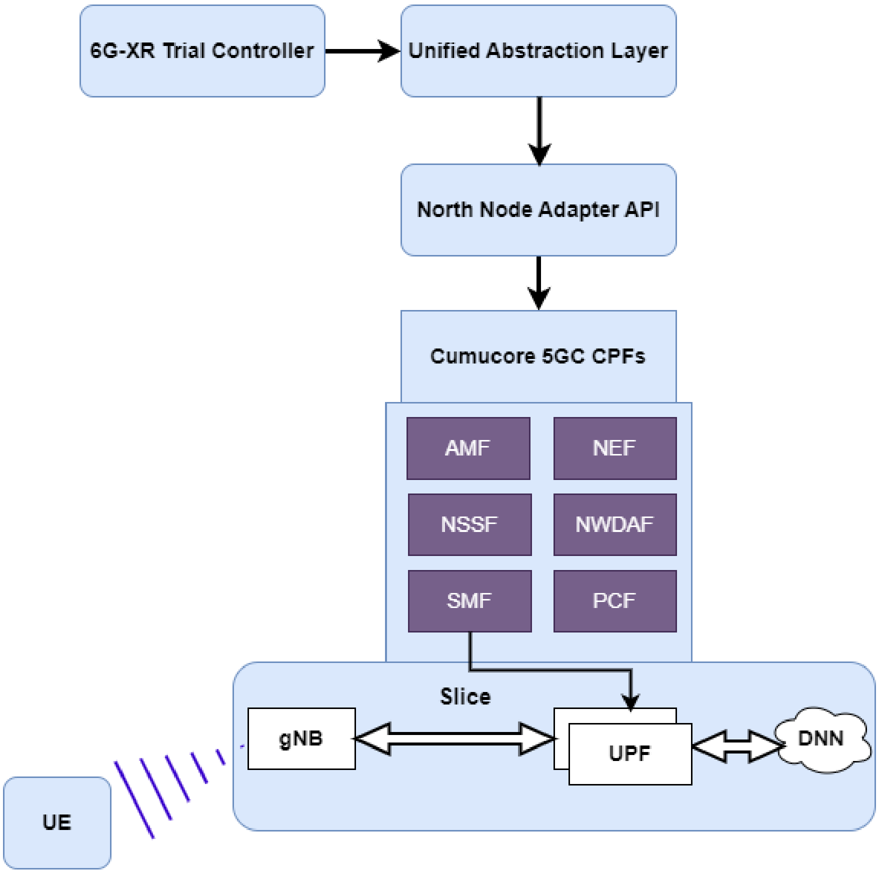

Figure 7: E2E network slicing architecture/solution design in the 5GTN facility

5 EDGE NORTHBOUND INTERFACE (NBI) API

As part of the 6G-XR ambition and use cases, the development beyond state of the art of Network as a Service (NaaS) enablers in the South Node will allow XR Enablers to properly interact with the Edge and the network. By means of the Edge NBI, the XR Enablers will be able to require and manage compute resources on-demand in the optimal and most efficient location to fulfil certain compute requirements and network requirements or Quality of Service for each application.

5.1 GAP ANALYSIS

The use case and functional requirements related specifically to the Edge Northbound Interface (NBI) from 6G-XR UC1 *Resolution Adaptation or Quality* and UC2 *Routing to the Best Edge* collected in [1] are gathered in Table 8:

Table 8: 6G-XR use cases requirements relevant for Edge NBI API

Requirement No.	Requirement Description
UC2.R1	Edge onboarding capabilities, which involves being able to deploy a virtualized media function (container-based) on the Edge – renders on behalf of the user.
UC2.R2	Edge discovery, which involves that the network provides a subset of Edge locations where service (is/can be made) available.
UC2.R3	Traffic Influence allows UPF (breakout) reconfiguration. UPF selection is decided by the network, but triggered by the service.
UC2.R4	Load mobility: If the service metrics degrade, then an Edge Discovery process is triggered, and a new Edge can be selected. App is re-instantiated with status migration.
UC1-3.FR23	The XR Holo Orchestrator shall interoperate with an Edge Orchestrator to request the dynamic instantiation/termination of virtualized networked media functions on the edge and cloud based on demand.
UC1-3.FR25	The XR Holo Orchestrator shall be able to request the network to provide Quality on Demand (QoD) for a specific holographic session or to specific XR clients.

Based on the further elaboration of the requirements it can be stated that **UC1 and UC2 are demanding from Edge NBI in the South Node to provide means to**:

- Control the life cycle of the applications so resources are only assigned when and where needed.
- Offer the capability to provide a list of edge candidates for the application to choose according to a set of requirements and based on the network.
- Allow UPF reconfiguration, decided by the network, but triggered by the service.
- Request to the network a concrete Quality on Demand (QoD) for a specific holographic session or XR enabler.
- Offer the capability to control Quality of Service provided to these Apps and flows.

- Request XR enablers to adapt the resolution based on network congestion.

Among other initiatives, Linux Foundation CAMARA Project [7] is advocating for a set of APIs with the aim to support the transformation of Operators Networks into Service-Enablement platforms by providing a set of APIs abstracting the underlying network complexity facilitating integration with applications.

GSMA is working closely with CAMARA Project to align requirements in the different GSMA initiatives such as GSMA Operator Group (OPG) [2], looking to ensure:

- A simplified and user-friendly API
- Data Privacy and Regulatory Requirements enforcement
- Application to network integration

This approach will provide 6G-XR XR enablers with the toolset required to control and integrate with the South Node network via the Edge capability, which will implement the API and adhere to GSMA OPG principles.

At least the following CAMARA APIs will be considered:

- From the Quality on Demand – Mobile subproject: *Qod-api*
- From the Edge Cloud subproject: *Simple-edge-discovery*, *MEC-experience-mgmt*, and *Traffic-influence*

Here below an example from official Swagger of a QoD call flow is depicted in Figure 8. For further details refer to [8].

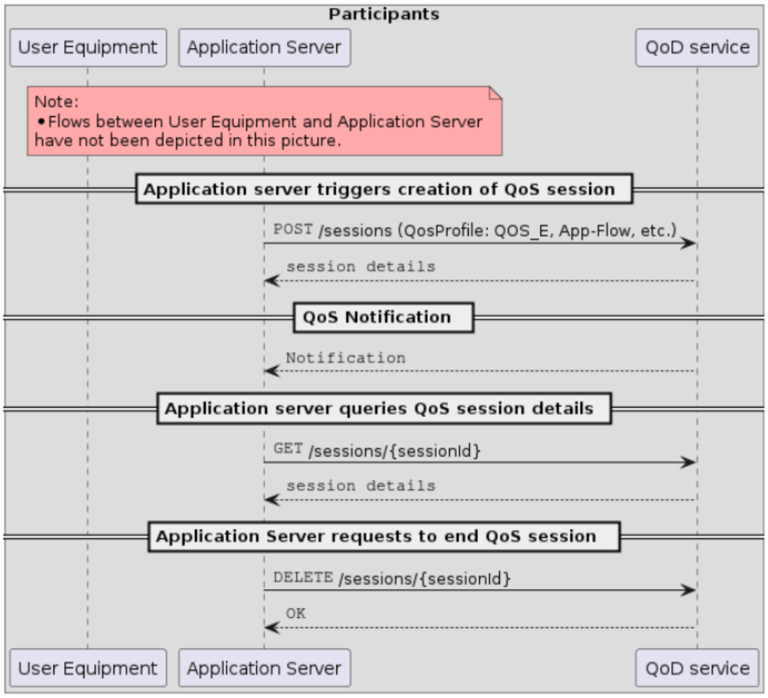

Figure 8: Example of a CAMARA QoD API call flow

The specification status of these APIs is heterogeneous; in most of the cases they are still work in progress. 6G-XR will be compliant with the current specification and in some cases will be going ahead contributing to the CAMARA API specification as it is explained in section 5.2.

5.2 INITIAL SOLUTION DESIGN

6G-XR is actively collaborating with Linux Foundation CAMARA API, more concretely within the Edge-Cloud workgroup and related NBI API that will be used as part of UC2. These NBI API will facilitate to XR enablers seamless usage and customization for both the RAN and CORE network and the Edge capabilities. The NBI API considered will comprise:

- *Quality On Demand* NBI API will be integrated to facilitate the XR Enablers to control the Quality offered to the UE on demand (UC1)
- *Simple Edge Discovery* API for the selection of the best edge serving the UE to run the XR Enablers (UC2)
- *Traffic Influence* to facilitate the adaptation of the network upon a change on the status of the UE in order to optimize the routing (UC2)

Besides, XR Enablers will use Capgemini's application Life Cycle Management API called IEAP NBI Edge API, which is implementing and following GSMA OPG Northbound Requirements, as well as GSMA Southbound User Network Interface.

5.2.1 Quality on Demand API Functional Description

This API would be aligned with the Quality on Demand CAMARA API that is looking to allow applications to set quality requirements (e.g. required latency, jitter, bit rate) using a predefined set of QoD levels as well as to get notifications in case such quality level could not be fulfilled by the network.

The main functionalities of this API in 6G-XR will comprise:

- To set quality for a mobile connection (this is required latency, jitter, bit rate) for XR application.
- Create new session with QoS (through IEAP 5G Core Adaptor to NEF).
- To get notification if network cannot fulfil such quality level (get session details through IEAP 5G Core adaptor from NEF).
- Upon a network congestion in a cell where the XR (UE) device is connected QoD API will be invoked to prioritize XR flows.

This capability will be used by XR enablers in UC1 to request new QoD Profiles via the CAMARA API and Edge Capability to the Network which will modify accordingly the Network Slice.

5.2.2 Edge Cloud Discovery APIs Functional Description

It comprises functionalities of two CAMARA APIs: i) Simple Edge Discovery and ii) MEC Experience Management and Exposure, and it will enable the service to select and make use of the most appropriate Edge based on specific goals.

It will be used by XR enablers to control the selection of the best edge in order to:

- Request Edge Platforms info for App instantiation in a predetermined PoP (business decision based on multiple criteria – region, requirements, etc).

- Request a list of available edges platforms based on UE location (UE's IP) to serve a determined attached UE.
- Request to instantiate an App from the XR Enabler itself to a certain Edge Cloudlet from the provided lists.

This API definition is currently being elaborated within the CAMARA Edge Cloud workgroup. Several 6G-XR partners are contributing to the standardization efforts with inputs of the Use Cases required. Capgemini has provided a concrete contribution to this CAMARA API code based on 6G-XR work named Alignment of the simple edge discovery API as per the CAMARA commonality group guidelines [9].

While these definitions are being refined and finalized, IEAP in 6G-XR project will provide part of those features using its proprietary implementation of the GSMA Operator Group NBI Requirements [10] which fulfil the same intent.

5.2.3 Traffic Influence API Functional Description

This API aims to provide the developer with the possibility of having the minimal E2E latency towards the Edge Cloudlet by activating and configuring the proper routing in the mobile network and in this way influencing the traffic routing from the user device towards the Edge instance of the application.

It will provide 6G-XR enablers the means to request upon UE migration to a new network to find the best network routing to reach the new Edge Cloudlet in the new network. Traffic Influence API should enable to select for the application the best Holo SFU instance depending on the delay being experienced.

Therefore, the main functionality of this API will comprise:

- Request edge platform for the optimal routing (minimum delay) upon a new UE location (UE's IP).
- Communicates list of Edge Platforms based of the new UE location.

The API definition is currently being elaborated within the CAMARA Edge Cloud workgroup. It is expected that 6G-XR project will be contributing to its definition.

6 FEDERATION

Federation stands out as one of the key facilitators within the 6G-XR project, enabling use cases, e.g. UC2. In this instance, a vertical gain the capability to access a broader spectrum of resources for instantiating applications, thanks to the collaboration and federation of edges across multiple infrastructure owners. This federated agreement empowers the vertical to strategically choose the optimal edge for application instantiation, enhancing flexibility and resource optimization. Note that the federation's scope is confined to the South Node, specifically established between the edges in Madrid and Barcelona.

6.1 GAP ANALYSIS

Complex or more intricate scenarios may require that verticals have access to resources from multiple operators, to select the most optimal options for instantiation at any given moment. The initial step involves ensuring seamless inter-Edge system communication, requiring the following key assumptions:

- Edge to Edge inter-system discovery and communication.
- Edge to Edge inter-system communication across platforms.

This addresses the basic communication aspect between systems, which is inherently complex and necessitates also complex business agreements. However, beyond this lies the challenge faced by the vertical attempting to utilize the system. Presently, and until a standardized framework is fully defined, the situation is somewhat complicated, presenting various scenarios where a vertical may want to leverage resources from different operators. In most cases, the vertical may find itself navigating the diverse systems individually.

In the next chapter, we will explore ongoing efforts to address the mentioned issues and identify the key stakeholders involved in shaping a standard for Edge federation.

State of the art

Propelled by a consortium of multiple operators, GSMA is actively advocating for the integration of edge computing as an operator service. This initiative aims to ensure that, from the customers' perspective, the utilization of an edge application remains seamless, irrespective of whether the application is executed on their operator's edge cloud or on the edge cloud of another operator.

GSMA has established the Operator Platform Group (OPG) to unite operators, platform developers, edge cloud providers, and other influential stakeholders within the industry. This collaborative group is dedicated to crafting the Operator Platform, which outlines a standardized platform for presenting operator services and capabilities to customers and application providers. Initially centered around edge computing, the scope of this initiative has since broadened to encompass capabilities related to connectivity and network slicing.

Within the OPG, a significant milestone has been the definition of the East-Westbound Interface (EWBI) APIs as part of the Operator Platform. These APIs play a key role by enabling:

- Federation of functionality across networks, allowing application providers to seamlessly connect to a singular platform and access capabilities from multiple networks.

