5G key technology enableRs for Emerging media COntent pRoDuction services

Deliverable D3.2

Complete description of 5G components

Version 2.0

Date: 2023/03/13

Document properties:

Grant Number:	957102
Document Number:	D3.2
Document Title:	Complete description of 5G Components
Editor(s):	Pablo Pérez (NOK)
Authors:	Pablo Pérez (NOK); Jan Dürre (SEN); Peter Brightwell (BBC); David Artuñedo (TID); Daniel Corregidor (UPM); Julián Cabrera (UPM); Baruch Altman (LiveU); Manuel Fuentes (5CMM); Héctor Donat (5CMM); Teresa Pardo (5CMM); Morten Brandstrup (TV2); Antonio Garcia (ACC); Mika Skarp (CMC); Raymond Knopp (EUR); Pierre Jean Muller (RED); Paola Sunna (EBU); Pavlo Kondratenko (EBU); Thorsten Lohmar (EDD).
Reviewers	Pablo Pérez (NOK); Jan Dürre (SEN); Morten Brandstrup (TV2); Peter Brightwell (BBC); Ian Wagdin (BBC), Cristina Avellan, Adrian Rodrigo, Nerea Cilleruelo (UPV).
Contractual Date of Delivery:	2022/05/31
Dissemination level:	PU[1]
Status:	Final
Version:	2.0
File Name:	5G-RECORDS_D3.2_v2.0

Revision History

Revision	**Date**	**Issued by**	**Description**
1.0	2022/05/31	NOK	EC Final Version
2.0	2023/03/13	Adrián Rodrigo (UPV)	Editorial changes

[1] CO = Confidential, only members of the consortium (including the Commission Services)

PU = Public

Abstract

The European H2020 5G-RECORDS Project aims at investigating whether and to what extent the technological improvements introduced by 5G network can be exploited for the production of professional multimedia content.

Within the project, three main use case are considered that could benefit most from the improvements related to the 5G network: live audio production, multiple camera wireless studio and live immersive media production. For each of them, new 5G components are being designed, developed and used by project partners to investigate the potential of using 5G. To this purpose, three different 5G network deployments are provided to the project, one for each of the considered use case.

This deliverable provides a general description of all the components involved and list all major design and development activities carried out in the project.

Keywords

5G, Audio-visual sector, Component description, Component development and integration, Component elements, end-to-end infrastructures, Non-public networks, Professional media content production.

Disclaimer

This 5G-RECORDS D3.2 deliverable is not yet approved nor rejected, neither financially nor content-wise by the European Commission. The approval/rejection decision of work and resources will take place at the Final Review Meeting, after the monitoring process involving experts has come to an end.

Executive Summary

5G-RECORDS is a European project aiming at investigating some of the relevant business opportunities that 5G brings in the context of the professional multimedia content production.

In this project, three main use cases are considered: live audio production, multiple camera wireless studio and live immersive media production. Each use case may have a variety of scenarios on how the technology may be deployed. For each use case, the project considers specialised end to end 5G infrastructures.

The scope of this deliverable is to report a description of all components and infrastructures involved and to report the progress made in WP3 regarding design and development of new 5G component elements.

This deliverable lists, for each use case, all the component elements involved, distinguishing between component elements available at the beginning of the project and component elements designed and developed in the context of the project. Components' descriptions are subdivided in different chapters according to the use case they are used in. Descriptions of 5G components shared among different use cases are collected in a separate chapter.

In the live audio production use case, the goal is to demonstrate the feasibility of using 5G professional audio equipment within a high-quality, ultra-low latency local wireless audio production network. For this purpose, besides end user equipment such as microphones and in-ear monitoring receivers, most of the work is being done on components that constitute the 5G infrastructure, on network configuration and on media orchestration, including network slicing management and shared access servers.

The multiple camera wireless studio use case looks at the possibilities of having some studio equipment, such as wireless 5G-enabled cameras in a 5G S-NPN (Standalone Non-public network). These could be co-located or separate to the broadcaster's production centre. In this use case, the activities carried out are mainly around the design and development of the Media Gateway (MG), the codec board and modem, a possible new workflow for camera control, and the Media Operational Control Layer (MOCL).

The last considered use case is the live immersive media production. It consists of an end-to-end chain for a content production based on Free Viewpoint Video (FVV). Two production components (FVV system and media delivery) and two infrastructure components (5G+MEC and SDN+Edge) are described.

One of the shared components is the Media Operational and Control Gateway (MOCG). This component controls the setup of media resources between a 5G enabled device such as a 5G camera microphone or other devices. It sits on top of the 5G network and is used to automate and simplify the operational control of broadcast equipment deployed in different locations. In production, this is done through AMWA NMOS APIs, specifically IS-05. In the 5G network, the use of these APIs needs to be adapted through the introduction of several middleware components decoupling the remote operations from the studio and replicating/emulating NMOS control functions within a 5G environment. A full description of this will be done in D3.3.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

The acronyms list has a special style defined as “acronyms”. Each acronym is separated by a tabulation with each definition. As is shown below:

3GPP	3rd Generation Partnership Project
5G	5th Generation of mobile communications systems
AF	Application Function
AMWA	Advanced Media Workflow Association
API	Application Programming Interface
AV	Audio-Visual
CPE	Customer Premise Equipment
CH	Content Handler
CU	Centralised Unit
CU-CP	CU-Control Plane
CU-UP	CU-User Plane
EC	European Commission
ETSI	European Telecommunications Standards Institute
FVV	Free Viewpoint Video
FWA	Fixed Wireless Access
HEVC	High Efficiency Video Coding
gNB	gNodeB
ICT	Information Communication Technology
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IEM	In-Ear monitoring
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IS (AMWA)	Interface Specification
ISO	International Standards Organization
ITU	International Telecommunications Union
JWT	JSON Web Toke
KPI	Key Performance Indicator
LSA	License Shared Access
MAC	Media Access Control
MCR	Master Control Room
MEC	Multi-access Edge Computing
MG	Media Gateway
MNO	Mobile Network Operator
MOC	Media Orchestration and Control
MOCG	Media Orchestration and Control Gateway
MPEG	Moving Picture Experts Group (formally, ISO/IEC JTC 1/SC 29/WG 11)
MQTT	Message Queuing Telemetry Transport
Near-RT RIC	Near Real-Time Intelligent Controller
NEF	Network Exposure Function
NMOS	Network Media Open Specifications
NSA	Non Stand Alone
NPN	Non-Public Network
NVF	Network Function Virtualisation
NSA	Non-standalone
OPNFV	Open Platform for NFV
O-RAN	Open Radio Access Network

PCF	Policy Control Function
PLMN	Public Land Mobile Network
PMSE	Programme Making and Special Events
PNI-NPN	Public Network Integrated NPN
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
REST	REpresentational State Transfer, an architectural style used in APIs
SA	Standalone
SDP	Session Description Protocol
SIM	Subscriber Identification Module
S-NPN	Standalone NPN
SME	Small and medium-sized enterprise
SMPTE	Society of Motion Picture and TV Engineers
SDR	Software Defined Radio
ST	(SMPTE) Standard
TAI	International Atomic Time
UC	Use Case
UE	User Equipment (the 5G "device”)
UPF	User Plane Function
vRAN	Virtualised RAN
VSF	Video Services Forum
WAN	Wide Area Network

1 Introduction

1.1 Overview on 5G-RECORDS

In recent years, the new 5G technology and the related innovations are enabling new business opportunities in various industrial fields. One field that could benefit from this evolving scenario is multimedia content production. In this respect, the Horizon 2020 project named "5G-RECORDS" aims to investigate the possibility of exploiting 5G technology in professional audio and video applications.

In particular, in 5G-RECORDS new 5G components specific to the field of professional production of media content are designed, developed, and integrated. These components will be validated within a 5G infrastructure provided by project partners.

In this project, three main use cases are considered: live audio production, multiple camera wireless studio, and live immersive media production.

1.2 Scope of this deliverable and structure

Deliverable D3.2 aims to provide a complete description of the 5G components involved in the project, documenting the overall design, as well as the actual implementation of the project framework. The document completely covers the description of the components developed specifically within three project use cases. It also provides a brief description of the media orchestration components, which will be further detailed in deliverable D3.3.

The document is structured in the following way:

- *Chapter 1* introduces the objectives of the 5G-RECORDS project, related activities and describes the purpose of this document.
- *Chapter 2* lists all the component deployed, specifying what component elements were developed during the project.
- *Chapters 3 , 4 & 5* provide a complete description of 5G components involved and give a brief overview of the considered use cases, respectively named "Live Audio Production", "Multiple Camera Wireless Studio", and "Live Immersive Content Production".
- *Chapter 6* provides a brief description of the media orchestration services, which will be further covered in D3.3.

2 Components in 5G-RECORDS

2.1 List of all the components

The lists of components are shown in tables below according to each use case in which they are used.

Table 1: List of components for UC1

Component	Category	Deployment location	In charge of
Local Audio Processing	End device	On-site	SEN
Audio User Terminal	End device	On-site	SEN, EUR
Remote Server	Media orchestrator & management Server	Internet	SEN
5G RAN (inc. Shared Access Client)	Network	On-site	ACC, EUR
5G Core	Network	On-site	CMC
Shared Access Server	Shared Access Server	RED Technologies premises	RED
Time Service	Time Service	On-site	SEN
Network Slice Management	Network Component	Cloud	CMC

Table 2: List of components for UC2

Component	Category	Deployment Location	In charge of
Media Gateway	Video receiver + decoder	Aachen & Turin	EBU
5G Network	Network	Aachen	EDD
5CMM 5G Modem	Network equipment	Aachen	5COM
IM Encoder	A/V encoder	Aachen	IM
MCR	MCR	Cloud	EBU
LU800	Video tx + encoder	Aachen	LU
LU2000-SMPTE	Video rx + decoder	Turin	LU

Table 3: List of components for UC3

Component	Category	Deployment Location	in charge of
5G-Ready FVV Live	Free-Viewpoint Video Capture and Production system	Smart Venue & Near-Edge	UPM
Compact 5G Network (+ MEC)	Network	Smart Venue & Near-Edge	NOK

Media Delivery	VNF	Edge	NOK
Delivery cloud & SDN	Network component	Edge	TID

Table 4: List of components shared across multiple UCs

Component	Category	Deployment location	In charge of
Media Orchestration and Control Gateway	Media orchestrator	Cloud	BBC

2.2 Components' elements: what was available & what was developed so far

This section presents the list of the components' elements development during 5G-RECORDS within each use case.

Table 5: List of component elements developed during the project for UC1

Component	Existing elements (available before the beginning of the project)	Elements designed / developed / integrated during the project
Local Audio Processing	Rapid Prototyping Audio Hardware Platform	FPGA and software functions: • Network Audio Source/Sink • Audio Device Control Client • PTP Client • Audio Mixer
Audio User Terminal	Rapid Prototyping Audio Hardware Platform	FPGA and software functions: • Audio Device Control Client • Network Audio Source/Sink • PTP Client
Remote Server	-	Software functions: • Audio Device Control Server • Media Operational Control Gateway • Core Configuration Service
5G RAN (inc. Shared Access Client)	Software functions: • eMBB 5G CU • near-RT-RIC • 4G Shared Access Client	Software functions: • URLLC 5G CU • 5G Shared Access Client Open5Glab capabilities: • Several. Please refer to Table 1 of D4.1 [1]
5G Core	Basic 5G Core	• Slicing management capability including external API • Slicing functionality in 5G core • PCF

Shared Access Server	Path loss computation	• Lease creation • Protection of leases • Synchronization with leases database • Leases visualization
Time Service	-	Software functions: • PTP Server

Table 6: List of component elements developed during the project for UC2

Component	Existing elements (available before the beginning of the project)	Elements designed / developed / integrated during the project
Media Gateway	• Gstreamer-based framework • Typescript RESTful API framework	• RESTful API to manage media content handlers • Project-specific graphical user interface • Specific Gstreamer pipelines • Integration of NVIDIA Rivermax in the Gstreamer pipelines • Integration of RIST libraries in the pipelines • Configuration of the NVIDIA Jetson based image for the requirements of the project
Network Slice Management	• N5 • NSSF user interface	• MOG to N5
5G Network	• Trial network already available	• Test network is developed during the project
5CMM 5G Modem	The 5G modem has been designed and developed from scratch based on the needs and requirements of 5G-RECORDS and considering connectivity towards the encoder.	• Designed/developed: 5G Ethernet/USB board for broadband communications. • Integrated: 5G Rel-15 modem, 4x4 MIMO antennas, SIM card, microprocessor.
IM Encoder	Origami Square Bamboo FPGA Module	• Origami Square Carrier board for Broadcast Camera IO ZCU-106 IO board for Broadcast Camera
MCR	On-premise, S2110-based MCR, incl. signal acquisition (e.g. satellite downlink), processing, monitoring, switching and operations.	Public cloud end-to-end “MCR” chain, i.e. signal contribution (JPEG-XS), processing (AWS-CDI) and signal egress.

LU800	Device announced with project start, continued development with the period.	New capabilities and features: • adding Sierra 5G module for 5G SA and its integration/configuration • enhancing the IP pipe for the control traffic.
LU2000-SMPTE	Pre-commercial version already available.	New capabilities and features: • multi-networks support (master clock, input, output), • configurations • robustness • ongoing work on lower latency.

Table 7: List of component elements developed during the project for UC3

Component	Existing elements (available before the beginning of the project)	Elements designed / developed / integrated during the project
5G-Ready FVV Live	• Capture servers • 3 cameras per server • 2 GPUs per server (depth computation + encoding	• Capture simulation module for transmitting pre-recorded FVV sequences. • RTP traffic smoother. • Configurable MTU size.
	Smartphone to control virtual camera position show synthetied view	PC-based production console
	Online view renderer over desktop hardware platform	Online and Offline version that works reading pre-recorded video file. • RTP packet reordering capability. • Single-GPU working mode. • Logging statistics generation during execution • Integration in MEC hardware platform.
		Stream selector architecture which allows several virtual views rendered in parallel with the same set of cameras
Media Proxy / Media Delivery Server	Adaptive Streaming Server	Integration of adaptive streaming server, modules for RTP video processing, transcoding configuration, monitoring tools

<table>
<tr><td rowspan="2">5G Deployment / MEC</td><td>5G gNB in 3.5 GHz using Nokia radio.
Distributed core to support 5G NSA in 3.5 GHz</td><td>5G Platform:
Modem configuration and position to optimise performance
Integrate and configure mmWave equipment (antenna, baseband, controllers) in n257. Fine tune radio options for performance.
Update core software release and configuration to support n257.</td></tr>
<tr><td></td><td>MEC platform:
New hardware platform acquired and integrated.
OpenStack configuration.
Availability of GPU within OpenStack VM.</td></tr>
<tr><td rowspan="3">Delivery cloud / End-to-End SDN</td><td>Telefonica Edge infra-structure available on Peñuelas Central Office based on OCP servers, switches and storage.</td><td>A tenant has been configured in Telefonica Edge solution, to ensure compute, storage and network capacity for the required deployments to guarantee UC3 needs.</td></tr>
<tr><td>Existing SDN inside Telefonica Edge and Telefonica transport network.</td><td>New SDN Components:
Conditional DNS
Slice selector
Monitoring elements to manage the required service level and slices.
New edge cloud tenant connected to Telefonica Transport Network using two slices multimedia and best effort. Both slices connected to Segovia central office (local users) and to remote users in the Spain geography</td></tr>
<tr><td></td><td>A hmtl5 web player based on videojs to serve the video and collect metrics.</td></tr>
</table>

Table 8: List of component elements developed during the project for components shared across multiple UCs

Component	Existing elements (available before the beginning of the project)	Elements designed / developed / integrated during the project

Media Operational Control Gateway	• Sony open source NMOS C++ Node + Registry implementation (nmos-cpp)
	• Sony open source NMOS JavaScript Client implementation (nmos-js)
		• Designed general architecture for Media Orchestration and Control Gateway
		• Developed enhancements to nmos-cpp and nmos-js to additional functionality:
		    ○ Registration and connection of device "through the gateway"
		    ○ Compressed (non-2110) stream types
		    ○ Carriage of camera control information in IS-07

3 UC1 Components - Live Audio Production

In a typical live audio production, such as a concert, musical, theatre or studio performance, one or more artists perform live (i.e., acting, dancing, singing, or playing music instruments) either to create content that can be used later or to entertain an interested audience that can be present on site or follow the content live via a stream.

This use case will focus on an example AV production scenario to demonstrate the feasibility of 5G for professional audio equipment targeting a local wireless high-quality and ultra-low latency audio production network. For further details see D2.1 [2].

3.1 General Architecture

Figure 1 shows the general architecture which is used in this project to realise the UC1 scenario. Multiple 5G-enabled microphones and in-ear monitoring receivers are connected to a local 5G network. An in-ear mix for each IEM can be generated in a local audio processing device that is connected via Ethernet cable to the 5G system (5GS). Ultimately all audio devices should be controllable by a user over the Internet. Therefore, a media orchestration controller (MOC) translates the user inputs into NMOS control commands (for more information on NMOS, see section 6). A Media Orchestration Control Gateway (MOCG) translates these NMOS commands to device-specific protocols. Through a core configuration service (CCS) this MOCG also configures the QoS flows in the 5G network to allow best performance for the media devices. For spectrum management a shared access service allows user interaction.