- Sharing of edge platforms between networks, facilitating the provision of end-user services with access to a broader range of resources beyond those of the main operator, including those established through federation with other operators.
- Delivery of services to end users roaming outside their home network's footprint, ensuring continuity and accessibility of services across different networks.

Therefore, the GSMA's objective is to actively collaborate with standardization and open-source communities tasked with defining the Operator Platform (OP) standards. In this context, the pivotal role played by ETSI becomes apparent, as it endeavors to establish a robust standard through its work in the ETSI ISG MEC. This effort is strategically aligned with the ongoing initiatives within the GSMA Operator Platform Group (OPG).

Figure 9 represents ETSIs current system reference architecture for MEC, including MEC federation [11]. In the picture, the main synergy between the ETSIs proposal (Mff) and the OP EWBI (OP: E/WBI) is highlighted in red.

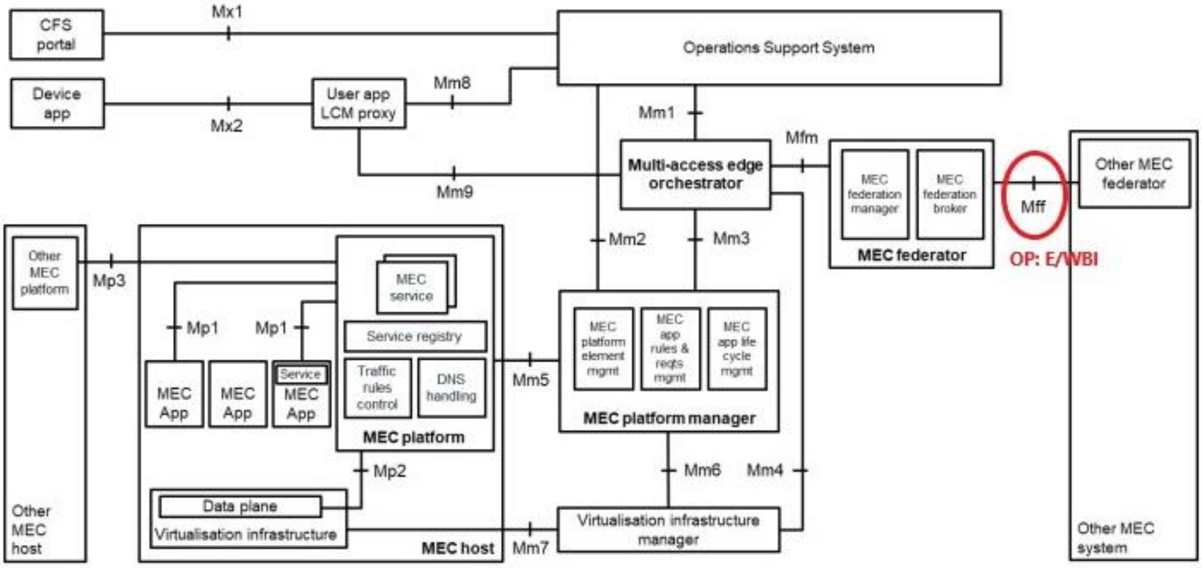

Figure 9: Multi-access edge system reference architecture variant for MEC federation.

All these interfaces are described in more detail in the ETSI White Paper MEC federation: deployment considerations [12], but for the purpose of the project the most relevant interface is the EWBI Interface.

Hence, in the deployment and orchestration of Edge systems, whether undertaken by developers or operators creating federated Edge systems, it is imperative that the resulting federation complies not only with ETSI MEC specifications (and 3GPP SA6) but also aligns seamlessly with the GSMA OPG requirements [10]. Compatibility is equally essential, with the requirements set forth by other influential industry organizations, such as 5GAA.

6.2 INITIAL SOLUTION DESIGN

6.2.1 Edge Federation Functional Description and Architecture

For UC2, which involves the validation of Edge Federation, 6G-XR project proposes the architecture illustrated in Figure 10.

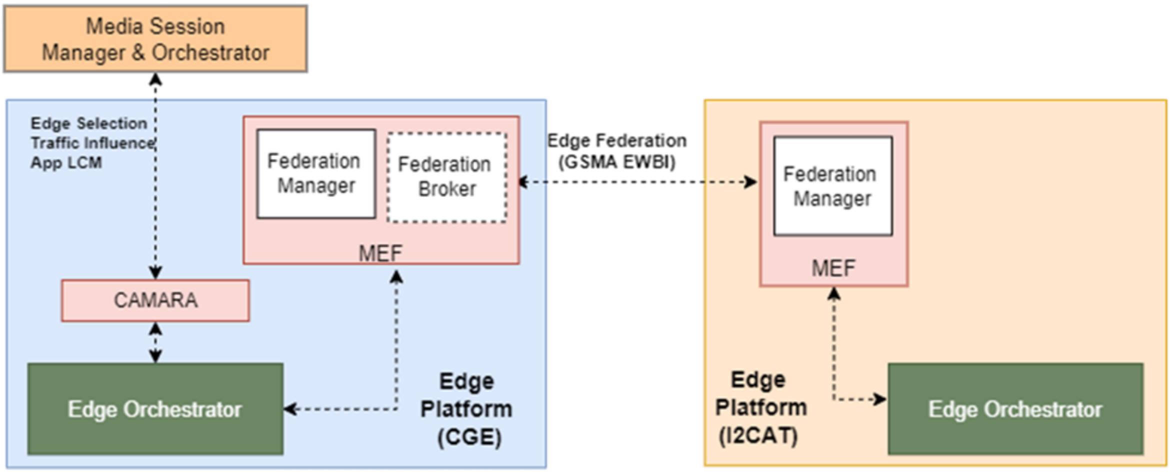

Figure 10: Federation Architecture

The depicted high-level architecture in the image outlines the structure of UC2, showcasing the two Edges. Each Edge features a box labelled MEF (MEC Federator), serving as the local agent for managing the federation. Together with the MEF, the picture shows the Edge Orchestrator, tasked with managing the resources and the life cycle of the applications locally.

The MEF interacts with other Federators through the EWBI [13], as well as with the local Edge Orchestrator. Additionally, the figure includes a box with dashed lines named Federation Broker, which is optional and likely won't be implemented for this use case.

The diagram also highlights the presence of CAMARA APIs, exclusively available on the Edge in Madrid. The vertical application will interact with the CAMARA APIs, however, and due to the federation, the vertical application gains the flexibility to deploy services by selecting the optimal edge for application instantiation. This choice is independent of whether it runs on the resources of the local operator or another operator within the federation.

Regarding functionality, the project will adhere to the guidelines outlined in the GSMA Operator Platform Group's East-Westbound Interface APIs document, Version 3.0, dated July 26th, 2023 [14]. Because not all the methods detailed in the document are essential for validating the use case, our emphasis will be on prioritizing the following functionalities and methods.

- **Federation establishment**: Federation between two Ops conceptually refers to an agreement to allow exposure of Edge Cloud resources and Network capabilities by the other OP. Note that these federation establishment procedures are not bidirectional, it is only done in one direction, if both Ops want to use each other resources, the federation establishment must be done twice, once per operator. Methods:
    - Create Federation
    - Remove Federation

    - Get Federation Meta Info
    - Update Federation
- **Availability zone info Exchange**. The OP offering resources may expose one or more Availability Zone(s) and the associated Edge to the OP. The requesting OP would be able to subscribe/unsubscribe to these zones or request information about the zone. Methods:
    - Zone Subscribe
    - Zone Unsubscribe
    - View Zone Information
- **Application Artefacts Management.** Considering that a federation has been established, an OP shall be able to onboard and manage application artefacts towards an OP partner, e.g. application images (Docker container image file(s)) and associated application component descriptors i.e., artefacts such as Helm charts, Terraform scripts etc. Note that the same artefact(s) can be reused by multiple applications of the same artefact provider. Methods:
    - Onboard Artefact
    - Remove Artefact
    - View Artefact
    - Upload File
    - Remove File
    - View File Info
- **Application Onboarding Management**. Onboarding of application profiles. Methods:
    - Onboard Application
    - Update Application
    - Remove Application
    - View Application
- **Application Deployment Management Service:** The EWBI should allow the OP to control the launch and termination of applications that have been onboarded on a partner OP. Methods:
    - Instantiate Application
    - Remove Application Instance
    - View Application Instance
    - List Application Instances

7 CAMARA AND NEF APIS FOR SLICING

This chapter refers to the APIs that will be provided by the Network Exposure Function (NEF) of the Core network provided by Ericsson in the South Node to handle slices. They will be used for 6G-XR project's UC1 *Resolution Adaptation or Quality on Demand* and UC2 *Routing to the Best Edge.*

7.1 GAP ANALYSIS

6G-XR project's UC1 *Resolution Adaptation or Quality on Demand* and UC2 *Routing to the Best Edge* are demanding from the network at the South Node to be able to dynamically adapt to the QoS demands from their application traffic flows as well as to be able to flexibly adjust the path for the user plane traffic.

To achieve the above, we need to explore the concept of network exposure, which means making network capabilities, such as data and network services, easily available for communication service providers (CSPs) and third parties (3PP). In 5G, capabilities can easily be exposed through APIs, which allows developers to create innovative services and also enables the capability to boost programmability and automated adaptability to each service needs. The automation and programmability are crucial to be able to handle the ever-growing number of IoT devices (already in the order of billions), autonomous vehicles, XR experiences and so many more. Obviously, it needs to be done in a controlled and secure way, under some conditions previously agreed with the CSPs. The NEF is the function defined for this purpose in the 5G reference architecture. Besides the NEF APIs, the CAMARA initiative by the Linux Foundation aims to provide a programmable interface for developers and other users (capabilities consumers) to request certain capabilities to Telco networks without the necessity to have an in-depth knowledge of the 4G/5G system or the overall complexity of the Telecom Systems. To this end, CAMARA requests the 5G network capabilities through the standard 3GPP networks APIs provided by NEF and exposes a Northbound REST API towards the Application Functions (AFs).

A second key lever is the use of network slicing. Network slices can be used to provide certain policies, rules, amount of network resources, etc. for certain application flows to meet their service needs, while keeping isolation from other flows. The isolation among flows is important, because the actions triggered by any AF should only be allowed to influence on their own application flows and must not impact any other flows.

At the time of starting this project, there was already an operative 5G SA core network at 5Tonic lab, provided by Ericsson. The essential network functions are deployed and UEs can get registered in the 5G network and establish data sessions. Those network functions are AMF, SMF, UPF, UDM+UDR, NRF, PCF and NSSF. For further details on how these functions work, please refer to any 5G System overview tutorial (e.g., at the 3GPP webpage [15]). The implemented Ericsson's 5G SA network allows the operator to define different slices. The gap to address is to deploy a NEF component that implements a series of APIs to expose the network slicing capabilities to 3PP AFs to meet the requirements of the use cases.

For both UC1 and UC2, it would be useful that when a user starts a holographic communication session, the AF could get the information of the cell where the user is located, so the AF can request the use of the most appropriate E2E slice, using the UPF and Data Network Name (DNN) which are closer to the user. This will be performed using the 3GPP Service Parameter and the 3GPP Monitoring Event APIs.

On UC1, the AF aims to trigger actions like resolution adaptation or quality of service changes in the network. The intention is to make the decisions for those actions based on the congestion detected in the network. To be able to achieve that, a mechanism is required that allows the AF to know in which cell the user is located, and to get subscribed to event reports from that cell to get a notification when congestion is detected. This can be obtained through the 3GPP Monitoring Event API and the 3GPP Analytics Exposure API.

On UC1 subcase 2, the holographic communication AF would request guaranteed bandwidth during a specific data communication session (AF session) of the application users located in a cell where congestion has been reported, to secure the network resources to their users. In this case the On-demand Quality of session request would be done using CAMARA QoD API which internally calls 3GPP AF session With QoS API to interact with NEF.

Summarizing, UC1 and UC2 will require mechanisms to allow the AMF to:

- Select the slice, the UPF and DNN to be used by an application flow.
- Know the application user's location.
- Modify on-demand the quality of service offered to the users.
- Determine specific network conditions at cell level under a congested area.

To address these gaps, the following XR enablers will be provided:

- 3GPP Service Parameter API.
- 3GPP Monitoring Event API.
- CAMARA QoD API / 3GPP AF session With QoS API.
- 3GPP Analytics Exposure API.

7.2 INITIAL SOLUTION DESIGN

7.2.1 Service Parameter API Functional Description and Architecture

The scope of the Service Parameter NEF Service is defined in 3GPP Technical Specification 23.502 v17.5.0 [16]. In particular, at the South Node we are interested in the procedure to allow an AF to provide guidance for UE Route Selection Policy (URSP) determination to 5G Systems via NEF described in section 4.15.6.10 of that document. This note is included in the description of the procedure: "*The operator can negotiate with an external party (typically a Corporate represented by an AF) dedicated DNN(s) and/or S-NSSAI(s) for the traffic of UE(s) of this external party*". Thus, the AF will request a policy subscription for a UE or a group of UEs to use a certain S-NSSAI and DNN for its own application data flows.

Figure 11 illustrates how Ericsson's implementation of the feature in the South Node works.