All audio devices and the 5G network itself are synchronised to a common time base to avoid increasing jitter through drifting.

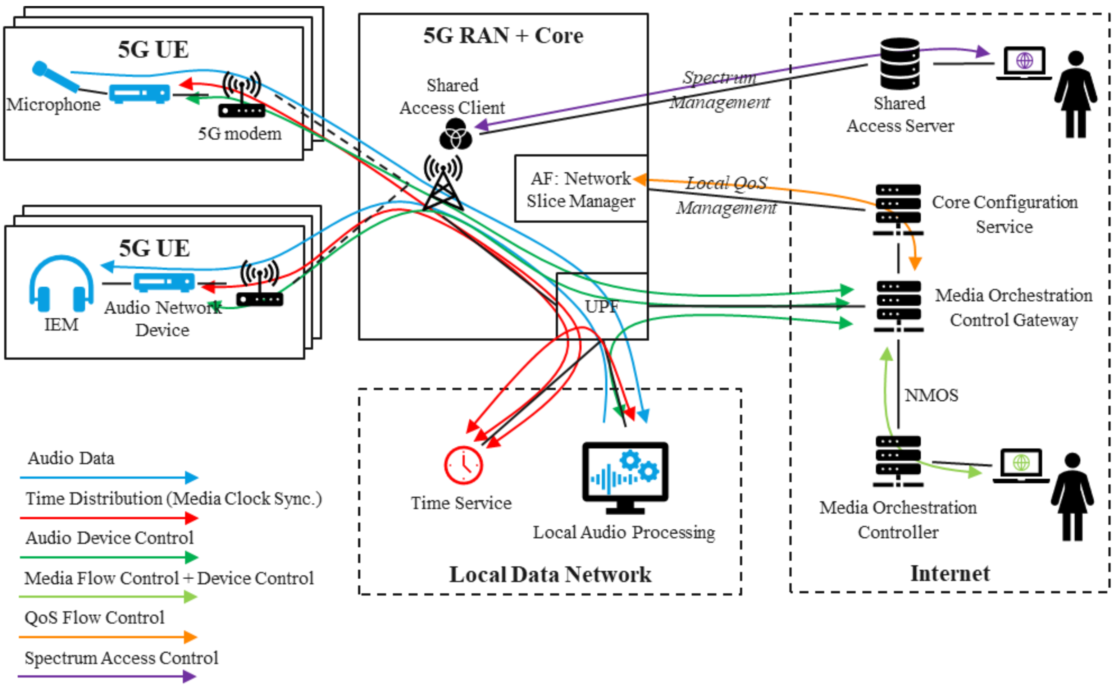


Figure 1: High level architecture of use case 1 (live audio production)

3.2 Control Data Flows

The control data flows of this use case are described in this section.

3.2.1 Setup 5G Network and connect UEs

To connect a UE to the local 5G network first the 5GS itself must be configured properly. Figure 2 shows the necessary steps. A spectrum access server (SAS) gets a frequency lease from the national regulator. When the 5GS powers up a list of channels is negotiated with this SAS server. When the RAN is operational the UE can connect and establish the default PDU session for basic connectivity.

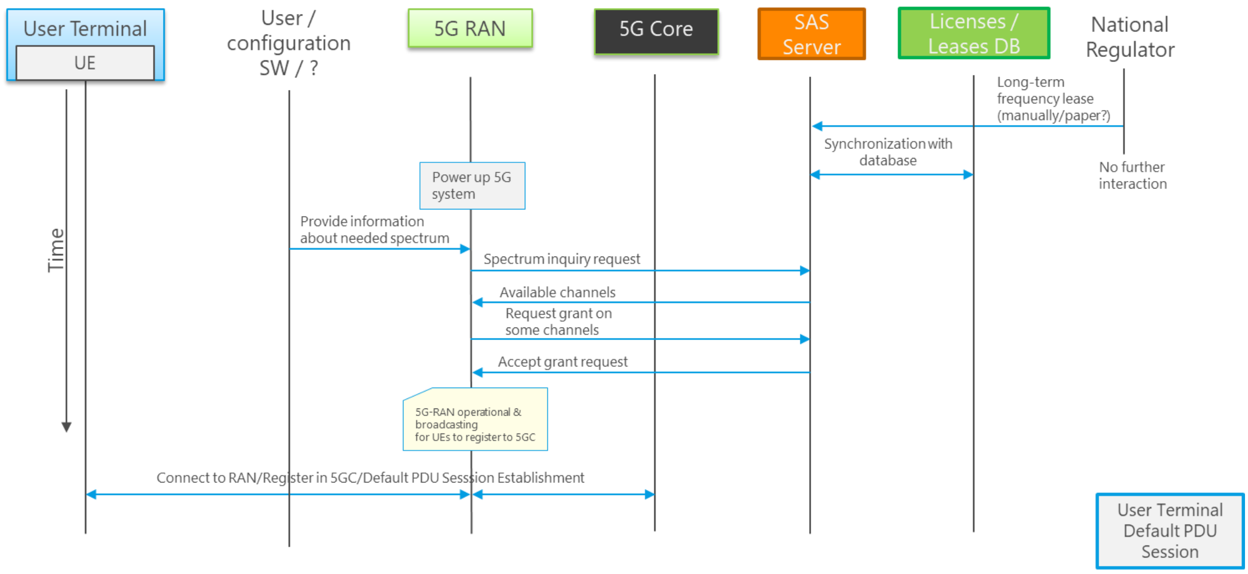


Figure 2: Data flow for setup 5G network and connect UE

3.2.2 Discovery of Audio Devices and establishing Audio Data Stream

When audio devices connect to the network, they need to be known to the media orchestration and control layer. Figure 3 shows this process. The audio devices announce their capabilities to the MOCG which allows the MOC to setup streams. When the MOC requests streams, the core configuration service (CCS) requests the required QoS flows from the 5GS. If the 5GS has capacity for these flows a new QoS-PDU session per UE is created to carry the audio payload. The MOCG configures the audio devices and the MOC shows the user that everything is up and running.

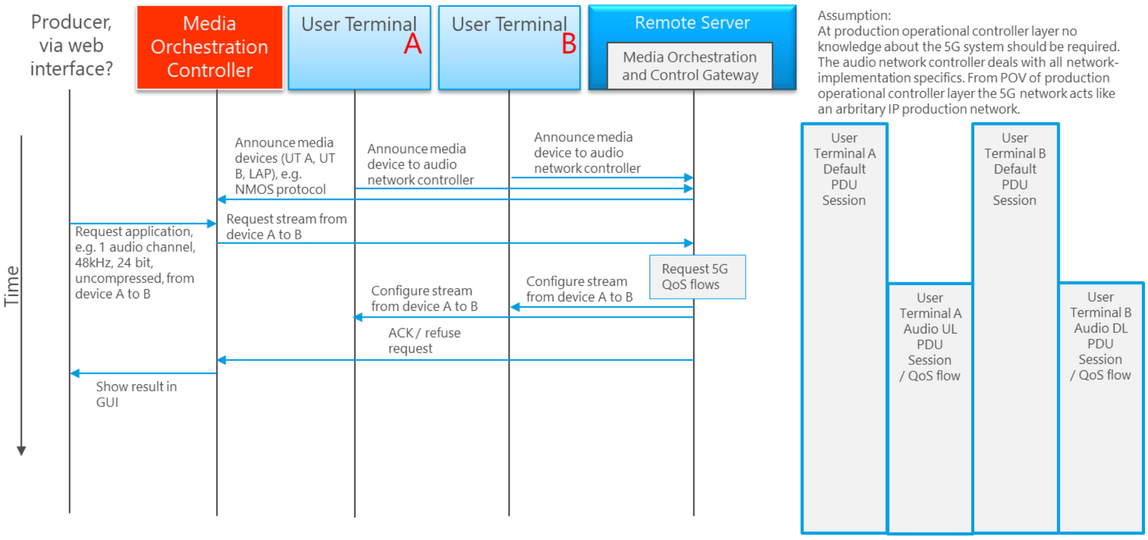


Figure 3: Data flow for device discovery and audio stream establishment

The process of requesting QoS flows from the 5GS is shown in Figure 4. The CCS requests a QoS flow from the network slice manager (NSM) which checks the availability of radio resources with the shared access client (SAC) in the RAN.

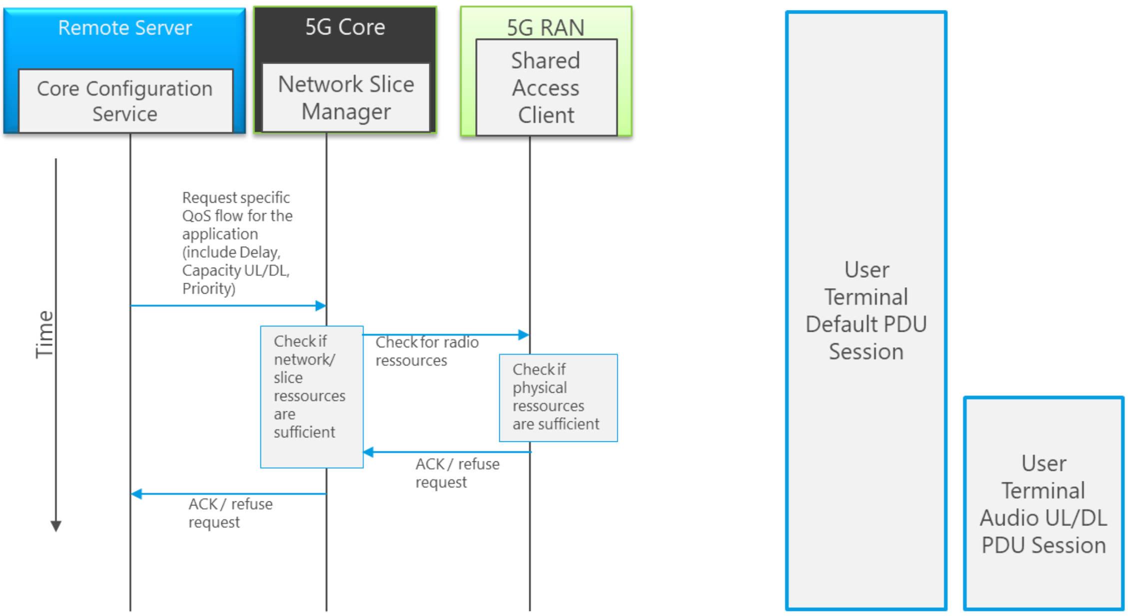


Figure 4: Data flow for requesting a QoS flow from the 5G network

3.3 Delay Budget

A key requirement for this use case is streaming latency. For a deeper understanding of the influence of different components and to identify potential for optimisation a detailed delay budget will be created. The impact on latency of the following functions will be identified either by calculations or measurements.

The delay budget for uplink and downlink includes the functions/parameters listed in Table 9 and Table 10.

Table 9: Delay budget for uplink audio transmission

function / parameters in UL	Description
Periodicity of transfers (related to 5GS)	-
UE layers processing (assuming RLC UM)	RAN protocol layers transmission/reception procedures (PHY, MAC, RLC, PDCP) including headers addition/removal
Slot boundary	Interval until the end of a slot
Scheduling request delay (SR periodicity)	Overhead associated with the transmission of requests for Uplink resources from the UE and the transmission of the Uplink grant from the gNB back to the UE, before the UE can start using the allocated resources on the Uplink. Constant bit rate scheduling is planned to be used instead of periodic SR to avoid this overhead.
Radio propagation	Over the air data propagation delay
RU/DU processing (inc. GTP-U)	Digital front end processing, lower PHY layer procedures (RU), physical link transfer delay between RU and DU, upper PHY, MAC , RLC transmission/ reception procedures (DU), GTP-U encapsulation/decapsulation to/from the CU.
DU to CU-UP networking	Physical link transfer delay between DU and CU-UP
CU-UP processing (inc. GTP-U)	35 µsecs (measured using Pingtool)
CU UP to UPF networking	N3 interface as specified by 3GPP
UPF processing	Encapsulation and decapsulation of GPRS Tunnelling Protocol for the user plane. Packet routing and forwarding. Per-flow QoS handling, including transport level packet marking for downlink.
UPF to DN networking	N6 interface as specified by 3GPP

Table 10: Delay budget for downlink audio transmission

Function / parameters in DL	Description
Periodicity of transfers (related to 5GS)	-
DN to UPF networking	N6 interface as specified by 3GPP
UPF processing	Encapsulation and decapsulation of GPRS Tunnelling Protocol for the user plane. Packet routing and forwarding. Per-flow QoS handling, including transport level packet marking for uplink (UL).
UPF to CU-UP networking	N3 interface as specified by 3GPP
CU-UP processing (inc. GTP-U)	55 µsecs (measured using Pingtool)
CU-UP to DU networking	Physical link transfer delay between CU-UP and DU

DU/RU processing (inc. GTP-U)	Digital front end processing, lower PHY layer procedures (RU), physical link transfer delay between RU and DU, upper PHY, MAC, RLC transmission/ reception procedures (DU), GTP-U encapsulation/decapsulation to/from the CU.
Slot boundary	Interval until the end of a slot
Radio propagation	Over the air data propagation delay
UE layers processing	RAN protocol layers transmission/reception procedures (PHY, MAC, RLC, PDCP) including headers addition/removal

3.4 5G End-to-end user plane protocol stacks

The end-to-end 5G user plane stacks involved in the communication between the audio source/output application on the UE side and the audio source/output application (local audio processing) on the Data Network side are shown in Figure 5. The figure shows the path of the user data packets across the different protocols involved. It is worth noting that although it is not explicitly shown with the red/blue lines, the protocols of the O-RAN Fronthaul interface would also be involved in a fully disaggregated DU and RU according to O-RAN Alliance specifications.

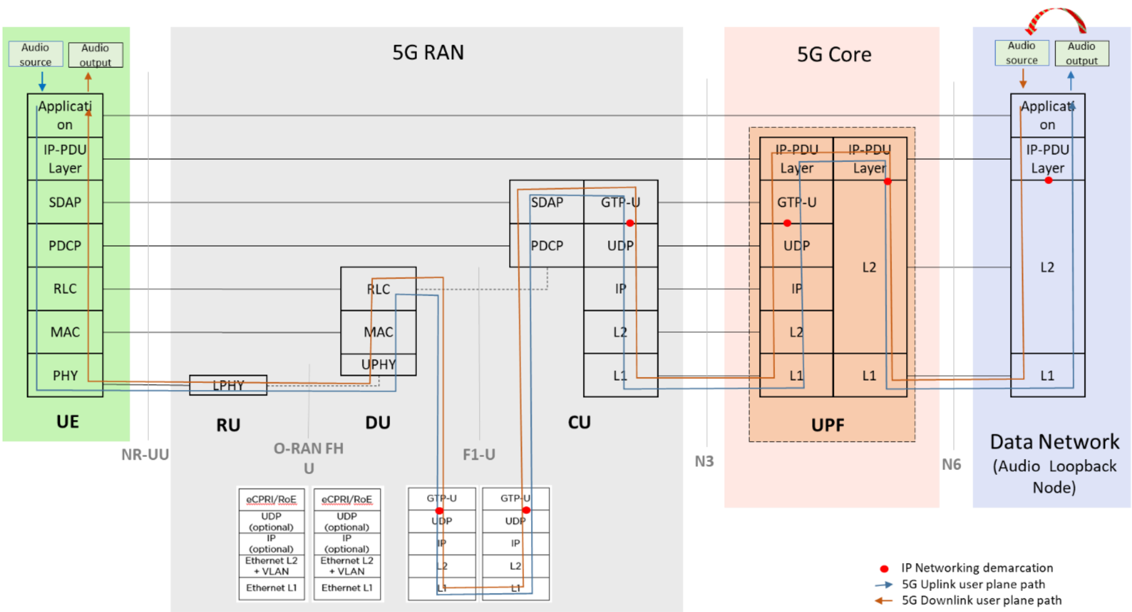


Figure 5: End-to-end user plane protocol stacks

3.5 List of Components

3.5.1 Component: Local Audio Processing

The local audio processing device is a stationary device connected wired to the 5G system. It is located on-site to allow low-latency interaction with the wireless audio equipment. Figure 6 shows the architecture of the local audio processing device. Network audio streams from microphones are received, mixed, and sent to IEM receivers in the 5G network. Additionally, local analogue audio inputs could be used in the IEM-mix. Analog outputs allow connection of speakers or headphones for monitoring. Table 11 describes the functionalities of the local audio processing device.

The device is based on a custom developed hardware including an FPGA for signal processing and an ARM CPU that runs the device control software.