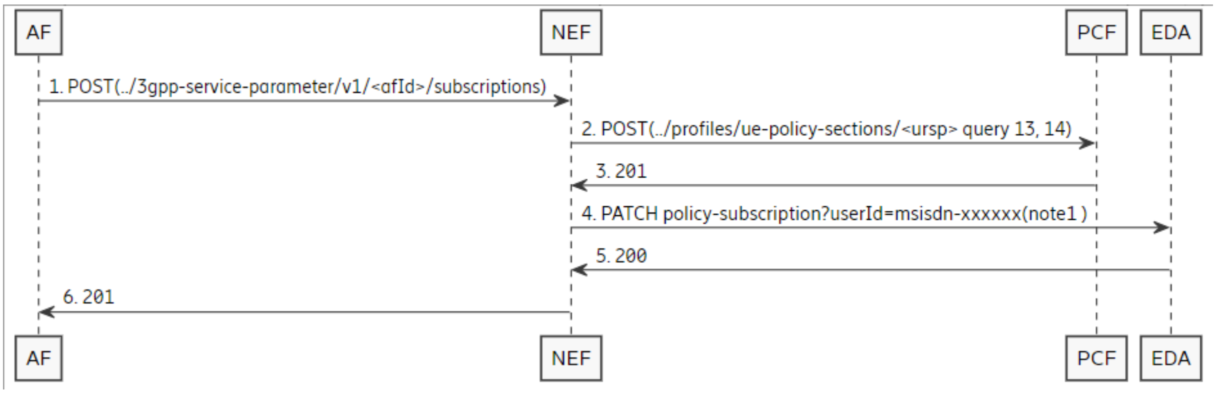

Figure 11: Service Parameter workflow

1. NEF receives a Service Parameter Provisioning request of AF Guidance to PCF Determination of URSP Rules for single UE by providing GPSI (Generic Public Subscription Identifier).
2. NEF provisions the URSP rules into PCF.
3. PCF acknowledges the provisioning with a 201 response.
4. NEF assigns the provisioned URSP rules to the specified GPSI with a PATCH request to Ericsson Dynamic Activation (EDA), which is a supporting function to the provisioning on UDM/UDR.
5. EDA responds with a 200 response.
6. NEF responds with a 201 response, acknowledging the creation of a subscription, and returns the subscription URI in the Location header of the 201 response.

To interact with this feature, the AF will use the 3GPP Service Parameter API, which can be found in 3GPP TS 29.522 repository [17]. An example of use is shown in Appendix A.

7.2.2 Monitoring Event API Functional Description and Architecture

As defined in 3GPP TS 23.502 section 4.15.3.1 [16], Event Exposure, also called Monitoring Events (MONTE), is a network function in 3GPP NEF, intended for monitoring specific events in the 3GPP system and making the information of these monitoring events available for AF via the NEF.

By means of Event Exposure, authorized AFs can get the events notification or immediate event status of the UE, for example the UE reachability, location or roaming status, which are valuable for AF to provide more services.

AF performs an Event Configuration to NEF for a single UE, group of UE or any UE (either MSISDN or UE External ID, UE External Group ID is needed to specify the UE identifier), sending the event monitoring subscription request to the NEF:

POST {apiRoot}/3gpp-monitoring-event/v1/{scsAsId}/subscriptions/
With HTTP Body: MonitoringEventSubscription (externalId/msisdn, monitoringType, maximumNumberOfReports, monitorExpireTime, notificationDestination, mtcProviderId)

Then AF can get:

- an Event Notification of configured events:

POST AFCallback/
With HTTP Body: MonitoringNotification (subscription, MonitoringEventReports (externalId/msisdn, monitoringType, eventTime, locationInfo)), and the locationInfo includes information (ageOfLocationInfo, cellId, enodeBId, routingAreaId, trackingAreaId)

- or an Immediate report in event configuration response to get the instant status of monitoring event of single UE if the UE status is available at the time of monitoring event configuration:

200 response
With HTTP Body: MonitoringEventReport

The Monitoring Event API can provide the Holographic AF with information about location of the UE (event LOCATION_REPORTING). This event is detected by the AMF, based on the Event Reporting Information Parameters that were received in the Monitoring Request such as:

- The accuracy: cell, gNodeB, TrackingArea/RoutingArea level granularity.
- The location type: Current Location and the Last Known Location.
- The report mode:
    - One-Time Report: The report is generated only once. After reporting, the subscription to this event is terminated. (For Last Known Location type, only One-time Reporting is supported, for example, AMF reports the Last Known Location immediately.)
    - Continuous Report: The report is generated when the location of UE with the accepted granularity is changed.
    - Periodic Report: The report is generated periodically.

A detailed workflow of an Event Configuration showing Southbound interactions in Figure 12:

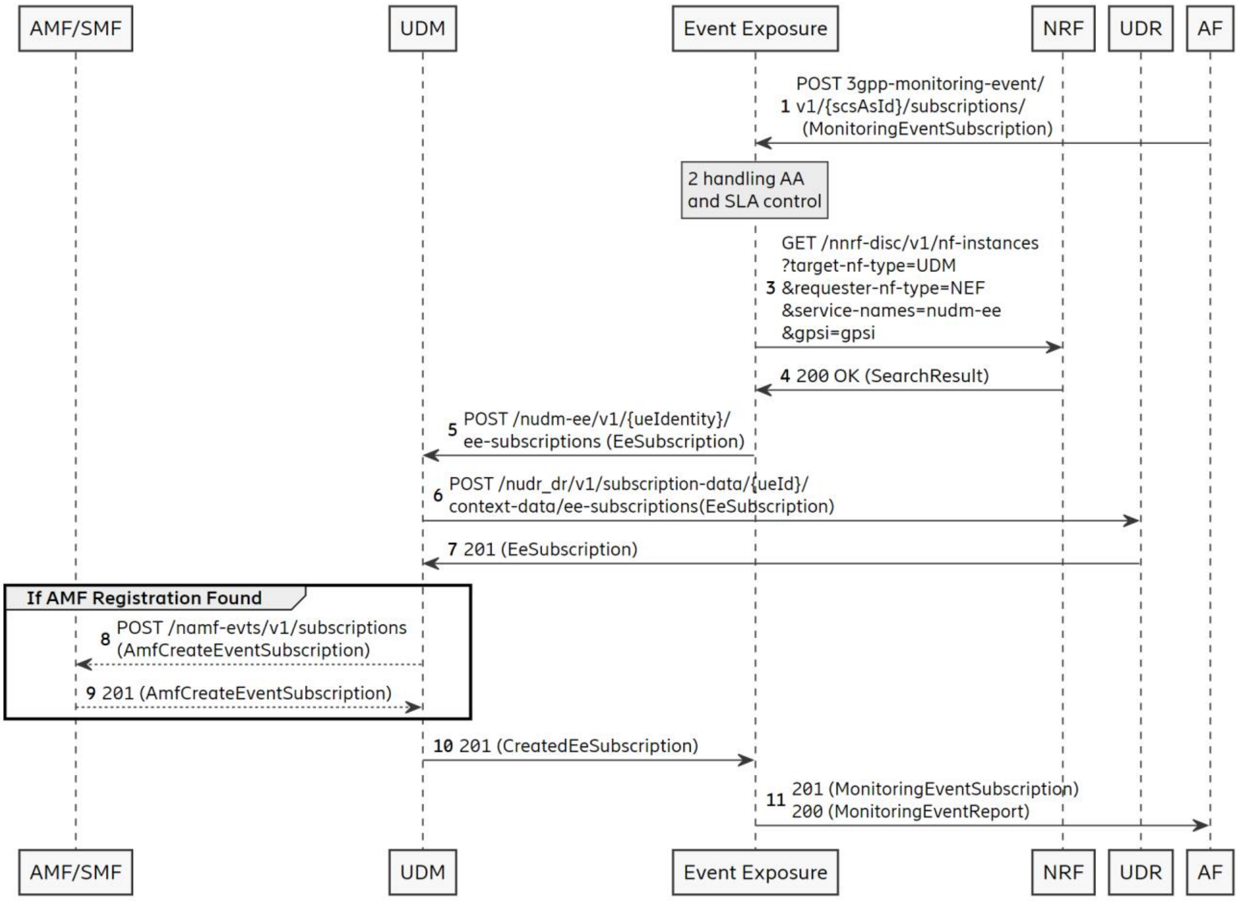

Figure 12: Event Configuration workflow

1. The AF sends an HTTP POST MonitoringEventSubscription message to the Event Exposure.
2. The Event Exposure authenticates and authorizes the AF to perform this monitoring event request and assigns a 64bit Ref-Id for this request.
3. The Event Exposure sends an HTTP GET message to NRF to retrieve the UDM information.
4. The NRF sends a response to the Event Exposure 200 SearchRequest with UDM Information.
5. The Event Exposure sends an HTTP POST message to UDM to authorize to configure monitoring events for a given UE.
6. The UDM sends an HTTP POST message to UDR to save the authorization.
7. The UDR sends a response to the UDM to acknowledge this request.
8. If the requested event (for example, Loss of Connectivity) requires AMF assistance, the UDM sends an HTTP POST message to AMF, including the notification endpoint of the Event Exposure. If the requested event (for example, PDU session status) requires SMF assistance, the UDM sends an HTTP POST message to SMF.
9. The AMF or SMF sends a response to the UDM to acknowledge this request.
10. The UDM sends a response to the Event Exposure to acknowledge this request. Immediate monitoring event report is embedded in this response if this report is available on UDM.
11. Either the Event Exposure sends a response to the AF to acknowledge the request, where the response is a 201 HTTP header location indicating the URI of the created resource, or the Event Exposure sends a 200 response if immediate monitoring event report is available on the Event Exposure.

For example, for Location Reporting event type, the MonitoringEventReport includes (externalId/msisdn, locationInfo, monitoringType), and the locationInfo includes information (ageOfLocationInfo, cellId, enodeBId, routingAreaId, trackingAreaId).

A detailed workflow of a Location Reporting Event Notification showing Southbound interactions is shown in Figure 13:

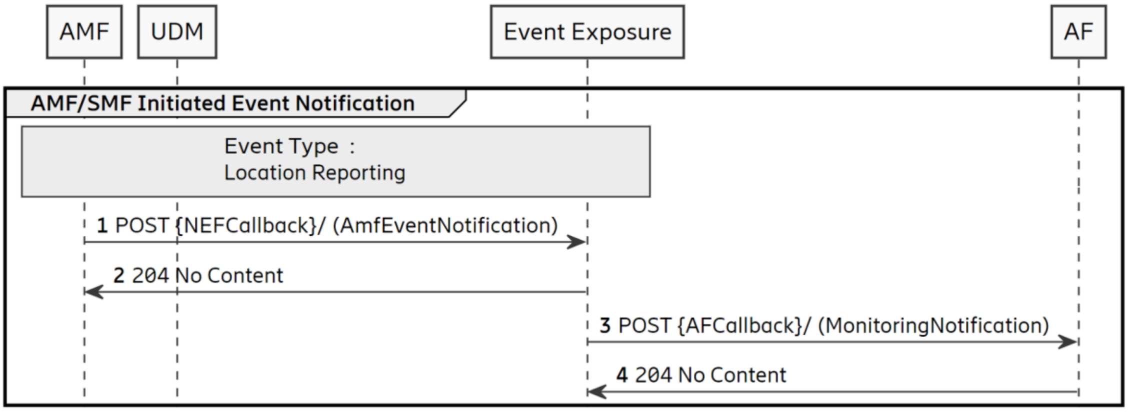

Figure 13: Event Notification workflow

1. The AMF sends an HTTP POST message to the Event Exposure callback address.
2. The Event Exposure sends a response to AMF to acknowledge the request.
3. The Event Exposure sends an HTTP POST message, specifying the type is Location Reporting, and including the information regarding the location: (ageOfLocationInfo, cellId, enodeBId, routingAreaId, trackingAreaId).
4. The AF sends a response to the Event Exposure with 204 No Content.

The detailed API is available at 3GPP TS 29.122 repository [18]. An example of use is shown in Appendix A.

7.2.3 CAMARA QOD / AF Session with QoS API Functional Description and Architecture

The CAMARA Quality on Demand (QoD) Service API offers to the AF the capability to set quality for a mobile connection, such as requesting a stable latency (reduced jitter) or throughput for some specified application data flows between application clients (within a user device) and application servers (backend services). There is a pre-defined set of Quality of Service (QoS) profiles which they could choose from depending on their latency or throughput requirements. The API is described at the CAMARA project *QualityOnDemand* repository [8].

A detailed workflow to change the QoS of an existing device connection is show in Figure 14:

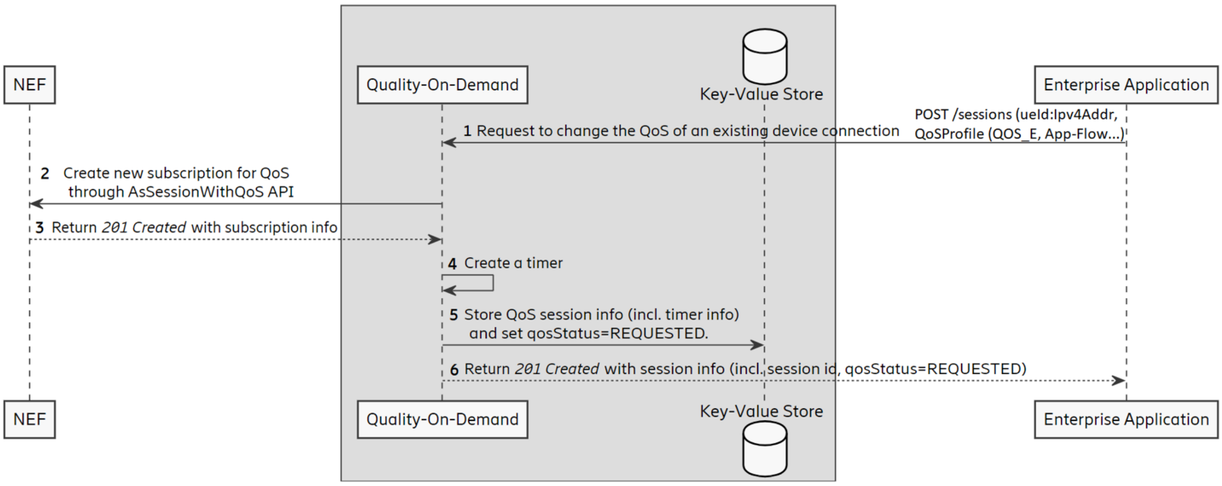

Figure 14: QoS change workflow from AF to NEF

1. Enterprise Application initiates a CAMARA QoD API request (POST /sessions) to request QoS-change of an existing device connection, specifying the desired QoSProfile (QoS_E, App-Flow, etc) and other parameters like ueId.
   The API request parameter ueId must include device IP address, as it is a required input to NEF AfSessionWithQoS API.
2. CAMARA QoD calls NEF AfSessionWithQoS API to create a new subscription for device connection QoS.
3. NEF returns HTTP response code 201 with the information of the created subscription to Quality-On-Demand.
4. CAMARA QoD creates a timer to control the QoS session duration.
5. CAMARA QoD creates a QoS session for the QoS change request and stores the session information (including the timer information) in the key-value store.
   The status of the QoS session is set to REQUESTED, which indicates that QoS change has been requested by creating a session.
6. CAMARA QoD returns HTTP response code 201 with the information of the created QoS session (including session id and the REQUESTED status) to the Enterprise Application.