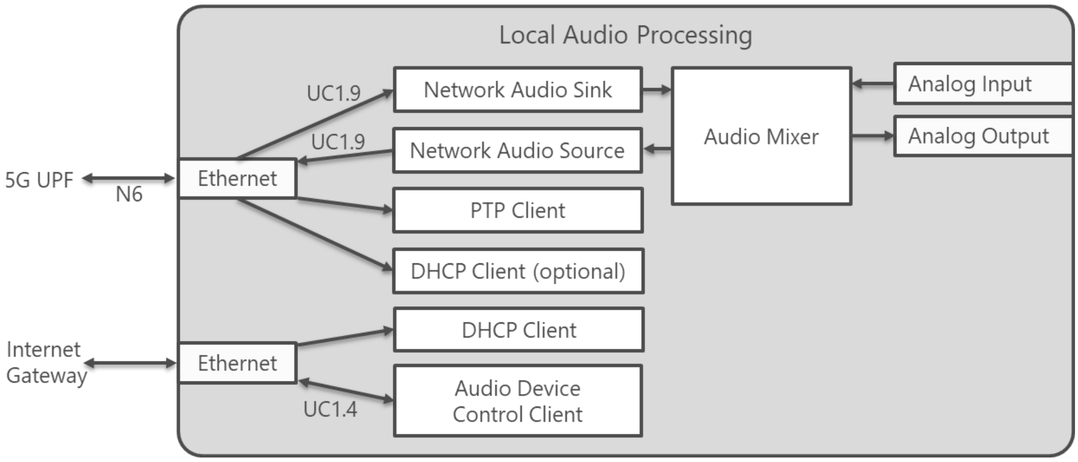


Figure 6: Block diagram of local audio processing component

Table 11: Functions of the UC1 local audio processing component.

Function	Description
Network Audio Source/Sink	The device can receive and transmit a network audio stream using interface/protocol UC1.9.
Audio Mixer	The device can mix local and network audio streams and forward them to local audio out or the network.
PTP Client	PTP client for synchronisation
DHCP Client	The two Ethernet interfaces must be supplied with an IP via DHCP. One has the option for a static IP setting.
Audio Device Control Client	Allows device control from the remote server via interface UC1.4.
Analog Input/Output	To add local audio sources to the mixes or to monitor via headphones/speakers

3.5.2 Component: Remote Server

The remote server is software that is installed locally or on the Internet that provides an interface between the Media Orchestration Controller (MOC) and the audio devices in the local network. Figure 7 shows the architecture of the remote server component.

A main function of the server is to provide a Media Orchestration and Control Gateway (MOCG) for UC1. It advertises the audio network devices capabilities to the MOC. The MOCG receives the desired configuration from the MOC and configures the audio network devices accordingly. The MOCG sets up and terminates audio data flows as requested by the MOC. This configuration is done via the audio device control server that translates the MOC commands to the device specific protocol.

The MOCG configures the QoS flows in the 5GS via the Core Configuration Service. This translates the desired configuration from the MOC into network resource requirements and communicates these to the core (Network Slicing Manager).

Table 12 summarises the functionalities of the remote server component.

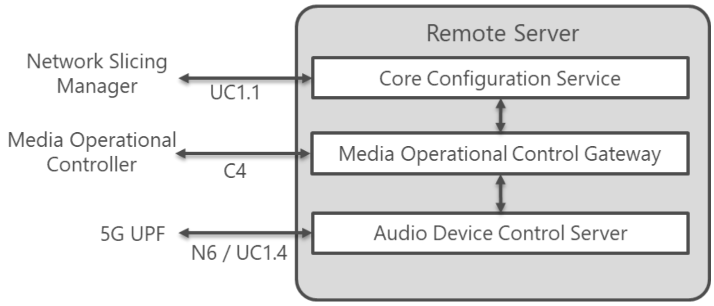


Figure 7: Block diagram of Remote Server

Table 12: Functions of the UC1 remote server component.

Function	Description
Media Orchestration Control Gateway	Interface between media devices, 5G network configuration and MOC
Core Configuration Service	Configures QoS flows in 5GS
Audio Device Control Server	Remote control of the audio devices

3.5.3 Component: Audio part of User Terminal

In this scenario both 5G-enabled microphones and IEMs are used. These are realised by a similar hardware/software implementation. Each of these Audio UEs consists of an audio part and a 5G part (the modem). The audio part of the UE is based on a custom developed hardware including an FPGA for signal processing and an ARM CPU that runs the device control software. The audio- and the 5G-part are connected by a 1 Gbit/s Ethernet connection.

The device can be configured to act as a microphone transmitter or an IEM receiver. Therefore, the device has analogue interfaces to connect a microphone and/or headphones. IP audio streams can be received or transmitted, and the audio network device converts between analogue and IP domain.

To control the functionality of the device it connects to a remote audio device control server. A PTP client for time synchronisation allows media clock synchronisation.

Figure 8 shows the architecture of the audio network device. An overview of its functionalities is given in Table 13.

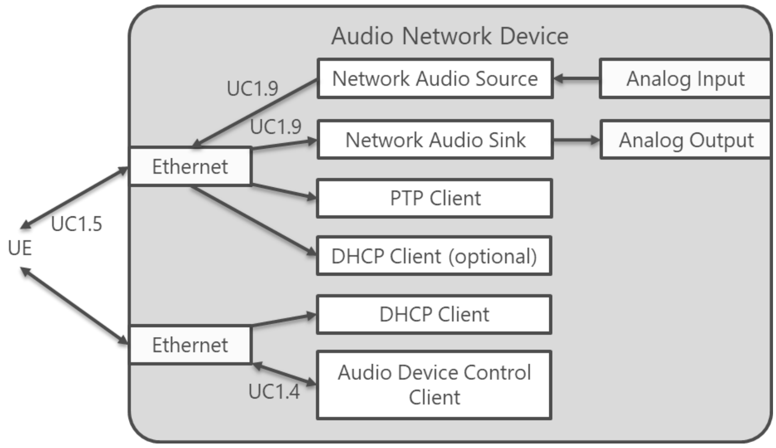


Figure 8: Audio Network device and UE block diagram

Table 13: Functions of the UC1 Audio UE component.

Function	Description
Network Audio Source/Sink	The device can receive and transmit a network audio stream using interface/protocol UC1.9.
PTP Client	PTP client for synchronisation
DHCP Client	The two Ethernet interfaces must be supplied with an IP via DHCP. One has the option for a static IP setting.
Audio Device Control Client	Allows device control from the remote server via interface UC1.4.
Analog Input/Output	To connect a microphone or a headphone.

3.5.4 Component: 5G Radios and COTS UEs

The following Radio Units (RU) are available in EURECOM's open5GLab: eight N310 USRPs, three X310 USRPs, two N320 USRPs, two AW2S eCPRI Jaguar, and one Benetel 5G RRU. For the 5G-RECORDS project, the radios being used in the first phase and to be used in the second phase for testing are: USRP N310 (2x2, soon 4x4, 2x10G UHD fronthaul), AW2S Panther (4x4, 100 MHz, 2x10G eCPRI split-8 fronthaul), and Mavenir M44VA (3.6-3.8 GHz, 4x4, 100 MHz, 2x10G eCPRI ORAN-7.2). Two commercial off-the-shelf (COTS) user terminals (UTs) are also being used (i.e., SIMCom Wireless Solutions Ltd. SIM8202G-M2 and Quectel Wireless Solutions Co., Ltd. RM500Q-GL). The features of the radios used for the first integration phase are presented below.

Ettus Research USRP N310 SDR

An SDR platform is a radio communication system that allows implementation of radio components handling signal processing via software running on general-purpose processor (e.g., personal computer or embedded system) rather than special-purpose hardware. The SDR cards used in the first phase of the 5G-RECORDS project are Ettus N310 USRPs from National Instruments.

The USRP N310 provides reliability and fault-tolerance for deployment in large-scale and distributed wireless systems. It has been selected to enable common reproduction of the OpenAirInterface (OAI) code, an open-source software project founded by EURECOM that provides a fully 3GPP compliant reference implementation. The USRP N310 device simplifies control and management of a network of radios by introducing the unique capability to remotely perform tasks such as updating software, rebooting, factory resetting, self-testing, host PC/ARM debugging and monitoring system health. It is one of the highest channel density devices in the SDR market, offering four RX and four TX channels in a half-wide RU form factor. The RF front end uses two AD9371 transceivers. Each channel provides up to 100 MHz of instantaneous bandwidth and covers an extended frequency range from 10 MHz to 6 GHz. Furthermore, the open-source USRP Hardware Driver (UHD) API and RF Network-on-Chip (RFNoC) FPGA development framework reduce software development effort and integrate with a variety of industry-standard tools such as GNU Radio.

Figure 9: Ettus Research USRP N310 SDR Board

USRPs B210 will be configured to operate in 3.3-3.4 GHz Band n78 with SCS 30 kHz. The USRP is connected through USB 3.0 to a GIGABYTE BRi5-8250 model mini-PC with Intel core i5-8250U (4 cores @ 3.4 GHz max) which is capable of further incorporating a Sennheiser audio UT and a Sennheiser local audio processing engine into the system via USB 3.0, USB 3.1 or 1Gb Ethernet standards for the OAI based UE side. Thus, there are three main interconnections on the PC: (i) one towards the radio, (ii) another for control of the Ethernet over USB3, and (iii) finally the application interconnection with the Sennheiser audio device. The corresponding devices will be connected over the monitoring interface(s) to a Eurecom's lab switch so that UC1 5G-RECORDS partners doing experimentations can go in and perform their tests.

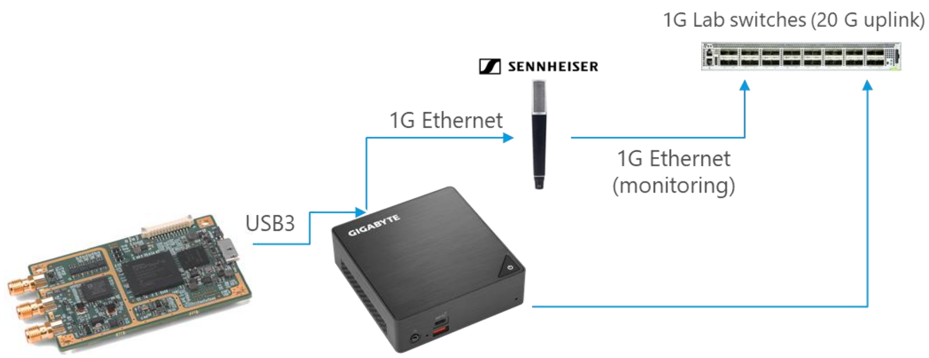


Figure 10: 5G-based UE prototype.

SIMCom Wireless Solutions Ltd. SIM8202G-M2

SIM8202G-M2 is a sub 6 GHz multi-band mobile communications UE module that supports the 3GPP Release 15 standalone (SA) 5G. EURECOM operates it within an industrial computer on a Pico-ITX form factor motherboard over the M.2 connector and using all 4 of its antenna ports.

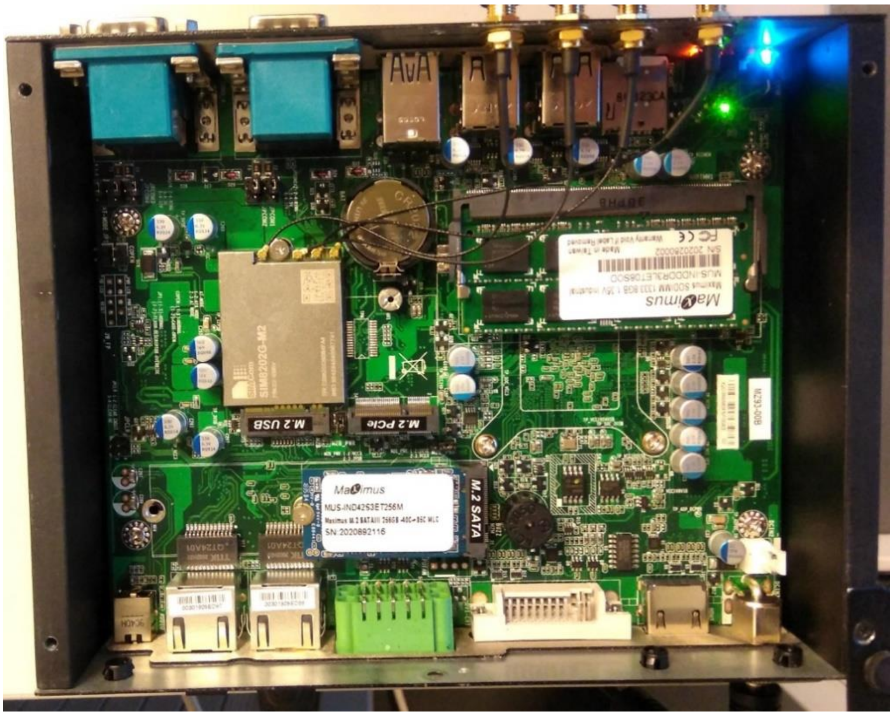


Figure 11: SIMCom SIM8202G-M2 module embedded on a Pico-ITX motherboard

Quectel Wireless Solutions Co., Ltd. RM500Q-GL

RM500Q-GL is another mobile communications UE module supporting the 3GPP Release 15 SA 5G New Radio (NR) specifications. It is deployed on a credit-card sized "UP Board", and again, all 4 of its antenna interfaces are used.

Figure 12: Quectel RM500Q-GL module connected over USB to an "UP Board"

3.5.5 Component: 5G RAN

The overview of the NG-RAN architecture in 3GPP is depicted in Figure 13.

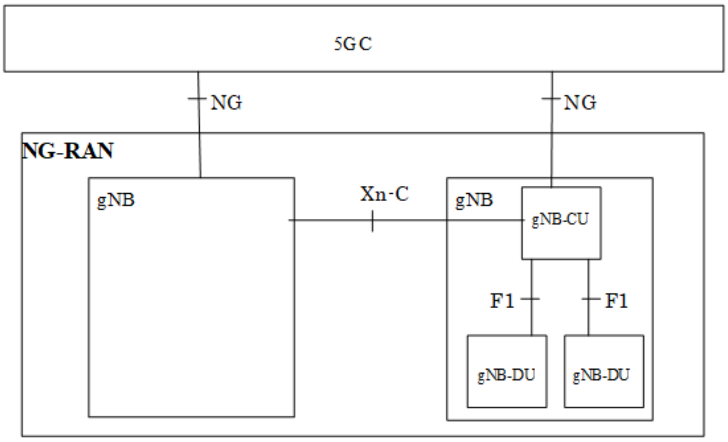


Figure 13: 3GPP NG-RAN architecture

3GPP 5G NR allows splitting of the gNB functionality into three logical modules: Radio Unit (RU), provisioned with RF circuitry; the Distributed Unit (gNB-DU), hosting gNB real-time functions; and the Centralised Unit (gNB-CU), hosting gNB non-real-time functions. 3GPP has defined a normative interface between the CU and DU components according to an HLS (High Layer Split) defined as F1. Additionally, CUPS (Control User Plane Separation) at CU level is enabled via normative E1 interface between CU CP and CU UP components as in Figure 14.

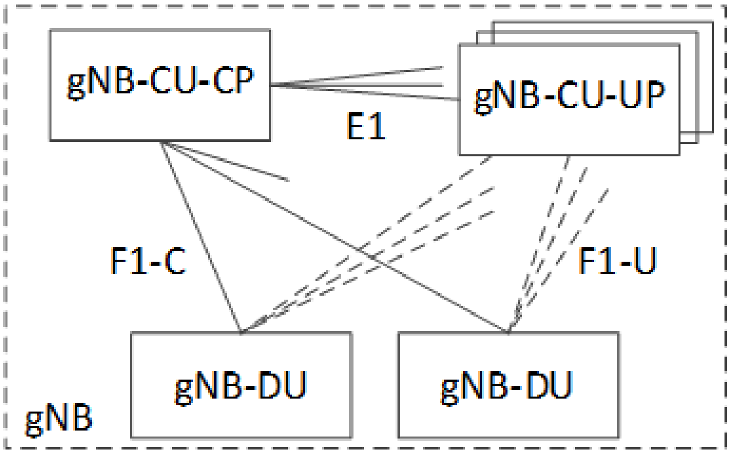


Figure 14: 3GPP gNB components

3GPP has not defined a normative interface for the Low Layer Split (LLS) between the DU and the RU but has recommended either one based on Split 7.2 (defined in the context of O-RAN Alliance) or Split 6 (defined in the context of Small Cell Forum). Different deployment options and topologies are therefore possible, some of which are shown in Figure 15.