As defined in 3GPP TS 23.502 section 4.15.6.6 [16], the Application Function (AF) session with required QoS service is to allow the AF to influence the QoS of one or multiple service data flows of one ongoing PDU session of a User Equipment (UE), providing policy requirements (specific QoS) to the corresponding PCF, which initiates the PDU session modification.

The requested QoS resource should be pre-provisioned in NEF with a QoS Reference before executing the creation of QoS subscription.

AF creates a new subscription for AF session with required QoS service with PCF, sending to NEF:

POST {apiRoot}/3gpp-as-session-with-qos/v1/{scsAsId}/subscriptions/
With HTTP Body: AfSessionWithQoSSubscription (UE IP address, flow information, qosReference and other attributes)

NEF responds to AF with successful creation of subscription:

201 - Created

A detailed workflow showing Southbound PCF interaction is shown in Figure 15:

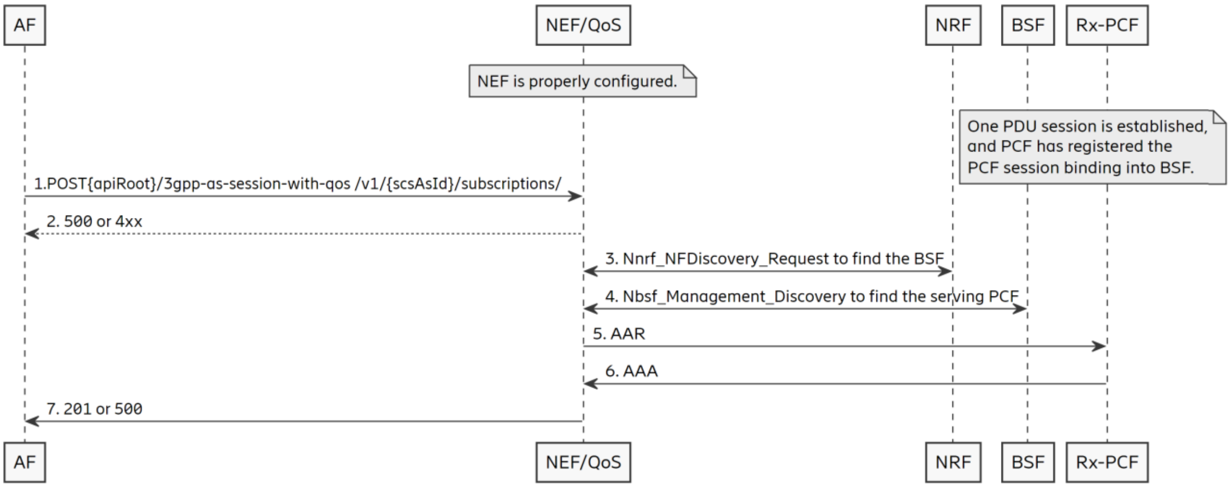

Figure 15: QoS change workflow from AF to PCF

1. NEF receives Nnef_AfSessionWithQoS_Create from the AF with UE IP address, flow information, qosReference and other attributes.
2. NEF performs internal check on the message syntax, attributes value, and so on, and might reject this request from AF.
3. NEF discovers the BSF through NRF by invoking Nnrf_NFDiscovery_Request service operation and using UE address, ipDomain and DNN (derived from qosReference) as query parameters.
4. NEF further discovers the serving Rx-PCF of concerned PDU session (identified by the UE address) via the discovered BSF by invoking Nbsf_Management_Discovery service operation and using UE address, ipDomain and DNN (derived from qosReference) as query parameters.
5. NEF maps the received qosReference to a set of QoS parameters, constructs and sends AAR to the discovered serving Rx-PCF to create the subscription.
6. NEF receives AAA from Rx-PCF.
7. If the Result-Code AVP in AA-Answer indicates “DIAMETER_SUCCESS 2001”, NEF regards that the Rx session is successfully established and responds to AF with successful response 201; otherwise, NEF regards the Rx session as failed and responds to AF with failure response 500.

The detailed API is available at 3GPP TS 29.122 repository [18]. An example of use is shown in Appendix A.

7.2.4 Analytics Exposure API Functional Description and Architecture

As defined in 3GPP TS 23.288 V17.4.0 [19], Analytics Information Exposure service allows Application Function (AF) to subscribe or retrieve analytics event/information, such as, specific area data congestion information, from a Data collection & analytics node by NEF. This node collects data from network functions and/or OAM system, analyses data and makes prediction. It replies with the analytics information to NEF per retrieval request or notifies NEF corresponding analytics events if subscribed.

AF subscribes Analytics Information for a Specific Network Area, sending the Analytics Exposure Subscription Request to the NEF:

POST {apiRoot}/3gpp-analyticsexposure/v1/{scsAsId}/subscriptions/
With HTTP Body: AnalyticsExposureSubsc (notifId, notifUri, analyRepInfo (immRep, notifMethod), analyEventsSubs (analyEvent="CONGESTION", analyEventFilter (locArea, reptThlds: congLevel), tgtUe: "anyUeInd"=true))

Then AF can receive:

- an Analytics Event Notification of subscribed events:

POST AFCallback/
With HTTP Body: AnalyticsEventNotification (notifId, analyEventNotifs (analyEvent="CONGESTION", congestInfos)), and the congestInfos includes information (locArea, cngAnas (cngType, tmWdw, nsi: congLevel))

- or an Immediate report included in Analytics Exposure Subscription Response to get the instant Analytics information from this specified network area:

201
With HTTP Body: AnalyticsExposureSubsc (eventNotifis (notifId, analyEventNotifs (analyEvent="CONGESTION", congestInfos)))

The Analytics Exposure API can provide the Holographic AF with congestion analytics information related to a specific area (analytics event CONGESTION). This event is detected by the Data collection & analytics node, based on the Analytics Information Parameters that were received in the Subscription Request such as:

- Immediate report (immRep=true): Report analytics information at Analytics Exposure Subscription Response.
- Notify method:
    - One-Time Report (notifMethod="ONE_TIME"): Report analytics information only once. After reporting, the subscription to this event is terminated.
    - On-event Detection Report (notifMethod="ON_EVENT_DETECTION"): Report analytics information when the subscribed event detected.
    - Periodic Report (notifMethod="PERIODIC"): Report analytics information periodically with reporting interval (repPeriod) specified in subscription.

A detailed workflow of an Analytics Exposure Subscription showing Southbound interactions is shown in Figure 16:

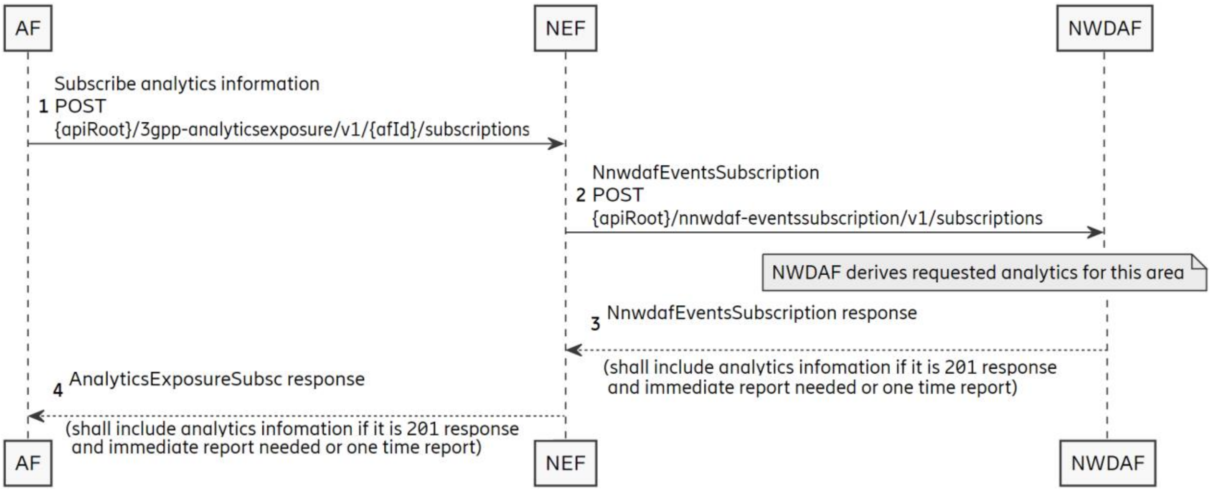

Figure 16: Analytics Exposure Subscription workflow

1. NEF receives a Nnef_AnalyticsExposure create request for analytics information subscription by providing a network area where requested analytics locate and "anyUeInd" set to true in tgtUe.
2. NEF sends Nnwdaf_AnalyticsSubscription create request for analytics subscription to Data collection & analytics node.
3. Data collection & analytics node responds to NEF for this request.
4. NEF maps the response from Data collection & analytics node to AF:
5. NEF responds 201 if subscribe is successful.
6. Analytics information from this specified network area is included in response if subscribed for immediate report or with notifMethod="ONE_TIME".
7. NEF responds error status code if subscribe is unsuccessful.

A detailed workflow of Analytics Information Notification showing Southbound interactions is shown in Figure 17:

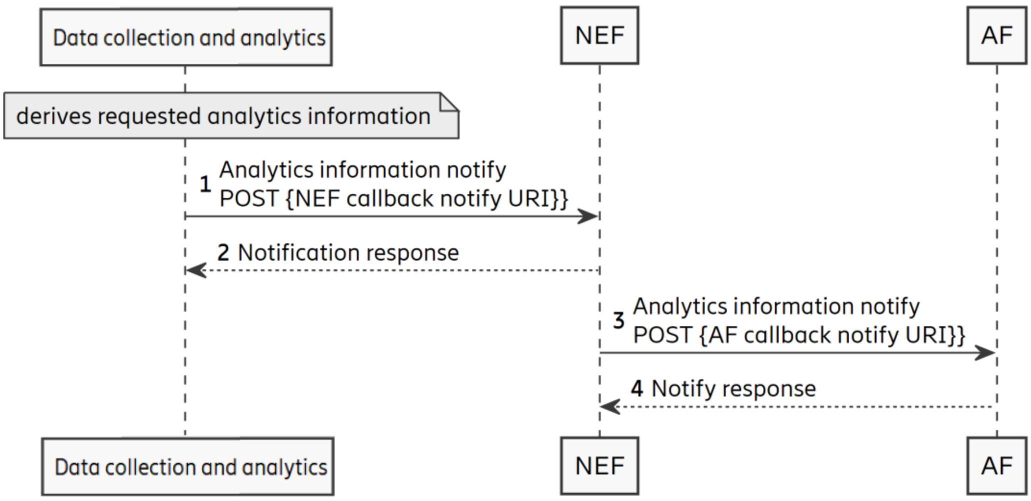

Figure 17: Analytics Information Notification workflow

1. Data collection & analytics node derives requested analytics, sends analytics information to NEF by HTTP POST.

2. NEF sends 204 No Content or error status code response to Data collection & analytics node when error happens.
3. NEF maps analytics information to AF.
4. AF sends the response to NEF.

The detailed API is available at 3GPP TS 29.522 repository [17]. An example of use is shown in Appendix A.

8 5GTN SLICE CREATION

This chapter includes the gap analysis of slice creation in the University of Oulu 5G Test Network (5GTN). First, the current state of Slicing in the 5GTN is described, then the 6G-XR target state for slice creation explained and finally the implementation how the target state is to be achieved is presented.

8.1 GAP ANALYSIS

Current State:

5GTN has several different 5G networks in use. Main ones being Non-Stand Alone (NSA) macro based outdoor network, SA based outdoor network and SA based indoor network. One of the base requirements of the 6G-XR project is 5G SA network with E2E slicing support. Main 5GTN Stand Alone 5G network is currently built using Open5GS Core and a commercial 5G Base Station. However, current set-up, as described earlier in this document, does not support dynamic slicing including RAN slicing. Addition to commercial Base Transceiver Station (BTS), another Open5GS is connected to OAI-RAN to produce second SA 5G network. This setup does not support dynamic slicing either. Both 5G SA networks use their own Public Land Mobile Network (PLMN) and radio frequency. The main SA network uses a macro BTS and covers outdoor area in the University of Oulu campus area. The OAI-RAN version instead is using USRP radio for indoors.

Target State:

5GTN will support 5G SA network with dynamic slicing. A 5G core with capability to dynamically create 5G slices is needed. Core should have an API that can be utilized to dynamically manage 5G slices. Core needs to be integrated with 5G RAN. RAN slicing is to be supported. Core is to be integrated together with the North Node Adapter API, that will orchestrate the slice life cycle.

For the 5G slice life cycle management through the North Node Adapter API the 6G-XR project requires following functionality from the underlying 5GTN Facility:

- Slice creation
- Listing of active slices
- Slice re-configuration
- Slice deletion

As the 5GTN does not support the dynamic slicing, the above-mentioned target state can be seen as the identified gap.