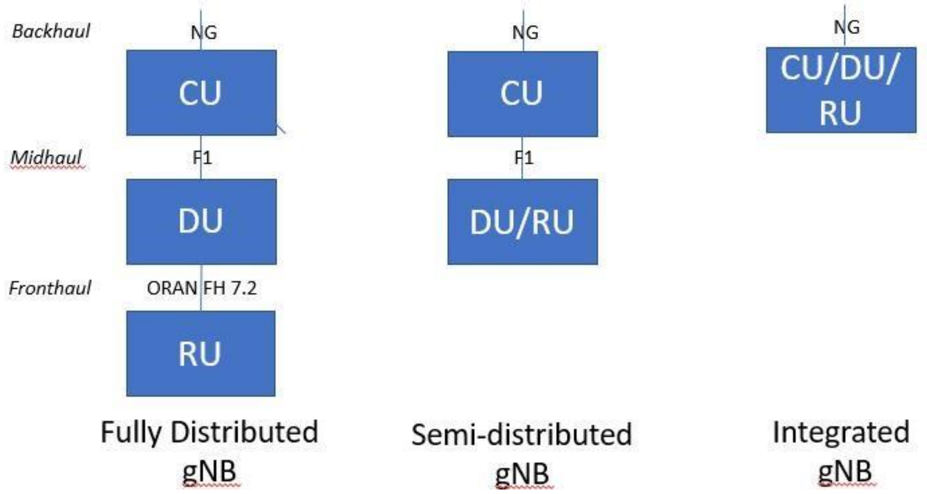


Figure 15: gNB deployment topologies

As part of O-RAN Alliance additional interfaces and components are being defined and standardised in addition to the standard 3GPP ones, with the general reference architecture shown in Figure 16.

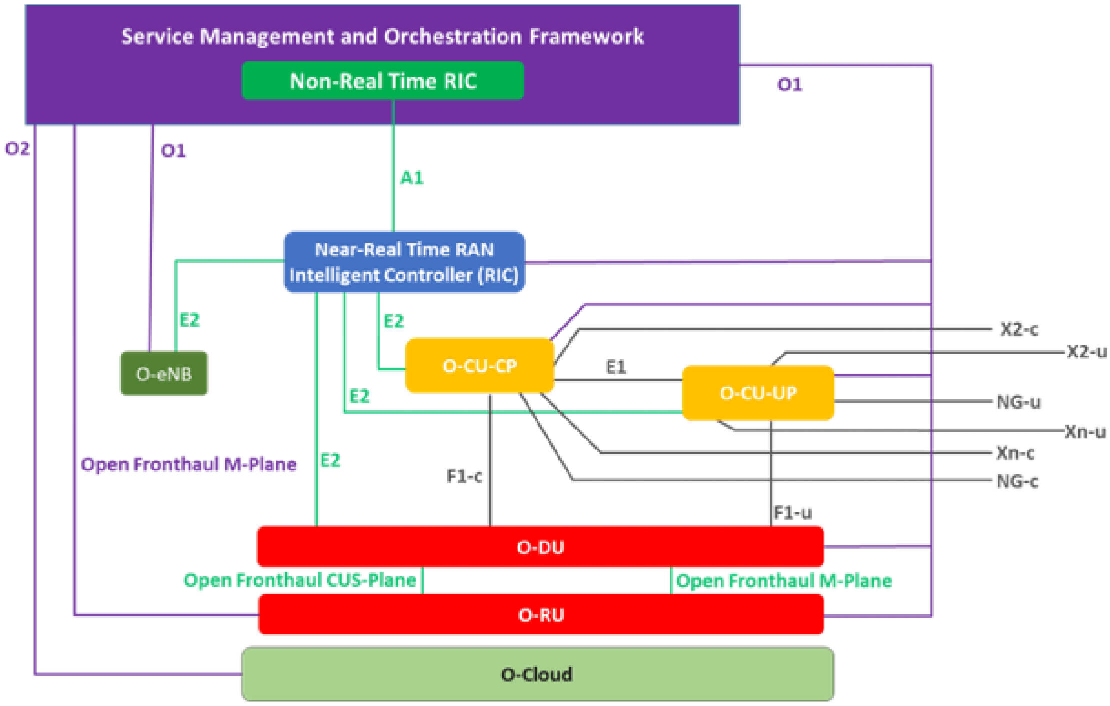


Figure 16: O-RAN Alliance reference architecture

3.5.5 nRT-RIC, CU and Shared Access Spectrum client

Accelleran brings dRAX™ is the marketing name for Accelleran´s virtual RAN product line, a cloud native and O-RAN aligned 5G SA vRAN solution consisting of a near-RT (nRT) RIC, CU-CP, CU-UP and xApp framework components. It is engineered to provide an open and extensible software framework for the control plane functions of 4G and 5G RAN and aligns with Open RAN architecture principles defined by both 3GPP and the O-RAN Alliance. dRAX™ is aligned to O-RAN architecture, but not all interfaces are fully interoperable, as most of them are still being defined in O-RAN Alliance.

dRAX™ is a genuinely cloud-native architecture based on containerised microservices communicating with each other via an asynchronous messaging framework. Each of the major components of the RAN (CU-CP, CU-UP, near-RT RIC) are themselves disaggregated into a fine-grained set of service entities. For example, the CU-CP is composed of a set of collaborating services handling: (i) F1AP connections to DU, (ii) E1AP connections to CU-UP, (iii) NGAP connections to AMF, and (iv) RRC connections to UEs.
The lifecycle (deployment, upgrade, scaling requirement) of these components are managed independently and they are unaware of location since the dRAX™ messaging framework handles routing of all messages between services.

The near-RT RIC is a key component of dRAX™. It supports the deployment of xApps (again as microservices) and provides them with services in the context of the dRAX™ environment including a fully productised xApp SDK. dRAX™ leverages Accelleran field-proven RAN software and is compatible with Accelleran 4G CE certified carrier-grade small cells and supports 5G gNB using standards-based DU/RUs from the developing ecosystem of 5G Open RAN.

The key features of dRAX™ are summarised in Table 14.

Table 14: Key Features of dRAX

Key feautures of dRAX	Description
O-RAN aligned vRAN	The Accelleran dRAX™ vRAN platform delivers a true multi-vendor, disaggregated and open virtualised RAN Intelligence aligned with the O-RAN Alliance. Implementing 3GPP Control User Plane Separation (CUPS), the user and control planes are fully decoupled. Support for 3rd party DUs, RUs & e/gNB encourage an open disaggregated eco-system to bring innovative 4G/5G products to the commercial market.
Open Orchestration & Data APIs	dRAX™ is open. Orchestration supports the industry preferred NFV/SDN framework APIs above and the NIB (Network Information Base) data APIs support industry standards and best practices (NFV/SDN, O-RAN, OAM, 3GPP, Netconf/Yang, …).
4G and 5G SA support	dRAX™ was field proven before for LTE and has been integrated also with 5G SA support, leveraging standards-compliant DU/RUs from the Open RAN ecosystem.
Extensible RIC xApps	dRAX™ provides an open platform and fully productised SDK for the development of customised RAN intelligence, either by the customer, Accelleran or 3rd parties
Scalability	dRAX™ can be implemented on a single microserver for the smallest edge cloud. At the other end of the scale, dRAX™ is designed to scale to clusters of hundreds of cells and run in hyperscalers.
Mission Critical Reliability	All dRAX™ code is written to Accelleran's unique set of SW development standards based on established practices from safety critical industries.

Accelleran dRAX™ in Figure 17 will be enhanced to support the requirements of the live audio production use case and will be integrated with the other needed 5G components on the associated use case testbed, namely the 5GC from Cumucore, DU/RU from Eurecom and Shared Access Spectrum server from RED Technologies.

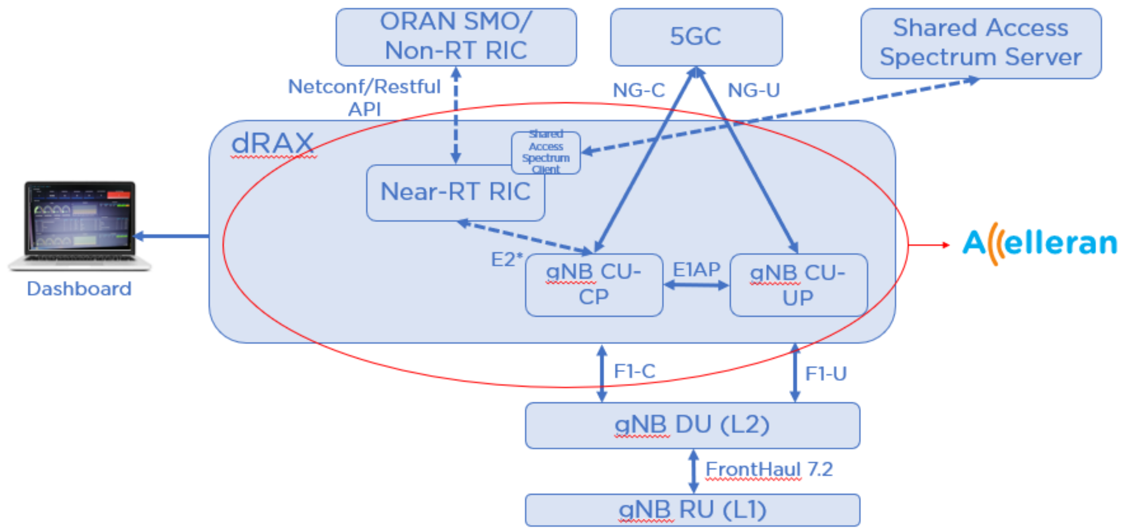


Figure 17: Accelleran dRAX™ in UC1 Live Audio Production

Near-RT RIC:

The near-RT RIC (Near Real-Time RAN Intelligent controller) is a key component of dRAX™. It has been developed as a true open development platform where 3rd parties can leverage dRAX™ open data and control interfaces.

It supports the deployment of containerised xApps and provides them with services in the context of the dRAX environment:

- xApp on-boarding and lifecycle management,
- Access to real-time RAN measurements and events,
- Configuration of RAN components,
- Real-time commands to direct RAN behaviour (e.g., force a handover, sub band masking),
- Real-Time state database,
- Inter xAppS communication,
- API-driven xApp configuration and policy management.

CU-CP and CU-UP:

The gNB-CU-CP is a logical node hosting the RRC and the control plane part of the PDCP protocol of the gNB-CU for a gNB. The gNB-CU-CP terminates the E1 interface connected with the gNB-CU-UP, the F1-C interface connected with the gNB-DU and the NG-C (N2) interface connected to the AMF in the 5GC.

The gNB-CU-UP is a logical node hosting the user plane part of the Packed Data Convergence Protocol (PDCP) protocol and the Service Data Adaptation Protocol (SDAP) protocol for a gNB. The gNB-CU-UP terminates the E1 interface connected with the gNB-CU-CP, the F1-U interface connected with the gNB-DU and the NG-U (N3) interface connected to the UPF in the 5GC.

Figure 18 shows the internal CU-CP and CU-UP components.

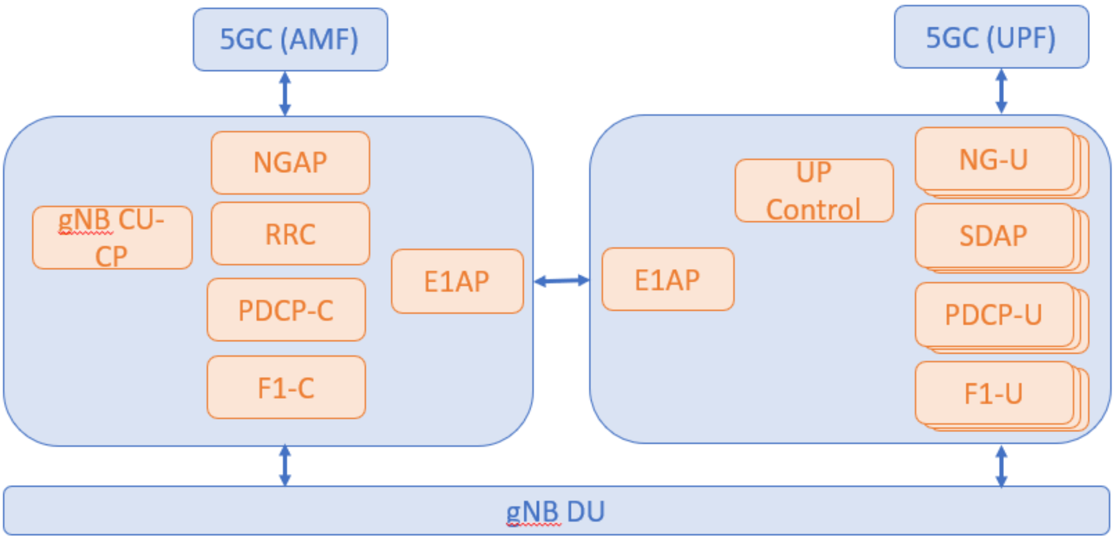


Figure 18: Accelleran CU components

Shared Access Spectrum client:

The Accelleran will enable a Shared Access Spectrum client as a cloud native microservice that communicates and interoperates with RED Technology Shared Access Spectrum server for the acquisition of 5G shared spectrum dynamically and the configuration of the RF parameters automatically in the OAI DU/RU.

3.5.5.2 *DU and CU*

OpenAirInterface (OAI) is an open-source project delivering implementations of both 5G NSA and SA RAN. OAI gNB is able to support an end-to-end SA setup based on a 5G CN (OAI or other) and SA capable COTS UE devices. As there are many developments taking place in parallel and capturing the whole RAN stack, intermediate validation steps have taken place using first the OAI UE (which is also developed in parallel to support SA deployments) in simulation and RF mode, and then the COTS UE devices.

The 5G SA access mode does not depend on legacy 4G LTE. However, it requires a new 5G core network (5GC). This new 5GC uses a cloud-aligned Service-Based Architecture (SBA) that supports control-plane function interaction, re-usability, flexible connections, and service discovery that spans all functions. The main 5GC functions are AMF, SMF, NRF and UPF (SPGW-U-tiny), all of which have been implemented in OAI and can easily be deployed using docker-compose [3].

As per the 3GPP Specification Series 38, compared to NSA, in SA the gNB needs to also implement the complete RRC layer and handling of all the associated messages as well as the NGAP to interface with AMF (N2 interface) and UPF (N3 interface). Moreover, the gNB needs to support multiple bandwidth parts as the initial access happens only on the initial bandwidth part, which has a smaller bandwidth than the full cell bandwidth. Further support for contention based random access is needed, as well as support for common and dedicated control channels. Only after the initial connection and authentication with the AMF is the full bandwidth part configured and used for user-plane traffic.

From a deployment perspective, two options are provided for OAI gNB: The monolithic and the CU/DU functional split modes. As it has been introduced before, the first option corresponds to a single gNB program on a single host running the whole 5G NR RAN stack. In the second option, the OAI gNB portion is divided into two blocks: the Central Unit (CU) that contains the implementation of RRC and PDCP layers, and the Distributed Unit (DU) that contains the implementation of RLC, MAC and PHY layers. The two units communicate with each other over the F1-C interface for the control plane and configuration exchanges based on F1AP protocol. The Downlink and Uplink user plane data transfer is made through the F1-U interface over GTP-U protocol. The CU and DU portions can thus run as separate programs in different hosts, offering significant flexibility for the deployment of the OAI 5G SA setup and the interoperability of OAI blocks with other commercial CUs or DUs. The CU/DU functional split mode uses Accelleran CU with OAI DU.

Figure 19 depicts the 5G RAN protocol architecture of the OAI gNB according to the CU/DU functional split deployment. The layers that had to be extended to support end-to-end SA functionality are highlighted in yellow.

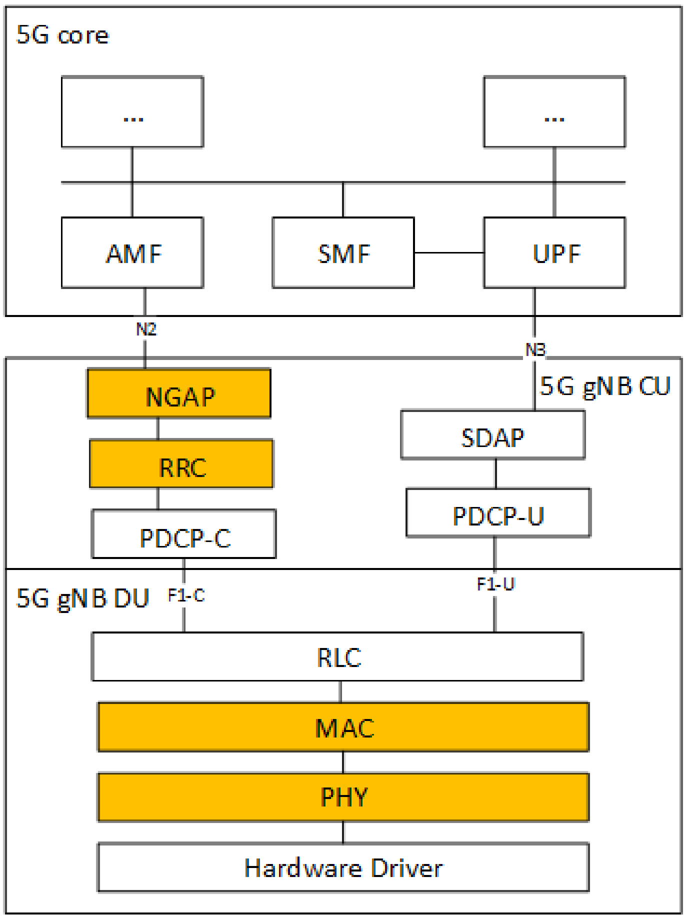


Figure 19: OAI gNB protocol architecture supporting 5G SA with CU/DU split

Interoperability tests of the OAI gNB with different 5G CNs and UE components from different vendors are ongoing. Specifically, interoperability with the OAI CN and the Nokia SA Box has been fully validated. With respect to the UE components, interoperability has been fully validated with the Quectel RM500Q-GL module and OAI UE.

In the following, some checkpoints for the validation of the end-to-end SA setup are provided to highlight the establishment of the 5G radio connection, the UE registration to the 5G CN, the PDU session establishment for the exchange of user plane traffic and some basic traffic testing. The underlined tests were performed using the Quectel RM500Q-GL module.

After the UE synchronises to the 5G cell and receives the System Information messages from the gNB, it initiates the contention based random access procedure (CBRA) to connect to the 5G cell. The procedure is finalised through the reception of Msg4 (*RRCSetup*) acknowledgment from the gNB (Figure 20). Then the UE replies with the *RRCSetupComplete* message which encapsulates the *NAS registration request* message towards the AMF. Upon reception of RRC Setup Complete, the UE state at the gNB becomes RRC Connected.

```
1170 [0m [0m[NR_MAC]   (ue 0, rnti 0x70bd) Received Ack of RA-Msg4. CBRA procedure succeeded!
1171 [0m [0m [34m[RLC]   RB found! (channel ID 1)
1172 [0m [0m[RLC]   /home/timarque/panos/openairinterface5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:452:deliver_sdu: delivering SDU (rnti 28861 is_
     1 rb_id 1) size 92
1173 [0m [0m[NR_RRC]   Received message NR_RRC_DCCH_DATA_IND
1174 [0m [0m[NR_RRC]   Decoding DCCH : ue 28861, inst 0, ctxt 0x7f2ba7ffeeb0, size 86
1175 [0m12 c0 00 20 00 04 f7 e0 17 b6 ce d3 60 47 e0 04 14 90 00 bf 20 5f 51 00 10 00 00 00 00 02  [0m[NR_MAC]    884. 4 RNTI 70bd: 3 bytes from DC
     (ndata 3, remaining size 14)
1176 [0m [0m[NR_MAC]    884. 4 RNTI 70bd: 0 bytes from DCCH 1 (ndata 3, remaining size 8)
1177 [0mb2 e0 2f 07 07 10 03 07 e0 04 14 90 00 bf 20 5f 51 00 10 00 00 00 00 02 b1 00 10 02 e0 2f 07 02 f0 50 40 1d 14 3a 55 20 5f 51 00 00 00 11 80
     17 40 00 09 05 30 10 10
1178 [0m[NR_RRC]   [FRAME 00000][gNB][MOD 00][RNTI 70bd] [RAPROC] Logical Channel UL-DCCH, processing NR_RRCSetupComplete from UE (SRB1 Active)
1179 [0m [0m[NGAP]   [gNB 0] Build NGAP_NAS_FIRST_REQ adding in s_TMSI: GUAMI amf_set_id 0 amf_region_id 1 ue 70bd
1180 [0m [0m[NR_RRC]   [FRAME 00000][gNB][MOD 00][RNTI 70bd] UE State = NR_RRC_CONNECTED
```

Figure 20: Successful CBRA procedure and reception of RRC Setup Complete message at the gNB

This message is conveyed transparently from the gNB to the AMF through the NGAP *InitialUEMessage* (Figure 21). A sequence of NGAP/NAS messages are exchanged afterwards between the gNB, the UE and the AMF to perform the authentication and security procedures leading to the successful Registration of the UE to the AMF (*Registration Accept* and *Registration Complete* NAS messages).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.18.203	192.168.69.131	NGAP	134	NGSetupRequest
2	0.003745	192.168.69.131	192.168.18.203	NGAP	614	NGSetupResponse
3	0.077278	192.168.18.203	192.168.69.131	NGAP/NAS-5GS	146	InitialUEMessage, Registration request
4	0.094364	192.168.69.131	192.168.18.203	NGAP/NAS-5GS	630	DownlinkNASTransport, Authentication request
5	0.123432	192.168.18.203	192.168.69.131	NGAP/NAS-5GS	146	UplinkNASTransport, Authentication response
6	0.126800	192.168.69.131	192.168.18.203	NGAP/NAS-5GS	462	DownlinkNASTransport, Security mode command
7	0.135349	192.168.18.203	192.168.69.131	NGAP/NAS-5GS/NAS-…	174	UplinkNASTransport, Security mode complete, Registration request
8	0.140345	192.168.69.131	192.168.18.203	NGAP/NAS-5GS	1302	InitialContextSetupRequest, Registration accept
9	0.257311	192.168.18.203	192.168.69.131	NGAP	122	UERadioCapabilityInfoIndication
10	0.459484	192.168.18.203	192.168.69.131	NGAP	86	InitialContextSetupResponse
11	9.338930	192.168.18.203	192.168.69.131	NGAP/NAS-5GS	118	UplinkNASTransport, Registration complete
12	9.341520	192.168.69.131	192.168.18.203	NGAP/NAS-5GS	710	DownlinkNASTransport, Configuration update command
13	9.341561	192.168.18.203	192.168.69.131	NGAP/NAS-5GS	162	UplinkNASTransport, UL NAS transport, PDU session establishment request
14	9.355758	192.168.69.131	192.168.18.203	NGAP/NAS-5GS	266	PDUSessionResourceSetupRequest, DL NAS transport, PDU session establishment accept
15	9.356056	192.168.18.203	192.168.69.131	NGAP	214	PDUSessionResourceSetupResponse

Figure 21: NGAP/NAS exchanges with the Core Network for UE Registration and PDU Session establishment

The UE then initiates the PDU Session Establishment which is validated through the *PDU Session Establishment Accept* NAS message coming from the CN. This message contains the IP address of the UE provided from the SMF. In Figure 22 the configured IP address is shown through the Quectel connection manager software.

```
[06-01 11:01:51:500] Quectel_QConnectManager_Linux_V1.6.0.16
[06-01 11:01:51:500] Find /sys/bus/usb/devices/1-1.3 idVendor=0x2c7c idProduct=0x800, bus=0x001, dev=0x014
[06-01 11:01:51:501] Auto find qmichannel = /dev/cdc-wdm0
[06-01 11:01:51:501] Auto find usbnet_adapter = wwan0
[06-01 11:01:51:501] netcard driver = qmi_wwan, driver version = 22-Aug-2005
[06-01 11:01:51:501] ioctl(0x89f3, qmap_settings) failed: Operation not supported, rc=-1
[06-01 11:01:51:501] Modem works in QMI mode
[06-01 11:01:51:508] cdc_wdm_fd = 7
[06-01 11:01:51:594] Get clientWDS = 15
[06-01 11:01:51:625] Get clientDMS = 1
[06-01 11:01:51:658] Get clientNAS = 4
[06-01 11:01:51:690] Get clientUIM = 1
[06-01 11:01:51:722] Get clientWDA = 1
[06-01 11:01:51:754] requestBaseBandVersion RM500QGLABR11A02M4G
[06-01 11:01:51:882] requestGetSIMStatus SIMStatus: SIM_ABSENT
[06-01 11:01:51:914] requestGetProfile[1] oai.ipv4///0
[06-01 11:01:51:946] requestRegistrationState2 MCC: 0, MNC: 0, PS: Detached, DataCap: UNKNOW
[06-01 11:01:51:977] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[06-01 11:01:51:977] ifconfig wwan0 0.0.0.0
[06-01 11:01:51:979] ifconfig wwan0 down
[06-01 11:03:22:967] requestRegistrationState2 MCC: 505, MNC: 1, PS: Detached, DataCap: UNKNOW
[06-01 11:03:22:999] requestRegistrationState2 MCC: 505, MNC: 1, PS: Detached, DataCap: UNKNOW
[06-01 11:03:23:031] requestRegistrationState2 MCC: 505, MNC: 1, PS: Detached, DataCap: UNKNOW
[06-01 11:03:41:088] requestRegistrationState2 MCC: 505, MNC: 1, PS: Detached, DataCap: UNKNOW
[06-01 11:03:41:729] requestRegistrationState2 MCC: 505, MNC: 1, PS: Attached, DataCap: 5G_SA
[06-01 11:03:41:985] requestSetupDataCall WdsConnectionIPv4Handle: 0x3b50c6b0
[06-01 11:03:42:113] ifconfig wwan0 up
[06-01 11:03:42:114] busybox udhcpc -f -n -q -t 5 -i wwan0
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 192.198.0.2
udhcpc: lease of 192.198.0.2 obtained, lease time 7200
[06-01 11:03:42:190] ./netup wwan0 192.198.0.2
```

Figure 22: Quectel module connection manager

At the same time, the gNB also sends a dedicated *RRCReconfiguration* message to the UE containing the configuration of the DRB that should be established at the UE to enable the user plane traffic flow at the RAN stack (PDCP, RLC, MAC layers). The UE replies with a *RRCReconfigurationComplete* message to signal the successful reconfiguration. Upon reception of the Reconfiguration Complete message the gNB performs its own DRB configuration for the lower layers, as can be seen from the figure above. In parallel, the gNB establishes a GTP-U tunnel with the UPF to enable the user-plane traffic flow over the N3 interface.

```
[0m[NR_RRC]   Receive RRC Reconfiguration Complete message UE 70bd
[0m[0m[NR_RRC]   Configuring PDCP DRBs/SRBs for UE 70bd
[0m[0m[PDCP]   /home/timarque/panos/openairinterface5g/openair2/LAYER2/nr_pdcp/nr_pdcp_oai_api.c:add_drb:878: added DRB for UE RNTI 70bd
[0m[0m[NR_MAC]   Modified UE_id 0/70bd with CellGroup
[0m[0m[NR_MAC]   Adding SchedulingRequestconfig
[0m[0m[NR_MAC]   Adding BSR config
[0m[0m[NR_MAC]   Adding TAG config
[0m[0m[NR_MAC]   Adding PHR config
[0m[0m[NR_MAC]   Adding LCID 1 (SRB 1)
[0m[0m[NR_MAC]   Adding LCID 2 (SRB 2)
[0m[0m[NR_MAC]   Adding LCID 4 (DRB 4)
[0m[0m[NR_RRC]   Configuring RLC DRBs/SRBs for UE 70bd
[0m[0m[RLC]   Trying to add SRB 2
[0m[0m[RLC]   /home/timarque/panos/openairinterface5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:710:add_rlc_srb: added srb 2 to UE with RNTI
[0m[0m[34m[RLC]   /home/timarque/panos/openairinterface5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:792:add_drb_am: added drb 1 to UE with
0x70bd
[0m[0m[RLC]   /home/timarque/panos/openairinterface5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:add_drb:879: added DRB to UE with RNTI 0x70bc
[0m[0m[NR_RRC]   [gNB 0] Frame      0 CC 0 : SRB2 is now active
[0m[0m[NR_RRC]   [gNB 0] Frame  0 : Logical Channel UL-DCCH, Received NR_RRCReconfigurationComplete from UE rnti 70bd, reconfiguring DRB 1
```

Figure 23: DRB establishment at the gNB upon reception of RRC Reconfiguration Complete message

After these steps the UE can exchange IP traffic through the CN. Finally, a ping test initiated from the CN towards the UE is shown in the figure below.

```
bourdon@bourdon:~$ ping 192.198.0.2
PING 192.198.0.2 (192.198.0.2) 56(84) bytes of data.
64 bytes from 192.198.0.2: icmp_seq=1 ttl=64 time=26.7 ms
64 bytes from 192.198.0.2: icmp_seq=2 ttl=64 time=39.9 ms
64 bytes from 192.198.0.2: icmp_seq=3 ttl=64 time=22.8 ms
64 bytes from 192.198.0.2: icmp_seq=4 ttl=64 time=65.9 ms
64 bytes from 192.198.0.2: icmp_seq=5 ttl=64 time=28.9 ms
64 bytes from 192.198.0.2: icmp_seq=6 ttl=64 time=22.8 ms
64 bytes from 192.198.0.2: icmp_seq=7 ttl=64 time=65.9 ms
64 bytes from 192.198.0.2: icmp_seq=8 ttl=64 time=23.9 ms
64 bytes from 192.198.0.2: icmp_seq=9 ttl=64 time=32.9 ms
64 bytes from 192.198.0.2: icmp_seq=10 ttl=64 time=35.8 ms
64 bytes from 192.198.0.2: icmp_seq=11 ttl=64 time=23.9 ms
64 bytes from 192.198.0.2: icmp_seq=12 ttl=64 time=21.9 ms
64 bytes from 192.198.0.2: icmp_seq=13 ttl=64 time=49.8 ms
64 bytes from 192.198.0.2: icmp_seq=14 ttl=64 time=22.9 ms
```

Figure 24: Ping test for user plane traffic with the OAI SA setup

3.5.6 Component: 5G Core

Cumucore has developed dynamic network features to meet the 5G-RECORD project requirements. This includes the capability to connect local data network directly to UPF, which enables local audio mixing within UC1. In addition, the 5G network virtual functionalities have been further developed to handle Network Slice lifecycle functionality. Cumucore has also developed PCF functionality to be able to dynamically create data flows in 5G network and provides Cumucore Network Wizard to add and remove data flows using external API.

Cumucore 5G SA network

3GPP Release 15 already includes the specifications on the 5G Core (5GC) Network, in TS 23.501 [4]. The 5GC follows a number of principles that are mainly targeted for reaching higher flexibility, supporting many different use cases. This includes the introduction of service-based principles, where network functions provide services to each other. A clean control plane/user plane split allows independent scaling of control plane and user plane functions, and supports flexible deployments in terms of where the user plane can run. (This principle was, in fact, already introduced in EPC in Release 14). The architecture allows for different network configurations in different network slices.

The 5GC control plane is based on the Service Based Architecture (SBA). In SBA, the network functions communicate with each other via a logical communication bus and network functions can provide services to each other. A network function instance is registered to a Network Repository Function (NRF). Using the NRF, a network function instance can find other network function instances providing a certain service. The goal of such an architecture is to provide more flexibility in the overall system, and to make it easier to introduce new services.

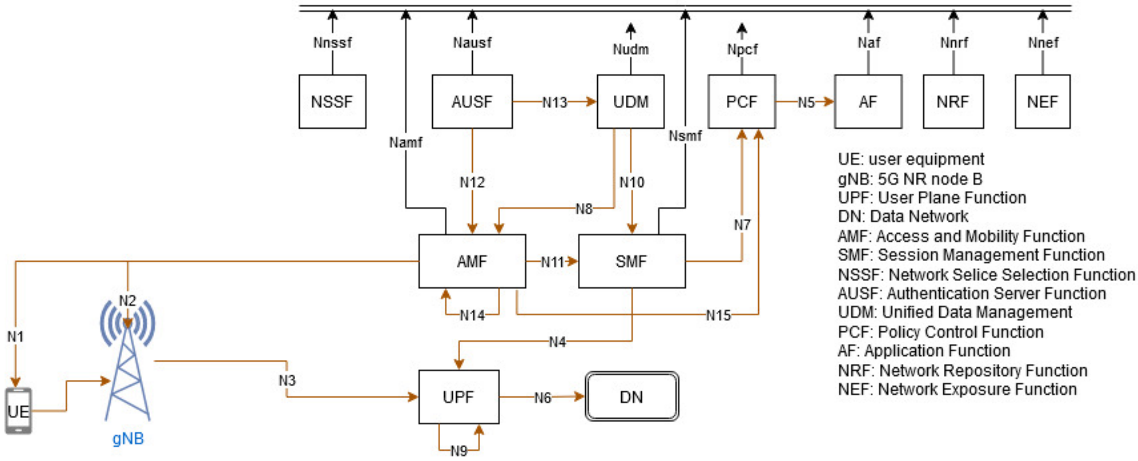


Figure 25: 5G Core architecture overview

The AF in 5G-RECORDS UC1 would consist of specific media related applications that have to interact with 5GC to request network slice with specific requirements for traffic. Moreover, additional AFs from other media services can request MEC capabilities to run specific audio/video processing.

Cumucore Network Wizard

Cumucore also provides Cumucore Network Wizard that has a capability to add and remove data flows using external API. This functionality is also used in the UC2, so further details can be found in 3.5.9.

3.5.7 Component: Shared Access Server

RED Technologies provides the Shared Access Server for UC1; its main features are presented in this section.

SAS server architecture

The SAS server is responsible for determining the maximum allowed transmission power of each device. Figure 26 shows the architecture of the SAS server.