8.2 INITIAL SOLUTION DESIGN

5GTN will take Cumucore 5G Core in use during 1Q/2024 parallel to other supported Core solutions. Cumucore 5G Core is to be used by 6G-XR project as it supports dynamic slice creation and configuration that is essential for the 6G-XR. Below is description of the Cumucore 5G Core and the API it has that supports also dynamic slice management.

Cumucore 5G NC [20] is a 3GPP compliant Packet Core that supports 5G SA functionality. It is designed using microservices for the functions that are defined in the 5G system. It can be deployed in bare metal or virtualized environments using for example OpenStack, OpenShift or Kubernetes.

Cumucore 5G NC includes all the functionality that is required for interoperability with 3GPP Rel 16 and it has been tested to work with different vendor RANs. It includes the Service Based Architecture (SBA) network functions to support network slicing: NSSF and discovery of Multi-Access Edge Computing (MEC) through the Network Repository Function (NRF).

Cumucore has a Network Slicing Manager integrated with a Cumucore Network Wizard and it includes a Graphical User Interface (GUI) that can be used to manually manage the network. Using the Network Slice Manager the user can define slice sizes, quality parameters and traffic rules for each slice including priorities and pre-emption rules.

The Core includes an API that the Network Slice Manager utilizes for the functions it offers. As the Network Slice Manager GUI requires manual use, it is as such suitable for 6G-XR as it requires that slicing orchestration through SW as the North Node Adapter API is to handle the slice life cycle. Therefore, the intention is that 6G-XR North Node Adapter API would directly use the API for the dynamic slice management.

Following is a brief definition of the Cumucore API that is fully described in the CNC v5.5 Test RST API Cumucore Network Configuration document [21] that contains the whole API description.

Cumucore slice life cycle management is divided into three phases [22]:

- Preparation Phase for Network preparation
    - Design
    - On boarding
- Commissioning Phase
    - Creation of a slice
- Operation Phase
    - Activate
    - Supervise, Modify and Report
    - Deactivate
- De-Commissioning Phase
    - Terminate

As can be seen from above Cumucore fulfils the 6G-XR slicing life cycle management requirements.

Table 9 provides the definition of the CNC-1: Slice instance from the CNC v5.5 Test RST API Cumucore Network Configuration [21] where the basic slice life cycle API functions are described:

Table 9: Slice instance API function

Method	Path	Description
GET	{apiRoot}/api/v1.0/network-slice/slice-instance	Getting a list of slice instances configured in the CNC. It shows both active and non-active slices example: curl -k -i -X GET https://192.168.9.114:3000/api/v1.0/network-slice/slice-instance

POST	{apiRoot}/api/v1.0/network-slice/slice-instance	Adding slice instance metadata to the CNC based on the RAN information for the list of supported NSSAI and PLMN list and other related data. Example: curl -k -i -X POST https://localhost:3000/api/v1.0/network-slice/slice-instance -d '{"sliceName":"slice1","activate_slice":1,"ServiceProfile":{"SNSSAIList":[{"sst":1,"sd":"000002"}],"PLMNIdList":[{"mcc":"999","mnc":"99"}],"maxNumberofUEs":300,"latency":0,"DLThptPerSlice":{"guaThpt":500,"maxThpt":1000},"DLThptPerUE":{"guaThpt":0,"maxThpt":0},"ULThptPerSlice":{"guaThpt":500,"maxThpt":1000},"ULThptPerUE":{"guaThpt":0,"maxThpt":0},"maxDLDataVolume":"1000 Mbps","maxULDataVolume":"1000 Mbps","survivalTime":"","dnn":"internet"},"NetworkSliceSubnet":{"SliceProfile":{"SNSSAIList":null,"PLMNIdList":null,"PerfReq":{"PerfReqEmbbList":{"expDataRateDL":21,"expDataRateUL":21,"areaTrafficCapDL":1,"areaTrafficCapUL":1,"userDensity":0,"activityFactor":20,"ExpDataRateDL":0,"ExpDataRateUL":0,"AreaTrafficCapDL":0,"AreaTrafficCapUL":0,"UserDensity":10},"PerfReqUrllcList":{"cSAvailabilityTarget":0,"cSReliabilityMeanTime":0,"expDataRateDL":0,"msgSizeByte":"","CSAvailabilityTarget":10,"CSReliabilityMeanTime":10,"ExpDataRateDL":10,"MsgSizeByte":""}},"CNSliceSubnetProfile": ok Method Path Description Status {"maxNumberofUEs":0,"latency":0,"DLThptPerSliceSubnet":{"guaThpt":0,"maxThpt":0},"DLThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerSlicePerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"maxNumberofPDUSessions":0,"CoverageAreaTAList":{"TAC":[16512]}},"RANSliceSubnetProfile":{"CoverageAreaTAList":{"TAC":[16512]},"UEMobilityLevel":"","resourceSharingLevel":0,"maxNumberofUEs":0,"activityFactor":0,"DLThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"uESpeed":0},"TopSliceSubnetProfile":""},"managedFunction":"","EpTransport":{"ioAddress":"","logicInterfaceId":"","qosProfile":9,"epApplication":["internet"]},"SNSSAIList":[{"sst":1,"sd":""}],"PLMNIdList":[],"OperationalState":true,"AdministrativeState":"","nsInfo":"","ManagedFunction":""},"SliceDescription":""}' Expected response: success: status code: 200 Failure: status code: 404

PUT	{apiRoot} /api/v1.0/network-slice/slice-instance/ {:slice_name}	Updating/modify the slice instance metadata. Expected response: success: status code: 200 Failure: status code 404 Example: curl -k -i -X POST https://localhost:3000/api/v1.0/network-slice/slice-instance -d '{"sliceName":"slice1","activate_slice":1,"ServiceProfile":{"SNSSAIList":[{"sst":1,"sd":"000002"}],"PLMNIdList":[{"mcc":"999","mnc":"99"}],"maxNumberofUEs":300,"latency":0,"DLThptPerSlice":{"guaThpt":500,"maxThpt":1000},"DLThptPerUE":{"guaThpt":0,"maxThpt":0},"ULThptPerSlic OK e":{"guaThpt":500,"maxThpt":1000},"ULThptPerUE":{"guaThpt":0,"maxThpt":0},"maxDLDataVolume":"1000 Mbps","maxULDataVolume":"1000 Mbps","survivalTime":"","dnn":"internet"},"NetworkSliceSubnet":{"SliceProfile":{"SNSSAIList":null,"PLMNIdList":null,"PerfReq":{"PerfReqEmbbList":{"expDataRateDL":21,"expDataRateUL":21,"areaTrafficCapDL":1,"areaTrafficCapUL":1,"userDensity":0,"activityFactor":20,"ExpDataRateDL":0,"ExpDataRateUL":0,"AreaTrafficCapDL":0,"AreaTrafficCapUL":0,"UserDensity":10},"PerfReqUrllcList":{"cSAvailabilityTarget":0,"cSReliabilityMeanTime":0,"expDataRateDL":0,"msgSizeByte":"","CSAvailabilityTarget":10,"CSReliabilityMeanTime":10,"ExpDataRateDL":10,"MsgSizeByte":""}},"CNSliceSubnetProfile":{"maxNumberofUEs":0,"latency":0,"DLThptPerSliceSubnet":{"guaThpt":0,"maxThpt":0},"DLThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerSlicePerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"maxNumberofPDUSessions":0,"CoverageAreaTAList":{"TAC":[16512]}},"RANSliceSubnetProfile":{"CoverageAreaTAList":{"TAC":[16512]},"UEMobilityLevel":"","resourceSharingLevel":0,"maxNumberofUEs":0,"activityFactor":0,"DLThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"uESpeed":0},"TopSliceSubnetProfile":""},"managedFunction":"","EpTransport":{"ioAddress":"","logicInterfaceId":"","qosProfile":9,"epApplication":["internet"]},"SNSSAIList":[{"sst":1,"sd":""}],"PLMNIdList":[],"OperationalState":true,"AdministrativeState":"","nsInfo":"","ManagedFunction":""},"SliceDescription":""}'
DELETE	{apiRoot} /api/v1.0/network-slice/slice-instance /{:slice_name}	Example: curl -k -i -X DELETE https://192.168.9.1:3000 /api/v1.0/network-slice/slice-instance /slice1

PUT (ACTIVATESlice)	{apiRoot} /api/v1.0/network-slice/slice-instance/ {:slice_name}	Updating/modify the slice instance metadata Example: curl -k -i -X PUT https://192.168.9.180:3000/api/v1.0/network-slice/slice-instance/slice1 -d '{"sliceName":"slice1","activate_slice":1,"ServiceProfile":{"SNSSAIList":[{"sst":1}],"PLMNIdList":[{"mcc":"244","mnc":"59"}],"maxNumberofUEs":300,"latency":0,"DLThptPerSlice":{"guaThpt":500,"maxThpt":1000},"DLThptPerUE":{"guaThpt":0,"maxThpt":0},"ULThptPerSlice":{"guaThpt":500,"maxThpt":1000},"ULThptPerUE":{"guaThpt":0,"maxThpt":0},"maxDLDataVolume":"1000 Mbps","maxULDataVolume":"1000 Mbps","survivalTime":"","dnn":"internet"},"NetworkSliceSubnet":{"SliceProfile":{"SNSSAIList":null,"PLMNIdList":null,"PerfReq":{"PerfReqEmbbList":{"expDataRateDL":21,"expDataRateUL":21,"areaTrafficCapDL":1,"areaTrafficCapUL":1,"userDensity":0,"activityFactor":20,"ExpDataRateDL":0,"ExpDataRateUL":0,"AreaTrafficCapDL":0,"AreaTrafficCapUL":0,"UserDensity":10},"PerfReqUrllcList":{"cSAvailabilityTarget":0,"cSReliabilityMeanTime":0,"expDataRateDL":0,"msgSizeByte":"","CSAvailabilityTarget":10,"CSReliabilityMeanTime":10,"ExpDataRateDL":10,"MsgSizeByte":""}},"CNSliceSubnetProfile":{"maxNumberofUEs":0,"latency":0,"DLThptPerSliceSubnet":{"guaThpt":0,"maxThpt":0},"DLThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerSlicePerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"maxNumberofPDUSessions":0,"CoverageAreaTAList":{"TAC":[16512]}},"RANSliceSubnetProfile":{"CoverageAreaTAList":{"TAC":[16512]},"UEMobilityLevel":"","resourceSharingLevel":0,"maxNumberofUEs":0,"activityFactor":0,"DLThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"ULThptPerUEPerSubnet":{"guaThpt":0,"maxThpt":0},"uESpeed":0}," OK TopSliceSubnetProfile":""},"managedFunction":"","EpTransport":{"ioAddress":"","logicInterfaceId":"","qosProfile":9,"epApplication":["internet"]},"SNSSAIList":[{"sst":1,"sd":""}],"PLMNIdList":[],"OperationalState":true,"AdministrativeState":"","nsInfo":"","ManagedFunction":""},"SliceDescription":""}'
NOTE: During activation of the slice check the plmnid, SNSSAIList that the gNB actually supports, qosProfile value, and the epApplication for DNN lists		

It is not practical to insert rest of the API definition into this document as it is relatively large. More of the Cumucore API (like CNC-8: gNB Slice Configuration or how to add user groups to the slices) can be found from CNC v5.5 Test REST API Cumucore Network Configuration document.

9 OPEN-SOURCE 5G SOLUTIONS: OAIBOX

As mentioned in 6G-XR's deliverable D1.1 [1], within this project, alongside commercial 5G solutions, multiple open-source 3GPP compliant 5G-Advanced components will be integrated into both the North and South Nodes. The objective is to augment the capabilities of primary network deployments within testing facilities and offer increased flexibility for the implementation of project use cases and scenarios. From that point of view, in the North Node an open-source solution called OAIBOX [23], which includes both the 5G Radio Access Network (RAN) and Core network, has been considered to work as an enabler for 6G-XR's UC5 *Energy Measurement Framework for Energy Sustainability*.

9.1 GAP ANALYSIS

Current State:

The 5GTN (6G-XR North Node) currently lacks an open-source environment for experimenting and validating UC5. As mentioned in section 4.1.3, the energy measurement framework will utilize the OAIBOX solution to validate UC5's KPIs/KVIs described in D1.1 [1].

Target State:

The North Node will be equipped with an open-source environment specifically designed for experimenting and validating UC5. By integrating with the energy measurement framework from section 4.1.3, OAIBOX provides a comprehensive platform for UC5 exploration within this environment.

- **RAN configuration control:** OAIBOX will enable non-dynamic adjustments to RAN configurations while requiring base station restarts.
- **Energy measurement and analysis:** To explore energy-saving potential, we will evaluate various configurations on an OAIBOX MAX device equipped with USRP N310 and measure its power consumption.
- **Real-time network and radio parameter monitoring:** OAIBOX will provide real-time insights into both network parameters (e.g., DL/UL throughput) and radio parameters (SINR, RSSI, etc.) under different configurations. By analysing these results, we will gain insights into the trade-offs between energy efficiency and network performance under different configuration changes.

Gap Analysis Summary:

Table 10: OAIBOX enabler Gap Analysis Summary

Requirements	Current State	Target State	Gap
OAIBOX integration	Missing open-source solution for energy measurement framework to validate UC5's KPIs/KVIs.	Integration of OAIBOX within the energy measurement framework to provide a comprehensive platform for UC5.	The integration of OAIBOX within the open-source environment is necessary to facilitate comprehensive UC5 exploration and validation, aligning with project objectives.

RAN configuration control	Limited capability for RAN configuration control	OAIBOX enables non-dynamic adjustments to RAN configurations.	Enhancing RAN configuration control through OAIBOX integration allows for more efficient testing and validation processes, but the requirement for base station restarts may introduce operational challenges or delays.
Energy saving measures	Absence of energy saving measures using OAI environment.	Evaluation of various configurations on an OAIBOX MAX device equipped with USRP N310 to assess energy consumption and potential savings.	The current setup lacks energy monitoring inside OAIBOX. Integrating OAIBOX with energy measurement framework will enable investigation of potential energy-saving methods.
Real-time monitoring of network KPIs	Lack of real-time insights into network and radio parameters under different configurations.	OAIBOX provides real-time insights into network and radio parameters, facilitating analysis of trade-offs between energy efficiency and network performance.	Real-time monitoring capabilities offered by OAIBOX, and energy measurement framework are essential for understanding the impact of configuration changes on network and radio parameters, aiding in informed decision-making for optimizing performance and energy efficiency.

9.2 INITIAL SOLUTION DESIGN

In this section, a short introduction about the OAIBOX solution, key features of it, and how this is going to be integrated in the UOULU 5GTN RI so that it works as an enabler for the above-mentioned use case, is described.

9.2.1 OAIBOX Enabler Functional Description and Architecture

OAIBOX is a cutting-edge 5G network testing solution for easy deployment and validation of various network functions and devices provided by Allbesmart pte [23]. It is a combination of Open Air Interface 5G gNB (OAI-gNB) and OAI 5G Core Network (OAI-CN5G) solution.

OAI-gNB:

Open Air Interface (OAI) gNB is an open-source implementation of the 5G New Radio (NR) protocol stack, which is the core technology for Fifth Generation (5G) cellular networks. It is a software-based solution that can be deployed on commodity hardware, making it a cost-effective and flexible option for a wide range of network configurations. Check its implementation at [24].

OAI-CN5G:

OAI-CN5G is a 3GPP-compliant 5G Standalone (SA) Core Network (CN) implementation with a rich set of features. It is an open-source project that aims to provide a freely accessible and modifiable platform for building and testing 5G networks. OAI-CN5G is designed to be modular and flexible, allowing for easy customization and integration with various hardware platforms and software components. It supports a wide range of use cases, including private networks, testbeds, and commercial deployments. Check its implementation at [24].

OAIBOX Key Features:

As mentioned above, OAIBOX has both the 5G-CN and 5G-RAN solutions in a box using the OAI open-source codes. As per the provider's information, there are multiple models of OAIBOX, but the specific one present in the 5GTN lab is referred to as OAIBOX-MAX. The key features of the OAIBOX-MAX [23] are as follows:

- Integrated with both the 5G core network and FR1 RAN.
- Suitable for deployment in both indoor and outdoor environments.
- Demonstrates interoperability with leading 5G modems and User Equipments (UEs).
- It has an interactive dashboard for the control and management of both the RAN and CN.
- Enables real-time visualization of Key Performance Indicators (KPIs) for network performance.
- Compatible with widely used Software Defined Radio (SDR) cards, including Ettus B210/N300/N310/X310/X410.
- Supports configurations up to 100 MHz BW with 2x2 MIMO using the external SDR.
- Allows for the storage of network data for subsequent offline analysis.

Following list describes the OAIBOX integration into 5GTN infrastructure to facilitate the 6G-XR's UC5:

- While OAIBOX-Max itself does not come with a radio device, it does provide support for such devices, as mentioned earlier. Within the UOULU 5GTN research infrastructure, the OAIBOX-Max is set to integrate seamlessly with the Ettus USRP N310 model.
- The setup will utilize NR band n77 with a bandwidth (BW) of 100 MHz (3900-4000 MHz).
- Various types of uEs, including Quectel 5G modems, iPhones, and Android phones, will be employed for testing the network performance.
- The comprehensive solution is designed to act as an enabler for 6G-XR's UC5, the implementation of an energy measurement framework in the North Node.
- UOULU will conduct energy-related measurements using the OAIBOX, as described in Section 4.2.3. The experiment will involve external meters (e.g., power analysers), an MQTT broker running on the server for a common interface within the North Node, and experimentation with energy forecasting and network parameters. Also, real-time energy consumption will be monitored using the energy measurement framework.
- These external meters will measure the power consumption of specific components within the OAIBOX (e.g., main unit, USRP) separately, allowing adjustments to network parameters (such as bandwidth, TDD slots, modulation coding scheme, and MIMO mode selection) based on real-time data.

The following Figure 18 and Figure 19 represent the OAIBOX based high-level architecture and setup in the UOULU 5GTN infrastructure.

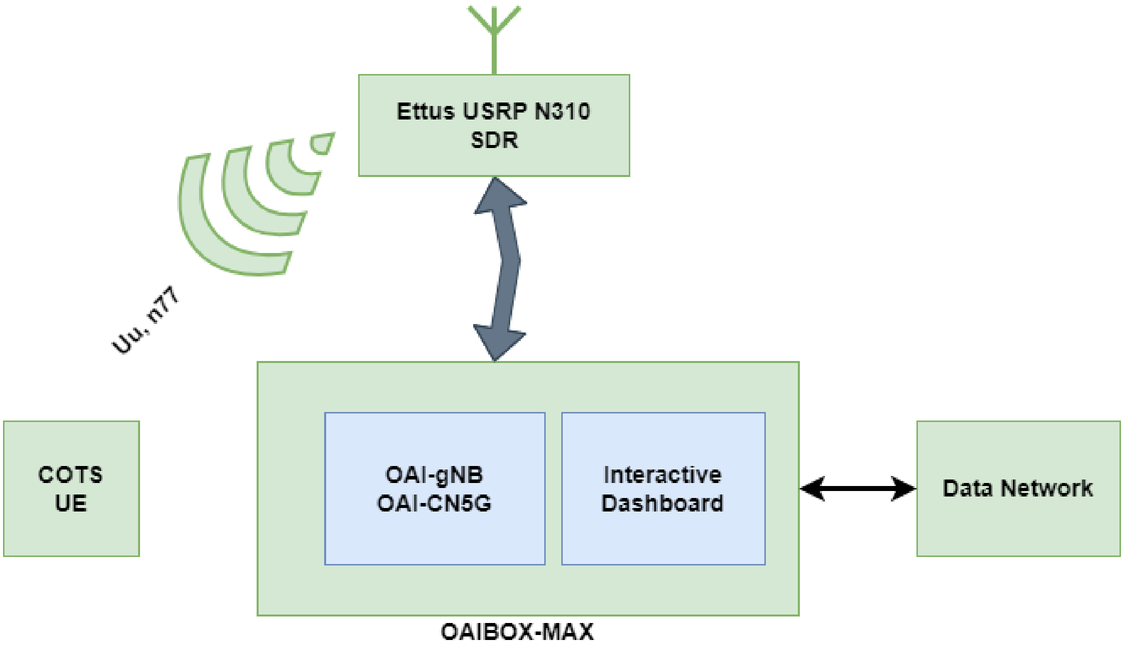

Figure 18: OAIBOX-MAX based high level architecture at the UOULU 5GTN infrastructure.

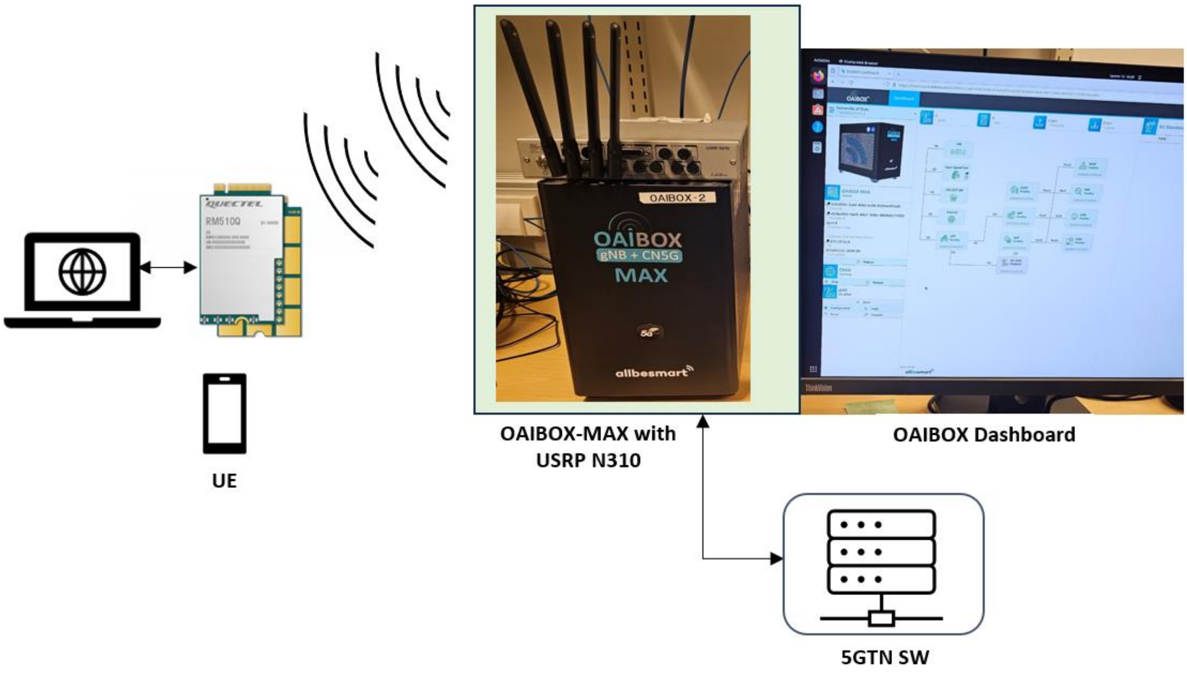

Figure 19: OAIBOX-MAX based solution setup at the UOULU 5GTN lab

10 CONTROL PLANE INNOVATIONS

This section deals with the evolution of the mobile network control plane communications. Specifically, it covers the analysis and initial solution design for deploying the IMS Data Channel (IMSDC) functionality in the South Node. It is required for validating UC3 *Control Plane Optimizations*.

10.1 GAP ANALYSIS

UC3 will focus on the creation of a network architecture capable of supporting interactive holographic calls, allowing their integration quickly and efficiently in third parties and covering the actual devices and different use cases of the future XR market.

The expected bandwidth compared to a traditional call is more than double per user. Therefore, benefiting from 5G and Edge technologies is essential, as it will provide the system with better quality and latency improvements in signalling and rendering processing.

This architecture will be based on IMS Data Channel (IMSDC) functionality, leveraging a standard IMS core network, and it is fully compatible with the general architecture of the 6G-XR project, taking advantage of the capabilities of the existing shared elements and introducing new changes in network elements that will be described below.

The development of the Data Channel Server (DCS) enabler will enhance the ability of IMS to handle different interactions between consumers and the world:

- With data channel functionality, IMS can now enhance the interactivity with the whole info sphere which is populated with informational entities (forms, web pages, games, devices, services).
- IMS Data Channel allows the integration of internet connectivity and IMS connectivity.
- Provides action-oriented and multi-modal communications.
- Blends the communications between subjects with the object control. IMS enhanced with the data channel represents a wider concept, which is the internet itself, providing a real time, reliable, secure, and QoS assured platform supporting web programming model in the industry.

IMSDC standardization is not yet fully completed:

- 3GPP Release 18 defines IMS Data Channel, and is evolving in 2024 to define a complete IMS architecture to expose control and media for IMS voice video and data channels to applications. (See Figure 20) Interfaces from IMS Functions to the DC Application server are key, especially in B2B scenarios with Business Voice Architectures (MDC2 and DC4 interfaces).
- GSMA Official Document NG.134 [25] has defined the UNI interface.
- 3GPP Release 19 will define new interfaces via NEF for IMSDC for Apps.
- GSMA IMSDCAS work group is still defining the device API.
- GSMA Permanent Reference Documents (PRD) are key to wider industry acceptance and implementation.

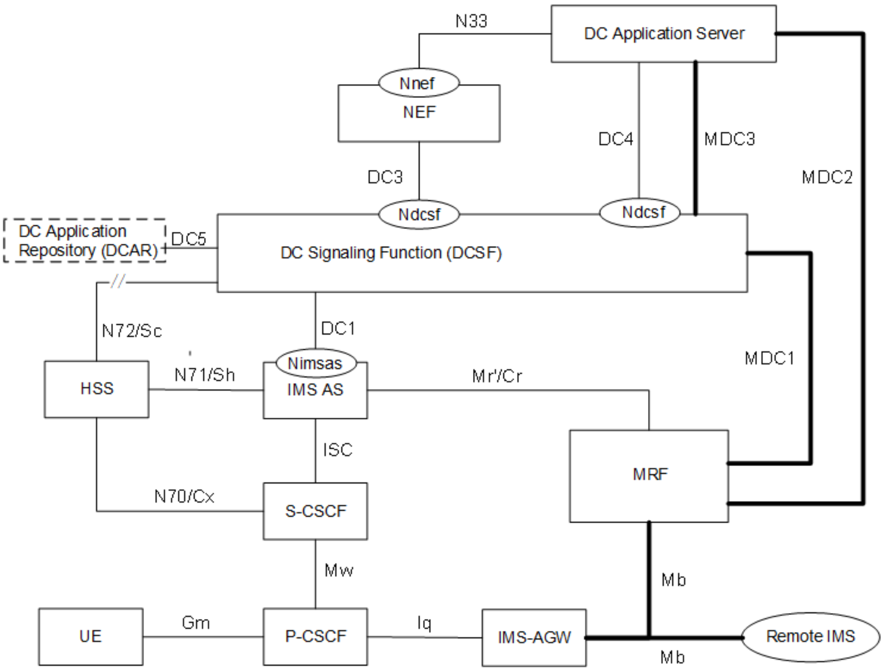

Figure 20: Architecture option of IMS supporting DC usage with MRF

An identified gap is related with NAT Traverse mechanisms widely used in Internet scenarios may not work well in Telco space:

- **ICE/STUN** is deeply integrated in WebRTC internet-oriented architecture.
- **WebRTC libraries** currently available implement it, but **not all Telco IMS infrastructure supports it.**
- In this scenario, potential hybrid scenarios with additional content downloaded from Internet servers can lead to issues.