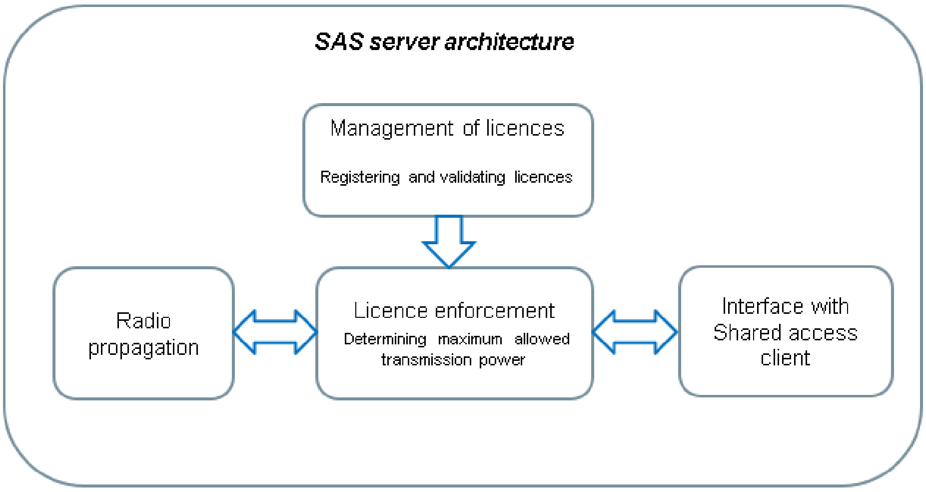


Figure 26: SAS Server Architecture

Management of licenses

A **licence** is associated with: *(i)* a licence identifier, *(ii)* an administrative area identifier, *(iii)* a frequency range, *(iv)* an expiration date, *(v)* a licensee identity, and *(vi)* an interference protection threshold (in dBm/MHz).

A **lease** is associated with: *(i)* a lease identifier, *(ii)* a lease area, *(iii)* a frequency range, *(iv)* an expiration date, *(v)* a lessee identity, *(vi)* the identifier of the licence associated with this lease, and *(vii)* an interference protection threshold (in dBm/MHz).
The owner of a lease or a licence is allowed to create protection zones within the area associated with this lease or licence.

Based on the planned deployment, desired transmission power, and radio environment the SAS determines the protection zone contour and identifies the most suitable frequencies.

License and lease enforcement / protection from interference

For each protection zone (associated with a license or a lease), the SAS server ensures that the aggregation of emissions from all devices (e.g., all 5G small cells) located outside this zone is below the interference threshold associated with this licence. The level of interference at given point within the protection zone is determined as follows:

1. Computing interference of each device located outside of this protection zone.
    a. Computing the path loss between each device and this given point (considering height, azimuth, beamwidth).
    b. Adding the transmission power currently allowed for this device.
2. Aggregating all interferences computed in 1).

PMSE setup operation

The overall flow to allow a PMSE to operate is shown below.

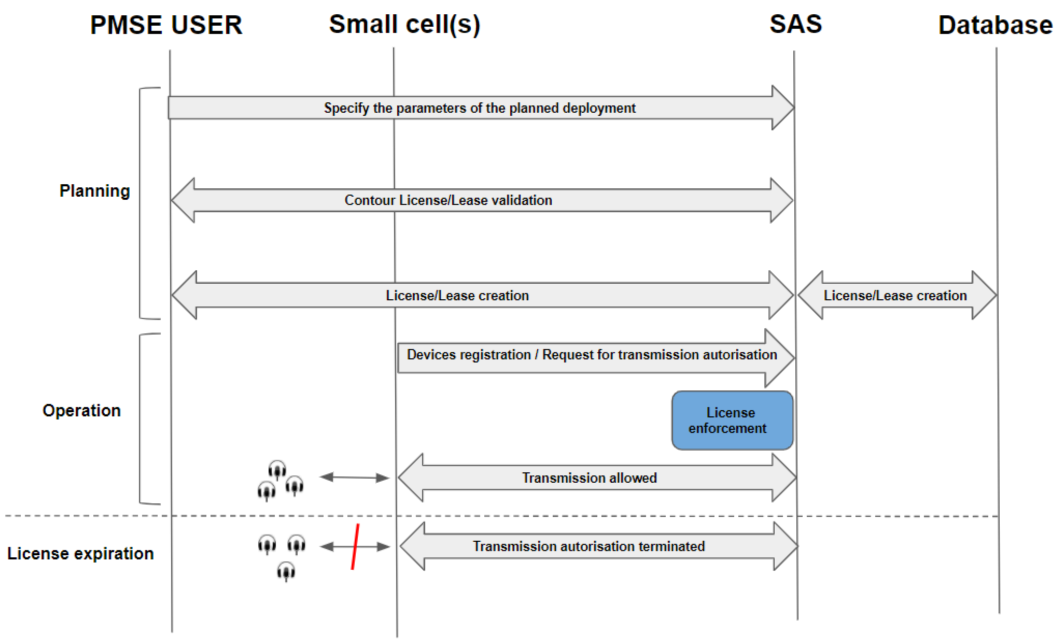


Figure 27: PMSE setup operation

3.5.8 Component: Time Service

Media networks use time synchronisation to synchronise media clocks to avoid buffer over- or underruns and to maintain best audio quality. In IP-based media networks time synchronisation is achieved by deploying PTP clients and server [5]. In a wired scenario this is typically achieved by providing a wired wordclock signal either via BNC or via PTP over Ethernet/IP.

During the first integration phase the PTP time service (the PTP server) will be a distinct service that bypasses the 5GS. In further steps the time information should be provided through the 5GS, and the time service could also be a function in the 5GS.

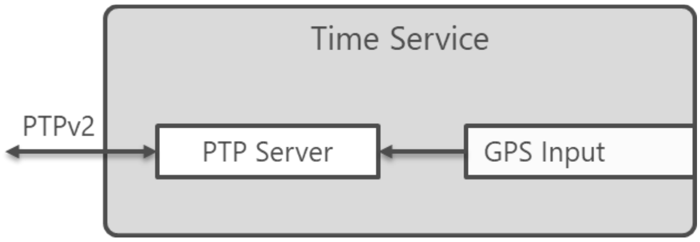


Figure 28. Block diagram of time service component

3.5.9 Component: Network Slice Management

Cumucore Network Wizard manages Network Slices and their usage at the data flow level. There is a user interface that can be used to create, operate, and terminate Network Slices. In the 5G-RECORDS project, APIs have been developed which are used to generate dataflows with specific QoS settings and related SLAs.

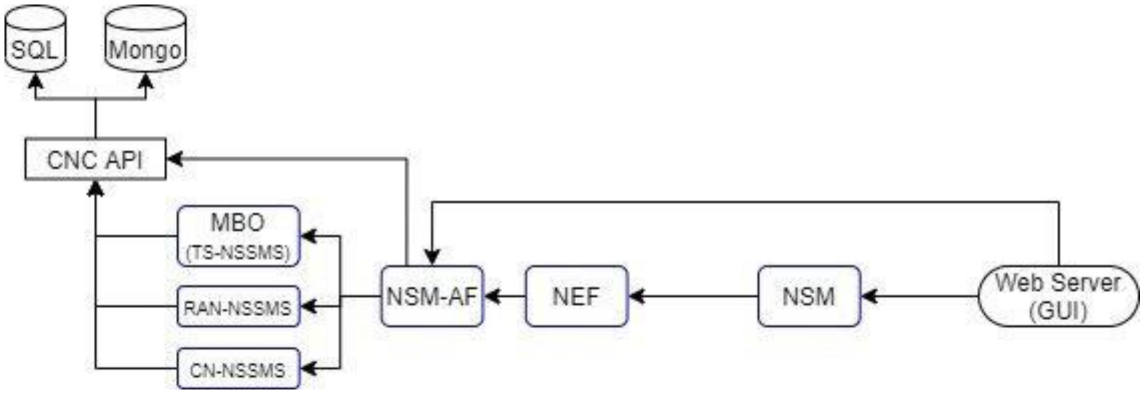


Figure 29: Block Diagram of Network Slice Manager

Network Slice Management is done through an Application Function which can be called NSM-AF, and in 3GPP specifications is called as a Network Slice Management Service (NSMS) or Network Slice Subnet Management Service (NSSMS) that supports different use cases defined in TS 28.531. A slice subnet is considered a different segment of the end-to-end system e.g., RAN subnet, Transport subnet, Core subnet.

The NSMS/NSM-AF can be an internal function only accessible from other network functions or from internal management console (i.e., GUI). The NSMS/NSM-AF can be made accessible from external applications through Network Exposure function (NEF).

NSMS/NSM-AF provides an interface to create, activate, terminate slices in the network and perform feasibility checks before creating new slices. The NSMS/NSM-AF will interact with different modules to create, delete, modify the network slices.

NSMS/NSM-AF will interact with the so-called Transport Network (TN) manager, which is equivalent to the Mobile Backhaul Orchestrator (MBO) to reconfigure the Transport Network to support new network slices. The NSMS/NSM-AF sends the transport network related requirements (e.g., external connection point, latency and bandwidth) to the TN Manager which reconfigures the TN accordingly (TS 28.531: section 5.1.1)

NSMS will interact with the other network or radio orchestrators to allocate the resources on different subnets for creating the new slice. NSMS/NSM-AF will interact with Radio Resource Manager (RRM) to configure radio cells for the slice. NSMS/NSM-AF will interact with Network Function Management Service or Network Function Virtualisation Orchestrator (NFVO) to allocate network functions required for the slice.

The NSMS/NSM-AF will also perform feasibility checks to confirm resources available for network slices through the NWDAF and other subnet management modules like MBO or RRM.

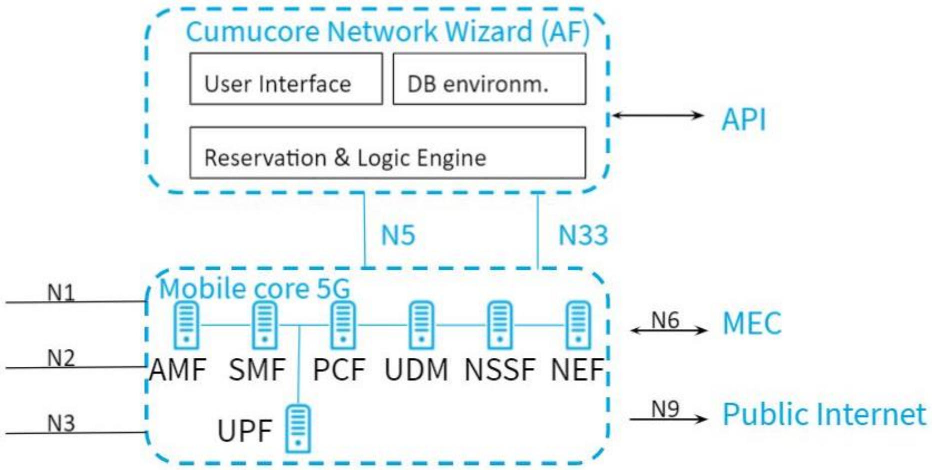


Figure 30: Interfaces and Interactions among modules

3.6 Interfaces

Figure 31 depicts the UC1 interfaces within the overall E2E infrastructure.

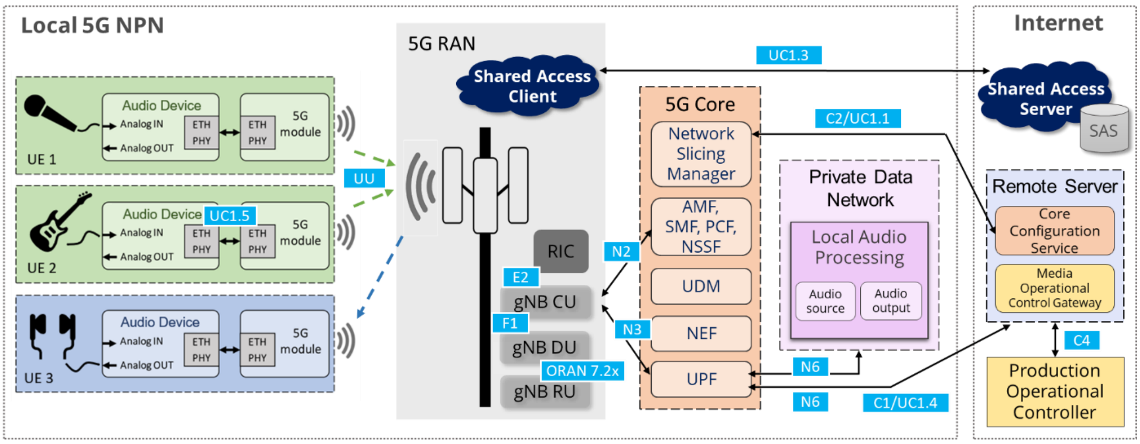


Figure 31: UC1 Interfaces

3.6.1 Analogue Input

Analog audio input to connect a microphone or line signal.

3.6.2 Analogue Output

Analog audio output to connect a headphone or a PA system (Public Address system).

3.6.3 GPS Input

The GPS (Global Positioning System) input is used as a reference for providing the PTP service.

3.6.4 [UC1.1] Core Configuration Protocol

REST API to request application specific data flows for media devices.

3.6.5 [UC1.4] Audio Device Control

A proprietary TCP-based protocol to control audio devices. It is an outbound connection initiated by the audio devices towards a remote server. The protocol allows control of audio settings as well as installation of new software.

3.6.6 [UC1.5] Audio Network Device to UE Interface

Physical: 1 Gigabit Ethernet.

3.6.7 [UC1.9] Network Audio Protocol

For network audio a custom IP/UDP-based audio transmission protocol is used. It is similar to RTP (Real Time Protocol) and contains timestamps and sample indexes alongside the audio data. In case of time synchronisation between source and sink this allows for calculation of an absolute one-way transmission latency.

3.6.8 Near-RT RIC

In the context of the 5G-RECORDS UC1, the Near-RT RIC will be managed and configured via the dRAX dashboard which embeds autonomous SMO functionality, i.e., no external SMO will be required. The near-RT RIC controls the CU CP and CU UP components via Netconf/RESTful APIs and an Accelleran dRAX data bus based on NATS and Kafka technologies.

3.6.9 [N2, N3, F1] CU

The CU CP interfaces northbound with the 5G Core AMF via the 3GPP-standardised NG-AP (N2) interface based on SCTP/IP protocols using Ethernet Datalink/Physical layers. The CU CP interfaces southbound with the DU via the 3GPP-standardised F1-C (F1AP) interface based on SCTP/IP protocols using an Ethernet-based Datalink/Physical layer.

The CU UP interfaces northbound with the 5G Core UPF via the 3GPP-standardised NG-U (N3) interface based on GTP-U/UDP/IP protocols using an Ethernet-based Datalink/Physical layer. The CU UP interfaces southbound with the DU via the 3GPP-standardised F1-U interface based on GTP-U/UDP/IP protocols using an Ethernet based Datalink/Physical layer.

The CU CP and CU UP interface with each other via the 3GPP-standardised E1 (E1AP) interface based on SCTP/IP protocols using Ethernet Datalink/Physical layers.

Spectrum Sharing Interfaces

Figure 32 presents the specific interfaces within the spectrum sharing component.

3.6.1

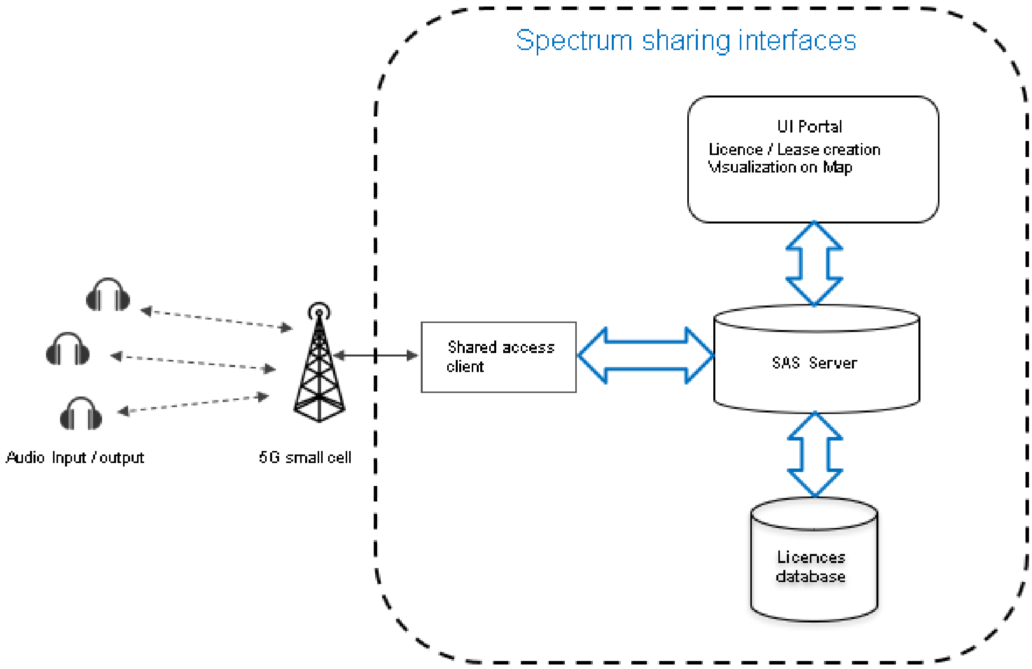


Figure 32: Spectrum Sharing Interfaces

3.6.10.1

[UC1.3] SAS Server – Shared access client

The SAS client interfaces with the SAS server using a JSON-based Winnforum protocol over a secure connection using TLS v1.2.

1. Device registration (e.g., 5G Small Cell): Reception of parameters from deployed devices required for interference protection including localisation and antenna parameters.
2. Handling of requests from a device to access spectrum: Reception of the frequency range and power over which to operate.
3. Real-time control of spectrum grants: The SAS can receive heartbeats from the devices at regular intervals.