10.2 INITIAL SOLUTION DESIGN

In order to achieve the objective of a holographic call between two users in UC3, it has been decided to carry out a series of initial cases that ensure proper functioning and correct efficiency.

The first proposed architecture (See Figure 21) identifies the different elements of the network that must be considered when establishing a holographic call between two users in presentation mode, where one agent user will produce a hologram and a viewer user visualizes it during an IMS Data Channel call with an IMSDC-capable, browsified phone's dialer.

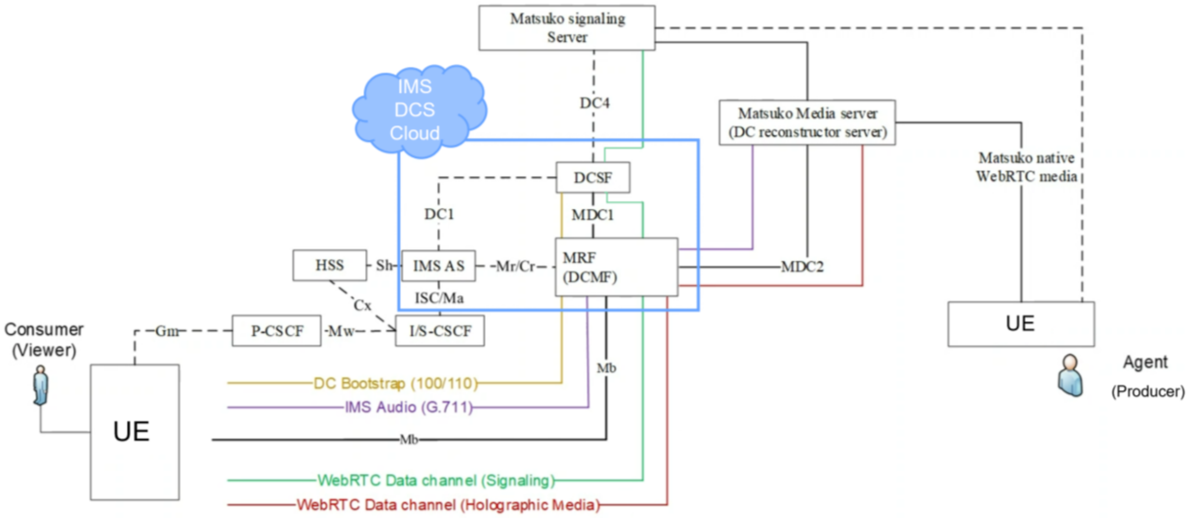

Figure 21: Current IMSDC Holographic Call architecture

The idea of this architecture is to support the separation of the signalling function and the media function that support data channel services. For this reason, a network function is specified for such purpose, the DCS.

10.2.1 Data Channel Server (DCS) Functional Description and Architecture

The elements that will be considered in the DCS are the Media Resource Function (MRF) and the Data Channel Signalling Function (DCSF).

The network function to handle Data Channel media can be provided in different ways but the selected approach has been to enhance the existing (DCSF) and the Media Resource Function (MRF) to perform DC-related media functions, which interacts with the IMS Application Server (AS) through improved Mr'/Cr interfaces.

The DCSF is the signalling control function that provides the data channel control logic. It supports functionalities such as:

- Receiving event reports from the IMS AS deciding whether to allow the data channel service to be provided during the IMS session.
- Managing the boot data channel application data channel resources.
- The capacity of the HTTP server to download applications from the data channel.

The MRF provides media resource management and forwarding of data channel media traffic. It supports:

- Terminating the boot data channel from the UE and forward HTTP traffic between the UE and DSCF through MDC1.
- Anchoring the application data channel in P2P scenarios, if necessary, and forward application data traffic to/from UEs.
- Relaying traffic on the A2P/P2A application data channels between the UE and the DC application server through MDC2.

To guarantee the proper functioning of the enabler, another element that should be considered as a complement is the existing Application Server (AS). The IMS AS has been enhanced to support the following functionalities:

- The IMS AS interacts with the DCSF through DC1 for event notifications.
- The IMS AS receives data channel control instructions from DCSF and consequently interacts with MRF through Mr'/Cr for data channel media resource management.

The detailed connectivity between the mentioned elements is detailed in Figure 22.

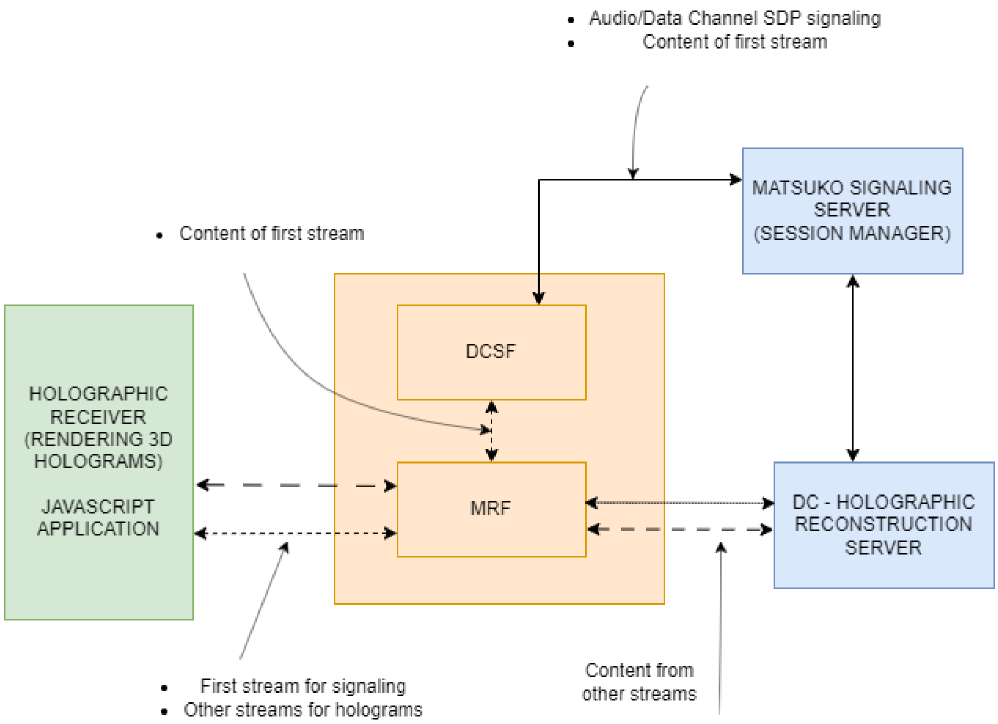

Figure 22: Current DCS Holographic call architecture connectivity

Figure 23shows a high-level diagram of the interaction between the different elements that compose UC3 is shown.

The user who generates the hologram (Agent sender) will first interact with MATSUKO signalling server for registration. When a user wants to start a call (Viewer Device), a classic call to the IMSDC service will be initiated to access the holographic services that will establish a connection between both users to start the holographic presentation.

For this to happen, once the signalling server receives the request from the viewer user, it will take care of creating the environment (connecting with the DCSF via DC4 interface) and assigning the most appropriate holographic reconstructor server (connecting with the MRF via MDC2 interface).

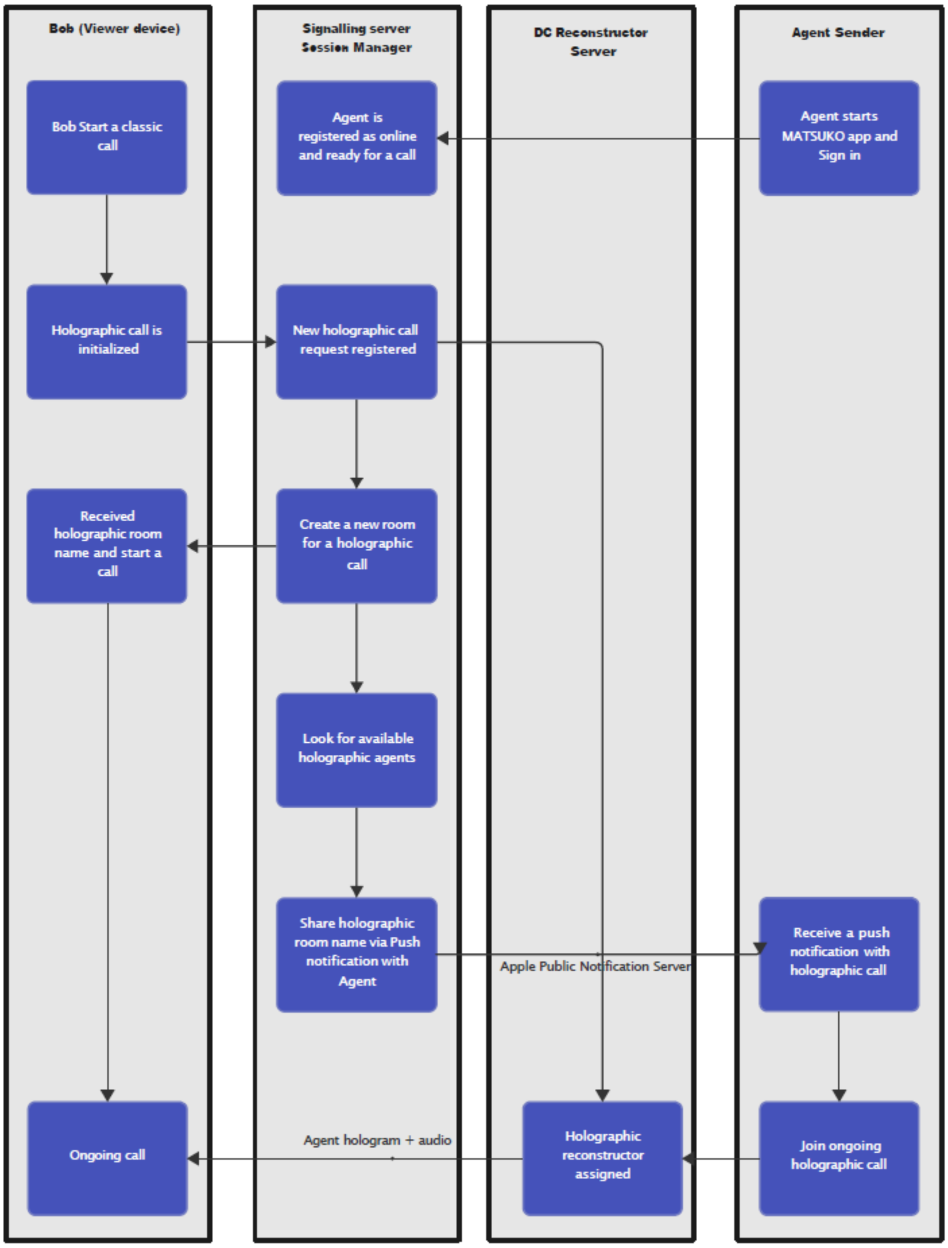

Figure 23: IMSDC Connectivity High Level Flow Draft

Information related to the connectivity between devices and the network elements is out of scope of this document.

11 SUMMARY

The main targeted outcomes of this document were to identify the Edge and Core network enablers to run 6G-XR's use cases and to start detailing the solution for those enablers.

After the analysis of requirements, the identified enablers are:

- E1.1 IEAP Edge orchestrator: will control the Madrid Edge platform resources and the LCM of the applications running at Madrid Edge. It will be used in the South Node for UC1 and UC2.
- E2.1 Barcelona Edge orchestrator: will control the Barcelona Edge platform resources and the LCM of the applications running at Barcelona Edge. It will be used in the South Node for UC1 and UC2.
- E3.1 North Node adapter: will translate the requests towards the North Node experimentation network. It will be used in the North Node for UC4 and UC5.
- E3.2 3D Digital Twin: will provide the resources in the North Node such as the Unity 3D Engine and a TSN switch that will allow the 3D Digital Twin use case validation. It will be used in the North Node for UC4.
- E3.3 Energy Management: will provide energy consumption data as input for the ML/DL algorithms for energy use optimization. It will be used in the North Node for UC5.
- E3.4 Resource Optimization: will enable radio resource re-allocation of the sliced 5G network in an optimal way for the North Node UOULU 5GTN.
- E4.1 QoD API: will allow the Application Function to request to the network the QoS it requires to work properly. It will be used in the South Node for UC1 and UC2.
- E4.2 Edge Cloud Discovery API: will allow the Application Function to discover the available Edge platforms. It will be used in the South Node for UC1 and UC2.
- E4.3 Traffic Influence API: will apply the most suitable network path according to the location of the UE. It will be used in the South Node for UC2.
- E5.1 Edge Federation: The Madrid and Barcelona Edge platforms will be federated, so Madrid Edge will know the available resources in Barcelona and will be able to use its capabilities. It will be used in the South Node for UC2.
- E6.1 Service Parameter API: The NEF in the South Node will offer an API that allows the Application Function to request a policy subscription for a UE or a group of UEs to use a certain S-NSSAI and DNN for its own application data flows. It will be used in the South Node for UC2.
- E6.2 Monitoring Event API: The NEF in the South Node will provide information to the Application Function such as the location of a UE or the congestion of a cell. It will be used in the South Node for UC1 and UC2.
- E6.3 AF Session with QoS API: The NEF in the South Node will allow the AF to choose from a list of QoS profiles. It will be used in the South Node for UC1 and UC2.
- E6.4 Analytics Exposure API: The NEF in the South Node will let the AF subscribe to a congestion notification service. It will be used in the South Node for UC1.
- E7.1 Cumucore Slice Creation API: will allow the creation and management of slices in the Cumucore solution. It will be used in the North Node for UC4.