3.6.10.2 This enables to move the station to another frequency when a more suitable frequency is identified and to update the transmission power as needed (e.g., to provide a higher maximum allowed power)

Database of licenses

The SAS server is connected to a database containing the list of active licenses and leases.

UI portal

A UI portal enables users to connect to the SAS server from a web browser to: *(i)* monitor spectrum sharing activity on a map, and visualise the following elements: licenses, leases, protection zones, devices deployment, and active spectrum grants; *(ii)* create protection zones within licenses and leases

3.6.10.3 [N6] 5G Core to Data Network Interface

Physical: 1 Gigabit Ethernet.

Routing traffic between this device and the Internet gateway must be possible.

3.6.11

4 UC2 Components - Multiple Camera Wireless Studio

4.1 General Architecture

The Multiple Camera Wireless Studio use case is based around a multi-camera audio and video production in a professional environment.

This use case will aim to the integration of a media production system, with up to 5 wireless cameras that will replicate existing technologies such as COFDM radio cameras in terms of performance and capabilities using 5G. Multiple location scenarios with production facilities local to an event as well as remote and distributed production models will be explored.

Existing media production solutions that use dedicated radio spectrum to transmit video are very mature but require bolt-on links for ancillary services (such as control, audio or reverse video). Meanwhile, there are several wired IP-based technologies using compressed and uncompressed media to support the production workflow. The SMPTE ST 2110 suite [8] standardises how to carry uncompressed video and audio over RTP (Real-time Transport Protocol). However, these solutions target fixed installations. Wireless solutions will need to use compressed video, as uncompressed video demands more bandwith than what wireless can handle, and it is therefore necessary to include encoding and decoding into the architecture.

Therefore, we wanted to build a simple architecture that would provide interoperability and seamless integration between these existing solutions and protocols.

In additional scenarios we expect to integrate 5G based contribution solutions using different types of network configuration to provide contribution links into production centres.

A basic overview of the use case architecture is presented in Figure 33.

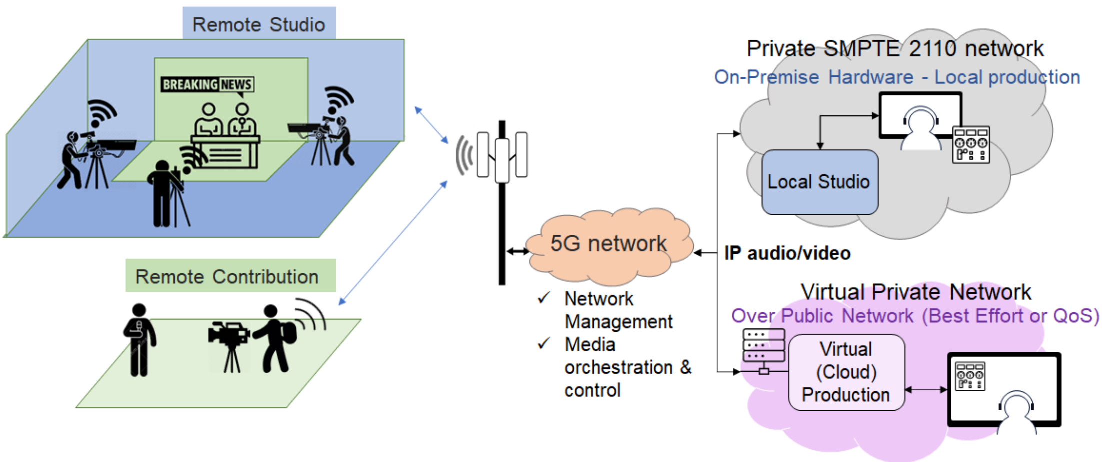


Figure 33: UC2 Basic architecture.

4.2 Use Cases under analysis

We set out to explore a scenario in which a production crew could easily establish a production set-up with multiple wireless cameras in just a few steps:

- The engineering manager sets up, rigs and configures a 5G radio and core network, and connects to the media production
- The camera crew mounts a wireless enabled camera
- When the wireless camera is put into position with other wired cameras and switched on, the 5G S-NPN network discovers the unit (as it is self-provisioning), and the camera is configured.

The production setup is now ready and with a full IP connection between cameras and their ST 2110 production site. Before and during production, services can be assigned, and the camera interface unit remotely controlled. Cameras are synchronised with a centralised PTP (Precision Time Protocol), thereby it is possible to cut seamlessly between 5G cameras and wired cameras. Assuming spectrum and bandwidth are available, scaling production by adding more 5G cameras is easy enough: 5G S-NPN are by nature highly scalable, as it is based on commodity hardware and its core functionality is software-based.

4.2.1 Wireless cameras within a production

The core aim for this scenario is to explore the substitution of current wireless RF cameras with 5G.

4.2.2 Integration of cloud-based distributed production

A MCR (Master Control Room) will be deployed to monitor and manage the incoming and outgoing feeds.

4.2.3 Remote contribution

Remote contribution consists of receiving the contribution feeds from the remote locations: a video encoder-transmitter or camera-mounted device is used to encode and broadcast video without the need for mobile units (vans) and/or satellite or microwave links.

4.3 Data Flows

This section presents the data flows present in UC2 systems.

4.3.1 Control Flows

For UC2 systems control flows, see Section 6.3 (Media Operational Control Gateway).

4.3.2 5G end-to-end user plane protocol stack

Below, Figure 34 depicts the 5G data flows in the UC2 systems.

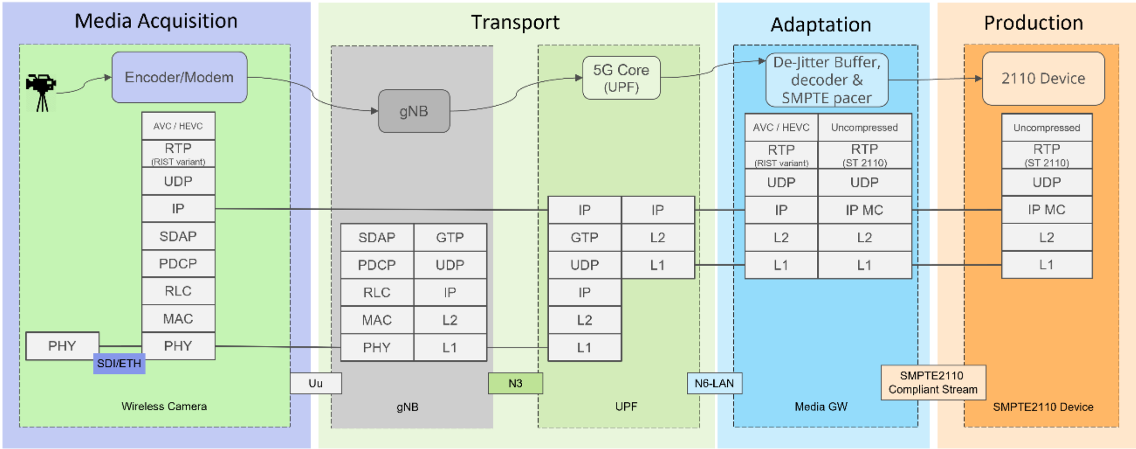


Figure 34: UC2 end-to-end user plane protocol stack.

4.4 Delay budget

The use case defines different latency requirements per user story (Table 15). The wireless studio scenario requires lower latency than the production or contribution scenario. For more information about latency requirements please refer to D2.1 [2]. The delay budget per category (D3.1) is reported for convenience also here.

Table 15. Latencies per category.

Category	Property	Description
Encoding and decoding	Video encoding, RTP/IP packaging	The time required by the encoder to compress the raw images, to package into a media transport protocol and make them available for transmission
Transport	5G transport	The time in which packets are transferred from the output of the encoder to the 5G network edge (N6)
Adaptation	HEVC to SMPTE	Latency induced by the media gateway including de-jitter buffer, decoder
Production	ST 2110 network	The latency induced by transferring the ST 2110 stream from the media gateway to the SMPTE ST 2110 receiver

4.5 List of Components

This section lists the components for the use case.

4.5.1 Component: Fivecomm 5G Modem (F5GM)

The Fivecomm 5G modem (5G BROAD) is an advance version of the devices described in previous deliverables. Two units have been exclusively developed for UC2, being integrated in Aachen several times as part of the end-to-end infrastructure, and connecting professional cameras to the 5G network. Fivecomm has accomplished their initial objectives and has developed and integrated this compact and flexible module in the 5G network from Ericsson. Performance measurements have been also reported in other deliverables to validate the component.

This solution, developed for fixed wireless scenarios, connects via Ethernet the video encoder component and connects to the 5G network via its external antennas. The 5G modem has simplified its electronics while minimizing the power consumption and cost. It is versatile enough to fit the different requirements specified by content production professionals in the context of this use case. The 5G modem is formed by to main components in turn, which are the 5G module, and a Raspberry Pi 4 that comes with a Linux distribution and permits to setup and configure the modem in an easy manner. Both components are integrated together and included as part of the final product in a steal or plastic case (depending on the needs of the scenario).

Figure 35 shows the final design of the 5G BROAD that has been used in the test-bed and will be used for the trials.

Figure 35: 5G BROAD – 3D design.

- ***Functionalities:***

An updated version of the different functionalities of the modem is provided below:

Table 16: Fivecomm 5G BROAD functionalities

Functionality	Description
Easy deployment	'Plug and play' fashion. It only needs to connect the cables from the video encoder to the digital connectors, fix the device to the infrastructure and press the 'ON' button.
Customisation	Different IP protection modes are available, with up to 4 internal antennas or external ports to provide the best experience even in low coverage scenarios.
Remote management	It comes with a SW management platform called OpenWRT that allows configuration, monitoring, and updating.

- ***Technical details:***

Table 17: Fivecomm 5G Modem technical specifications

Technical Specifications	Description
5G native mode	Both 5G Non-Standalone (NSA) and 5G Standalone (SA) modes are supported. Option 3x, 3a and 2 network architectures.
5G New Radio (NR) Release	Release-15.
Sub-6 GHz frequency bands	n41/n77/n78/n79/n1/n3/n5/n7/n8/n20/n28/n38/n40
Antennas	Up to 6 antennas (external or internal) to support both mid- and low-bands.
SIMs	Dual SIM.
Ports	1 Ethernet port.
DL&UL	Up to 2.5 Gbps in the DL and 900 Mbps in the UL.

Note that partners involved in the wireless scenario use case aim at going one more step beyond. The final objective with the 5G modem is to design and develop a portable solution that can be integrated into wireless professional cameras. Fivecomm is currently helping with the design of this solution, which includes not only the 5G BROAD, but also the coding/decoding components and a series of features that are described below

4.5.2 Component: Image Matters A/V Encoder – contingency plan

IM was in charge to deliver what we have called "smartboard" in the deliverables with the following capabilities:

- Encode and decode multiple video format resolutions (e.g., high definition, ultra-high definition) at different bit-rates using low latency settings
- Encode and decode audio
- Generate a genlock signal for the camera from the timing source signal coming over the 5G network (e.g. PTP)
- Timestamp the RTP streams using the same timing source signal
- Produce RTP streams
- Incorporate NMOS (Networked Media Open Specifications) to work with the MOCG
- Provide the right physical interfaces to be connected to the camera for remote control
- Integrate the 5G S-NPN modem from Fivecomm

A portable component, integrating all these features together, with the right form-factor to be plugged on the back of a broadcast camera does not exist in the market.

To overcome the setback due to IM bankruptcy, the partners involved in UC2 have elaborated a contingency plan aiming at the testing the different features even if not integrated all together:

- "Portable" feature
    - One of the goals is to test multiple portable 5G-enabled cameras, at least two. Ericsson and Fivecomm have agreed to deploy an integrated portable solution (encoder and modem) to be piggybacked on the camera
- "NMOS" control plane integration

  - BBC and Bisect are developing a MQTT based communication between camera and MOCG for discovery, connection and configuration of cameras on the network (jetson nano)
  - RAI is developing a PoC to control remotely the camera via the Cyanview interface.
- “Genlock” feature derived from the timing signal
  - Unlikely to be developed by any of the UC2 partners even if the partners are still investigating how to address it
- PTP (Precision Time Protocol) feature

Ericsson has developed an Ultra-reliable low-latency communication (URLLC) test network to focus on 3GPP Release 16 radio functions for time-critical communication. The network includes features primarily from Release 16 but also from Release 17, in particular, the support of SMPTE Profile for the use of IEEE Std 1588 Precision Time Protocol in Professional Broadcast Applications ST 2059-2:2015. This testbed will be used to measure the performances of the PTP over 5G and to compare them with the results measured using Rel.15 testbed.

Existing commercial codecs produce RTP streams TS (Transport stream) based: BISECT is adding the support of RTP TS based in the media gateway to be interoperable with these solutions that could be used as fallback solution if the portable integration fails.

With respect to the mobile camera, Ericsson started to develop a mobile encoder prototype with a SDI Input and 5G Modem. The main platform of the prototype is a NVIDIA Xavier Developer Kit which provides the necessary I/O ports and power to the components while maintaining small form factor and weight. For the SDI Input of any source, the prototype uses a Magewell ProCapture 4K PCIe Card which is mounted on the PCIe-Slot of the platform and fixated on a prototype case.

To ensure a stable 5G connection with a Quectel RM500Q-GL, the modem PCB is mounted below the Xavier and connected via USB, providing enough space for the antennas and the battery in the prototype. The prototype should have two possibilities to be powered. The first option is the utilisation of the cameras battery pack with an adapter cable to the Xavier platform. In the second option a 12V battery is powering the Xavier Unit.

Figure 36: Portable setup.

To operate the mobile encoder complete remotely, the system automatically announces itself via NMOS to the MOCG.

Fivecomm is doing the same, but using the Fivecomm modem.

Component: Media Gateway

The Multiple Camera Wireless Studio use case introduces several challenges in terms of media formats, timing and transmission modes:

4.5.3
- Unlike traditional wired production studios, there is limited radio spectrum available within wireless networks, so it is not practical to use uncompressed video. This means that compression is required, which can lead to increased latency, lower visual quality, or a combination of the two.
- Wired cameras are often genlocked, which may not be possible on a wireless network.
- Wired media networks often use multicast communications, which may also not be possible on a wireless network, more likely to support only unicast transmission.

The main role of the Media Gateway (MG) is therefore to adapt the media signals and to integrate the 5G network with the media production network, both when the latter is a SMPTE ST 2110-based production studio, or a Cloud/based one.

- **Functionality**

Arguably, the most important responsibility of the MG is to adapt the streams between different media formats. For example, the MG can receive HEVC over RTP and convert it to uncompressed video over ST 2110. Conversely, the MG can receive the return video from the production studio and convert it to HEVC/RTP, to deliver it to the venue.

Additionally, the MG performs address and unicast/multicast conversion, under the control of the MOC Gateway. When multicast is not supported on the 5G network, the MG can create multiple unicast replicas of the same multicast stream. Similarly, it can receive a multicast stream from the production network and generate one or more unicast streams that are delivered to individual components of the 5G network.

The MG can receive or send RIST (Reliable Internet Stream Transport) or SRT (Secure Reliable Transport), which can be used, not only as the transport from the cameras to the studio and back, but also to forward the streams to a virtual production studio on a Cloud. RIST and SRT offer functionality for sending data over a non-QoS provisioned wide area network.

- **Control and monitoring interfaces**

The main way to interact with the MG is through its REST interface. This interface provides all the facilities required to configure and control the different aspects of the MG. In addition to this, the MG also provides a WebSocket interface to report asynchronous or periodic events, such as status changes.

Finally, the MG also provides a browser-based graphical user interface (GUI), which allows the user to verify the presence of a signal, including statistics such as bit rate or packets per second, and includes a live stream confidence monitoring window for each stream. Figure 37 and Figure 38 below show the monitoring GUI when a stream is being received, as well as when it is not active.