- E8.1 OAIBOX: will allow the collection of power consumption metrics of its components as input for the Energy Management Framework in the North Node and will provide means to adjust its network configuration based on energy optimization decisions. It will be used for UC5.
- E9.1 IMS Data Channel Server: will allow to handle separately the signalling and the media flows within IMS sessions. It will be used in the South Node for UC3.

The up-to-date initial solution design of all these enablers has been described in this document. The work in tasks T2.1 to T2.4 will be continued until the final description of the solutions is described in D2.2.

REFERENCES

[1] D1.1 Requirements and Use Case Specifications, 6G-XR project, June 2023. https://www.6g-xr.eu/wp-content/uploads/sites/96/2023/10/D1.1-Requirements-and-use-case-specifications-V1.0.pdf

[2] GSMA Operator Platform Group – Requirements and Architecture, Version 5.0, July 2023 https://www.gsma.com/futurenetworks/resources/gsma-operator-platform-group-requirements-and-architecture-version-5-0/

[3] 5G!Drones, Unmanned Aerial Vehicle Vertical Applications' Trials Leveraging Advanced 5G Facilities, EU H2020 ICT-19-2019 Project. https://5gdrones.eu

[4] Microstack. Openstack for Kubernetes. https://microstack.run/

[5] 5G!Drones deliverables website. https://5gdrones.eu/deliverables/

[6] 5G!Drones gitlab URL. https://gitlab.5gdrones.droneradar.eu/

[7] Linux Foundation CAMARA project: https://camaraproject.org/

[8] Linux Foundation CAMARA project, QoD API: https://github.com/camaraproject/QualityOnDemand

[9] Alignment of the simple edge discovery API as per the CAMARA commonality group guidelines. Pull Request #122. https://github.com/camaraproject/EdgeCloud/pull/122

[10] GSMA Official Document OPG.02 – Operator Platform Telco Edge Requirements. https://www.gsma.com/futurenetworks/wp-content/uploads/2021/06/OPG-Telco-Edge-Requirements-2021.pdf

[11] ETSI GS MEC 003 V3.1.1 (2022-03): Multi-access Edge Computing (MEC); Framework and Reference Architecture https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.01.01_60/gs_MEC003v030101p.pdf

[12] ETSI White Paper No 49: MEC federation: deployment considerations https://www.etsi.org/images/files/ETSIWhitePapers/ETSI_WP_49_MEC-Federation-Deployment-considerations.pdf

[13] GSMA.com - GSMA Announces Details of Innovative Edge Cloud Applications and Dedicated Telco Edge Cloud Federation, 29 June 2021 https://www.gsma.com/futurenetworks/digest/gsma-announces-partnership-to-align-on-telco-edge-federation-with-etsi-and-3gpp/

[14] GSMA Official Document OPG.04 - East-Westbound Interface APIs, Version 3.0, 26 July 2023 https://www.gsma.com/futurenetworks/wp-content/uploads/2023/07/OPG.04-v3.0-GSMA-Operator-Platform-Group-East-Westbound-Interface-APIs-Version-3.0.pdf

[15] 5G System Overview, 3GPP, August 2022. https://www.3gpp.org/technologies/5g-system-overview

[16] 3GPP Technical Specification 23.502 V17.05, "Procedures for 5G System", 3GPP TS 23.502 description

[17] 3GPP Technical Specification 29.522 repository, 3GPP TS 29.522

[18] 3GPP Technical Specification 29.122 repository, 3GPP TS 29.122

[19] 3GPP Technical Specification 23.288 V17.04, "Architecture enhancements for 5G System to support network data analytics services", https://www.etsi.org/deliver/etsi_ts/123200_123299/123288/17.04.00_60/ts_123288v170400p.

pdf

[20]Cumucore 5G NC Specifications, Cumucore, 2023, "Cumucore_5G_NC_Product_Specifications_2023_March.docx.pdf", https://cumucore.com/mobile-private-network/

[21]CNC v5.5 Test RST API Cumucore Network Configuration, Cumucore, 2023, "5GC-CNC v5.5 Test-221123-085346.pdf"

[22]"Design Document Network Slicing and QoS management", Cumucore, June 2023

[23]OAIBOX, Open source 5G/6G network in a box, https://www.oaibox.com/

[24]Open Air Interface 5G implementation, Eurecom, https://gitlab.eurecom.fr/oai/openairinterface5g/

[25]GSMA Official Document NG.134, https://www.gsma.com/newsroom/wp-content/uploads/NG.134-v1.0-1.pdf

APPENDIX A: EXAMPLES OF NEF APIS CALLS

A.1 Example of 3GPP Service Parameter API

```
curl --http2-prior-knowledge -k -i -m 15 -X POST -H'content-type:application/json' -H'Authorization:Basic YWYxMjM6cGFzc3dvcmQ=' -d'{
    "afServiceId": "12345",
    "gpsi": "msisdn-52005275",
    "suppFeat": "32",
    "urspGuidance": [
        {
            "trafficDesc": {
                "appDescs": {
                    "CC93721B-8901-3456-1234-123456789123": {
                        "osId": "CC93721B-8901-3456-1234-123456789123",
                        "appIds": {
                            "1": "746573742e6f7267"
                        }
                    }
                }
            },
            "routeSelParamSets": [
                {
                    "precedence": 1,
                    "snssai": {
                        "sd": "C143A1",
                        "sst": 1
                    },
                    "dnn": "Ericsson-test31"
                }
```

Note that Ericsson's implementation of this feature is not using the Traffic Descriptor properties, so in the example, only the "routeSelParamSets" will be relevant, the "trafficDesc" properties must be in the request, but they will not be used.

A.2 Example of 3GPP Monitoring Event API

- Create LOCATION_REPORTING subscription:

```
{
  "supportedFeatures": "147F",
  "externalId": "100000001@ericsson.com",
  "notificationDestination": "https://af.example.com/event_report",
  "monitoringType": "LOCATION_REPORTING",
  "maximumNumberOfReports": 10,
  "monitorExpireTime": "2028-01-31T00:15:00+08:00",
  "repPeriod": 300,
  "locationType": "CURRENT_LOCATION",
  "accuracy": "CGI_ECGI",
  "minimumReportInterval": 1000
}
```

- Notification for LOCATION_REPORTING:

```
{
  "subscription": "https://example.com:27900/3gpp-monitoring-event/v1/EventMonitorApId/subscriptions/ffffffff-ffff-ffff-ffff-ffffffffffff",
  "monitoringEventReports": [
    {
```

```
      "externalId": "100000001@ericsson.com",
      "locationInfo": {
        "ageOfLocationInfo": 1000,
        "cellId": "460-15-584673"
      },
      "monitoringType": "LOCATION_REPORTING",
      "eventTime": "2028-01-31T00:15:00+08:00",
      "locationType": "LAST_KNOWN_LOCATION"
    }
  ]
}
```

A.3 Example of 3GPP AF Session with QoS API

- QoS reference provisioning in NEF:

```
{
  "afApplicationId": "TefQosTest.5TONIC",
  "mediaComponents": [
    {
      "afApplicationId": "TefQosTest.5TONIC",
      "maxRequestedBandwidthDl": "42 Mbps",
      "maxRequestedBandwidthUl": "42 Mbps",
      "mediaComponentNumber": 1,
      "mediaSubComponents": [
        {
          "flowNumber": 6,
          "maxRequestedBandwidthDl": "42 Mbps",
          "maxRequestedBandwidthUl": "42 Mbps"
        }
      ],
      "mediaType": 1,
      "minRequestedBandwidthDl": "42 Mbps",
      "minRequestedBandwidthUl": "42 Mbps"
    }
  ]
}
```

- AF request to create a new subscription resource:

```
{
  "self": "string",
  "supportedFeatures":
"#/TS29571_CommonData/components/schemas/SupportedFeatures",
  "notificationDestination": "string",
  "flowInfo": [
    {
      "flowId": 0,
      "flowDescriptions": [
        "string"
      ]
```



```
  }
 ],
 "ethFlowInfo": [
  "#/TS29514_Npcf_PolicyAuthorization/components/schemas/EthFlowDescription"
 ],
 "qosReference": "string",
 "altQoSReferences": [
  "string"
 ],
 "ueIpv4Addr": "string",
 "ipDomain": "string",
 "ueIpv6Addr": "string",
 "macAddr": "#/TS29571_CommonData/components/schemas/MacAddr48",
 "usageThreshold": {
  "duration": 0,
  "totalVolume": 0,
  "downlinkVolume": 0,
  "uplinkVolume": 0
 },
 "sponsorInfo": {
  "sponsorId": "string",
  "aspId": "string"
 },
 "qosMonInfo": {
  "reqQosMonParams": [

"#/TS29512_Npcf_SMPolicyControl/components/schemas/RequestedQosMonitoringParame
ter"
  ],
  "repFreqs": [
   "#/TS29512_Npcf_SMPolicyControl/components/schemas/ReportingFrequency"
  ],
  "repThreshDl": "#/TS29571_CommonData/components/schemas/Uinteger",
  "repThreshUl": "#/TS29571_CommonData/components/schemas/Uinteger",
  "repThreshRp": "#/TS29571_CommonData/components/schemas/Uinteger",
  "waitTime": "#/TS29571_CommonData/components/schemas/DurationSec",
  "repPeriod": "#/TS29571_CommonData/components/schemas/DurationSec"
 },
 "requestTestNotification": false,
 "websockNotifConfig": {
  "websocketUri": "string",
  "requestWebsocketUri": false
 },
 "dnn": "#/TS29571_CommonData/components/schemas/Dnn",
 "status": "string",
 "msisdn": "string",
 "imsi": "string"
}
```


- Response successful creation of subscription to AF:


```
{
 "self": "string",
 "supportedFeatures":
"#/TS29571_CommonData/components/schemas/SupportedFeatures",
 "notificationDestination": "string",
 "flowInfo": [
  {
   "flowId": 0,
   "flowDescriptions": [
    "string"
   ]
  }
 ],
 "ethFlowInfo": [
  "#/TS29514_Npcf_PolicyAuthorization/components/schemas/EthFlowDescription"
 ],
 "qosReference": "string",
 "altQoSReferences": [
  "string"
 ],
 "ueIpv4Addr": "string",
 "ipDomain": "string",
 "ueIpv6Addr": "string",
 "macAddr": "#/TS29571_CommonData/components/schemas/MacAddr48",
 "usageThreshold": {
  "duration": 0,
  "totalVolume": 0,
  "downlinkVolume": 0,
  "uplinkVolume": 0
 },
 "sponsorInfo": {
  "sponsorId": "string",
  "aspId": "string"
 },
 "qosMonInfo": {
  "reqQosMonParams": [

"#/TS29512_Npcf_SMPolicyControl/components/schemas/RequestedQosMonitoringParame
ter"
  ],
  "repFreqs": [
   "#/TS29512_Npcf_SMPolicyControl/components/schemas/ReportingFrequency"
  ],
  "repThreshDl": "#/TS29571_CommonData/components/schemas/Uinteger",
  "repThreshUl": "#/TS29571_CommonData/components/schemas/Uinteger",
  "repThreshRp": "#/TS29571_CommonData/components/schemas/Uinteger",
  "waitTime": "#/TS29571_CommonData/components/schemas/DurationSec",
  "repPeriod": "#/TS29571_CommonData/components/schemas/DurationSec"
 },
 "requestTestNotification": false,
 "websockNotifConfig": {
```

```
    "websocketUri": "string",
    "requestWebsocketUri": false
  },
  "dnn": "#/TS29571_CommonData/components/schemas/Dnn",
  "status": "string",
  "msisdn": "string",
  "imsi": "string"
}
```


A.4 Example of 3GPP Analytics Exposure API

- Create CONGESTION subscription:


```
{
  "analyEventsSubs": [
    {
      "analyEvent": "CONGESTION",
      "analyEventFilter": {
        "snssai": {
          "sst": 1,
          "sd": "D143A5"
        },
        "locArea": {
          "nwAreaInfo": {
            "ncgis": [
              {
                "plmnId": {
                  "mcc": "001",
                  "mnc": "002"
                },
                "nrCellId": "abcd00002"
              }
            ]
          }
        },
        "reptThlds": [
          {
            "congLevel": 20
          }
        ],
        "extraReportReq": {
          "accuracy": "LOW"
        }
      },
      "tgtUe": {
        "gpsi": "msisdn-52005275"
      }
    }
  ],
  "analyRepInfo": {
    "immRep": true,
```



```
    "notifMethod": "ON_EVENT_DETECTION",
    "repPeriod": 2
  },
  "notifUri": "http://af.example.com/v1/afid/subscriptions/0000001",
  "notifId": "0000001",
  "suppFeat": "8"
}
```


- Notification for CONGESTION:


```
{
  "notifId": "0000001",
  "analyEventNotifs": [
    {
      "analyEvent": "CONGESTION",
      "expiry": "2020-07-06T02:54:32Z",
      "timeStamp": "2120-07-06T02:54:32Z",
      "congestInfos": [
        {
          "locArea": {
            "nwAreaInfo": {
              "ncgis": [
                {
                  "plmnId": {
                    "mcc": "001",
                    "mnc": "002"
                  },
                  "nrCellId": "abcd00002"
                }
              ]
            }
          },
          "cngAnas": [
            {
              "cngType": "USER_PLANE",
              "tmWdw": {
                "startTime": "2021-07-05T02:54:32Z",
                "stopTime": "2120-07-06T02:54:32Z"
              },
              "nsi": {
                "congLevel": 1
              },
              "confidence": 2
            }
          ]
        }
      ]
    }
```

```
  ]
}
```