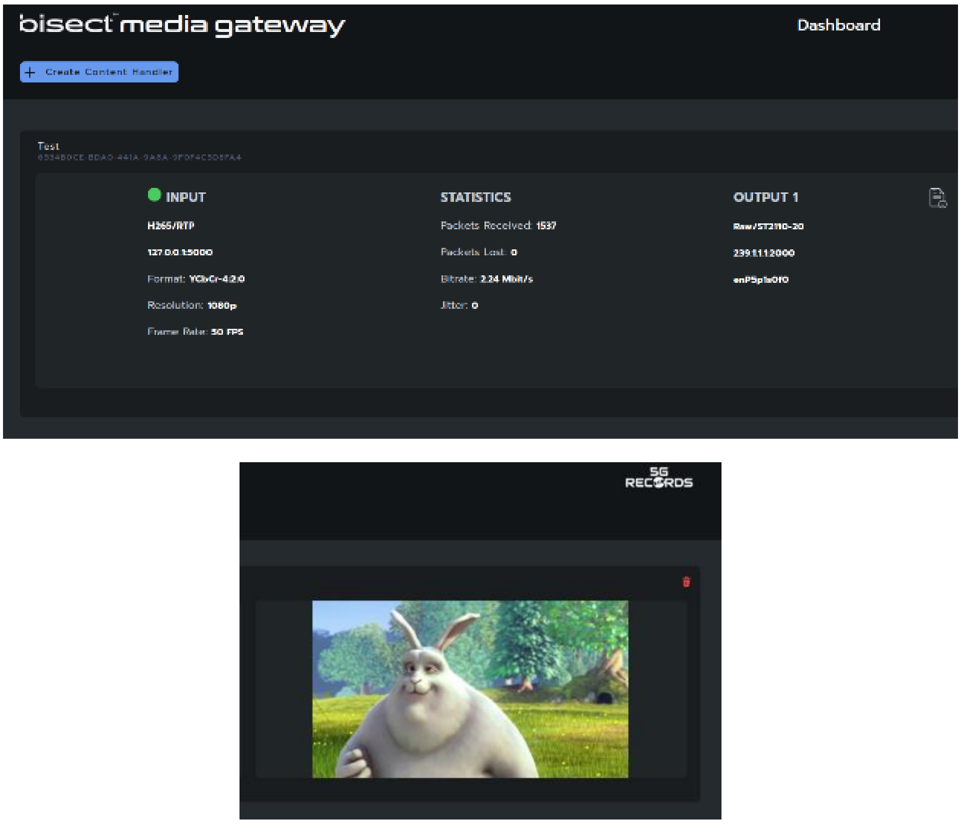


Figure 37: Media Gateway GUI showing an active stream status, statistics and live video monitor.

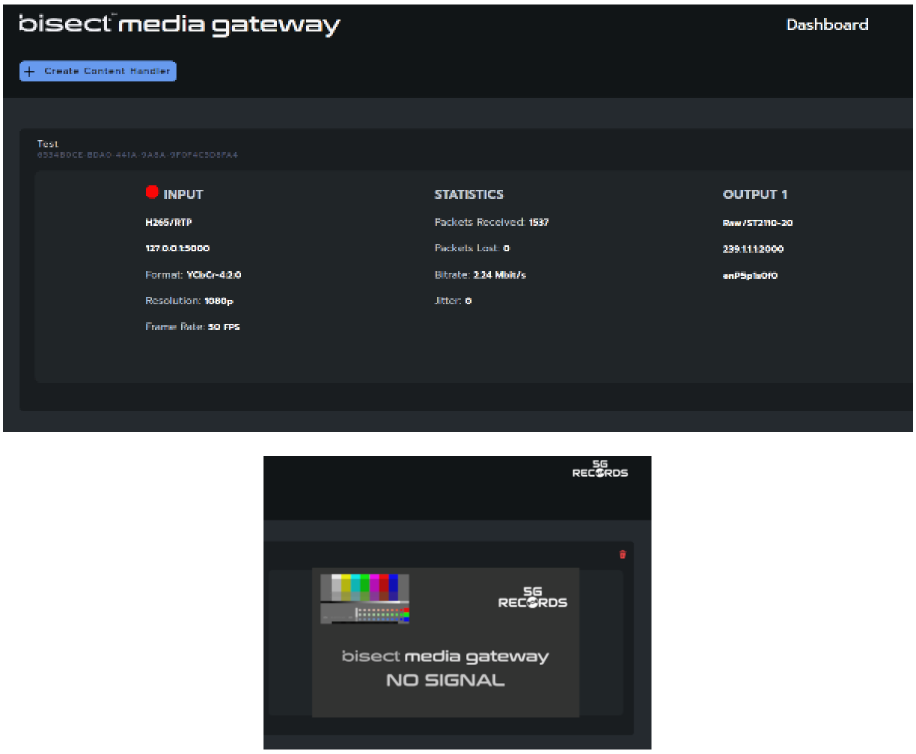


Figure 38: Media Gateway GUI showing no input status and fallback signal.

- **Architecture**

Figure below shows the MG architecture.

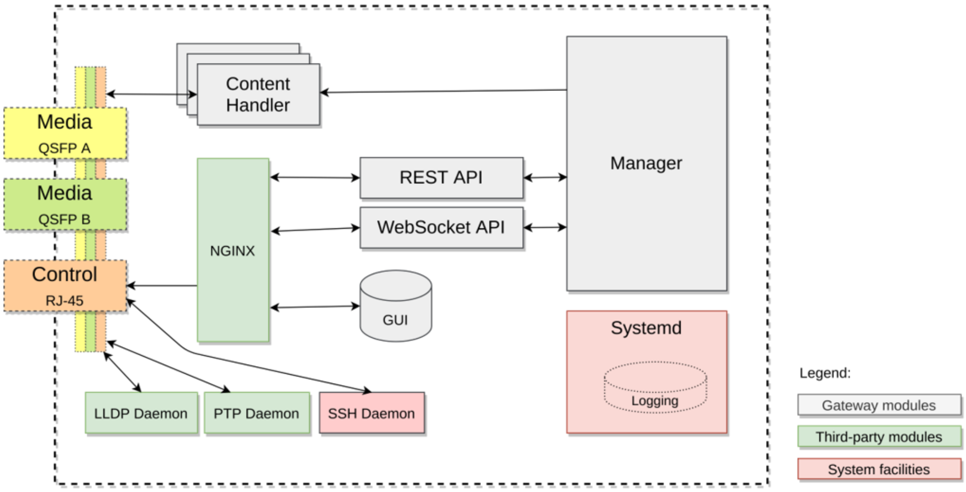


Figure 39: Media Gateway architecture.

The architecture of the MG can be coarsely divided into three main groups: infrastructure, business logic and media processing.

While the infrastructure modules provide fundamental facilities, such as PTP synchronisation, as well as basic services like the HTTP server for the GUI and the APIs, the business logic coordinates all aspects of the operation of the MG.

Last but not the least, the media processing aspect is provided by a module called Content Handler (CH). A new CH is instantiated by the business logic when that is requested by the controlling system, such as the MOCG.

Each CH operates independently from each other and the number of CH can grow infinitely, provided there are enough resources available.

This architecture allows the MG to be deployed as a stand-alone appliance, capable of processing a handful of streams simultaneously, or as an elastic deployment, over several physical nodes, with almost an infinite capacity.

- **Formats**

The MG supports the following formats:

- Video: HEVC, AVC, Uncompressed.
- Audio: Opus, PCM.
- Transport: Raw RTP, ST 2110, SRT, RIST

- **NMOS integration**

The Media Gateway is compatible with AMWA NMOS IS-04 and IS-05. It registers itself as a NMOS Node and automatically creates a new NMOS Device for each Content Handler that is instantiated by the MOCG. The CH creates a NMOS Receiver, as well as a set of NMOS Source, Flow and Sender for each output stream. Controlling applications can then use NMOS IS-05 to set up connections between the MG and other NMOS-capable devices.

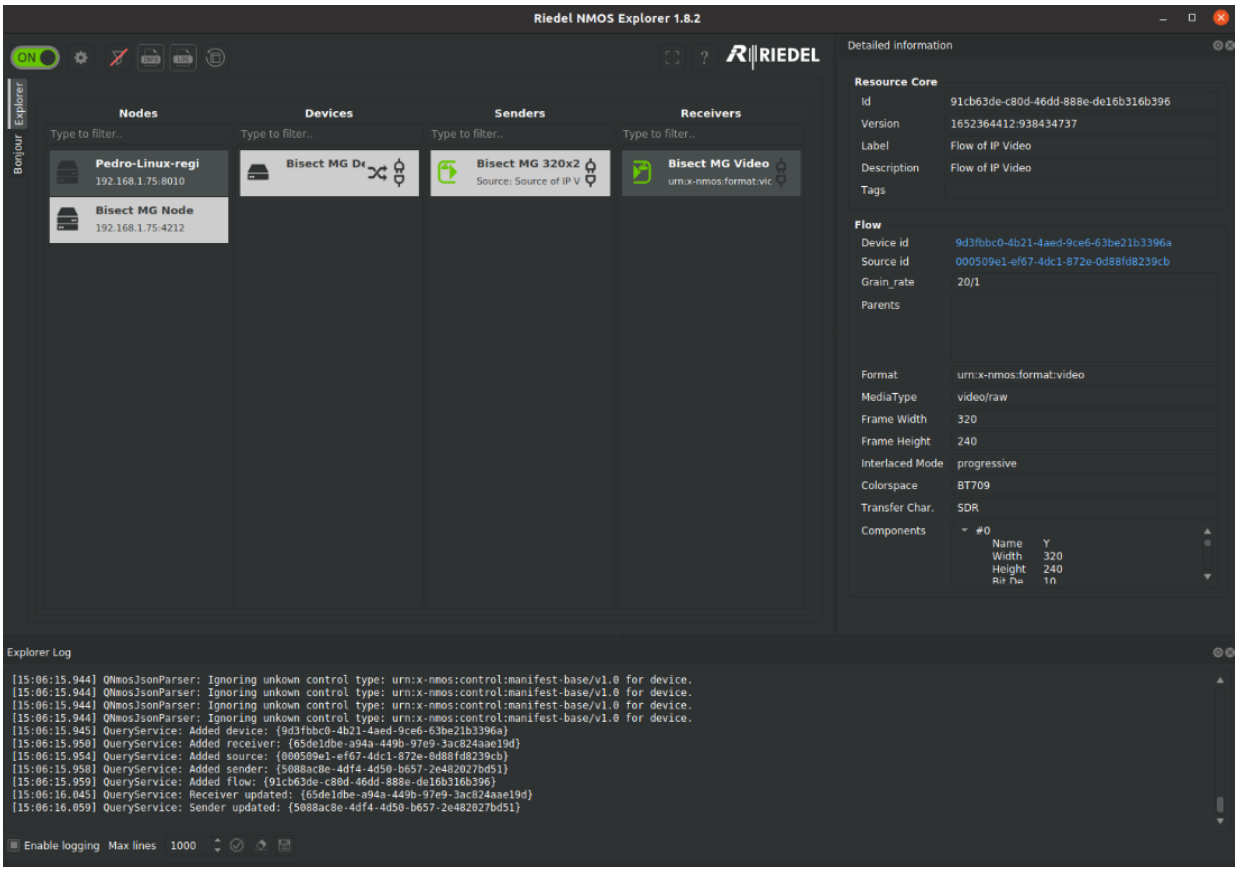


Figure 40: Browsing the Media Gateway NMOS registration using Riedel NMOS Explorer.

- **Deployment**

For the project trials, the MG was deployed as a stand-alone appliance, based on a NVIDIA Jetson AGX Xavier, equipped with a NVIDIA ConnectX-6 Dx Network Interface Card.

It is also possible to deploy the MG on a fully virtualised environment, using Docker containers and Kubernetes orchestration. This kind of deployment requires that the cloud provides the required GPU (Graphical Processing Unit) and network acceleration hardware.

4.5.4 Component: 5G networks

5G-NR (New Radio) is the latest standard for telecommunication networks. It was first introduced in 3GPP Release-15 in December 2018.

Ericsson provided different 5G Systems. A detailed description of the provided 3GPP Release 15 System is provided in Deliverable D3.1, Section 4.5.6 and Annex A.

For studying the precise time synchronisation capability of the 5G System, Ericsson provided a testbed which supports 3GPP Release 16 and 17 functions for precise time synchronisation using PTP.

The radio and core network are depicted in Figure 41. The prototype UE is also as rack shape and depicted in Figure 42.

Figure 41: 5G Core Network and Radio Network.

Figure 42: Experimental UE.

The testbed operates within Frequency Range 2, i.e., at 28GHz (mmWave) and with a 200MHz bandwidth. It is configured with a 1:1 TDD pattern, which means the scheduler can split radio resources equally between the uplink and the downlink. The subcarrier spacing is set to 120KHz, resulting in a time slot duration of 0.125ms. This configuration will result in low latency because the UE gets a slot for transmission every short period of time. The testbed network can provide a consistent RTT of ~2ms.

For the time synchronisation measurements, the testbed was configured to act as an end-to-end Transparent Clock using IP based PTP transmissions. Both, IP unicast and IP Multicast transmissions of PTP messages are supported.

4.5.5 Component: Master Control Room (MCR) in the cloud

The public cloud provides a vast resource of compute, storage and connectivity, all waiting to be called upon. As live events are by definition time-bound (and usually of significant importance), the public cloud becomes a very interesting place to host the necessary connectivity and MCR services.

It was decided to use an independent software vendor’s solution already available on the market, integrate it into the use case’s setup and test its interoperability with all applicable components of the use case.

Grass Valley Agile Media Processing Platform (GV AMPP) was selected as a proper candidate for the function of the virtual MCR (V-MCR). It may be hosted on any of the three major public cloud providers’ platforms: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

It also supports on-prem / private cloud deployments.

GV AMPP realises its functions in form of applications, each performing some specific tasks. The main functions are *(i)* Master Control Switching, *(ii)* Multiviewer, *(iii)* Recording, *(iv)* Audio Mixer, *(v)* Replay, *(vi)* Playout, and *(vii)* Graphics. We will be concentrating on the first two functions (switching, multiviewer) for the trials.

GV AMPP supports a number of codecs and transport protocols. Here is a list of them:

- Video CODECs:
    - AVC / H.264: 4:2:0 or 4:2:2, 8 and 10 bit
    - HEVC / H.265: 4:2:0 8 bit
- Audio CODECs:
    - AAC
    - Opus
- Transport Protocols
    - SRT
    - NDI
    - RIST Basic and Main Profile
    - RTMP(S)
    - SMPTE ST 2022-2 / MPEG-TS

We intend to use SRT for the “ground-to-cloud” video streams, as well as for distributing streams as the output from Virtual MCR to the broadcasters’ facilities. The output might also be using AWS Elemental MediaConnect service.

The overall diagram of the proposed UC2 topology is presented on Figure 43.

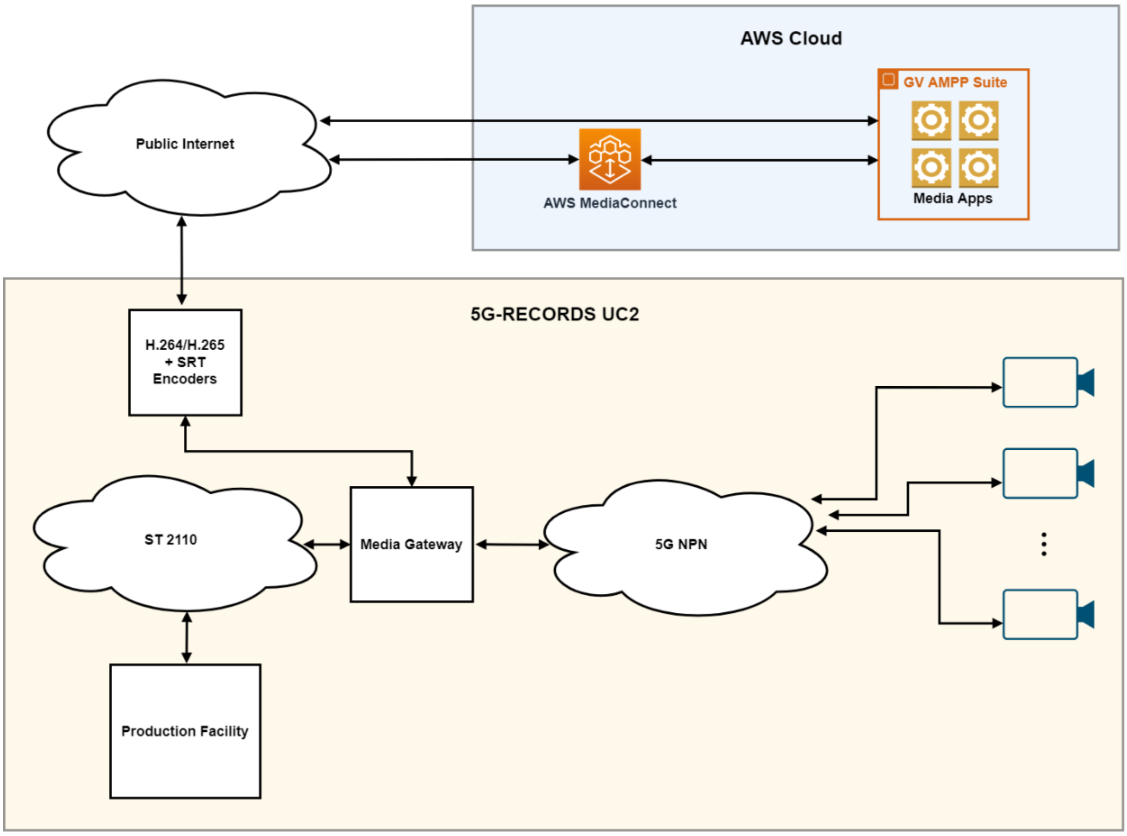


Figure 43: Virtual MCR within the UC2 topology.

For the trials planned in the project GV AMPP is deployed on GD4N.8XL AWS EC2 instance in the AWS account of EBU. For more resource-intensive workloads and when more functionality of the product is used, more powerful instances might be preferred such as GD4N.12XL or newer G5.8XL instances (or better), however G5 instances are not yet available in all AWS regions.

An example of GV AMPP switcher user interface is shown on Figure 44 below.

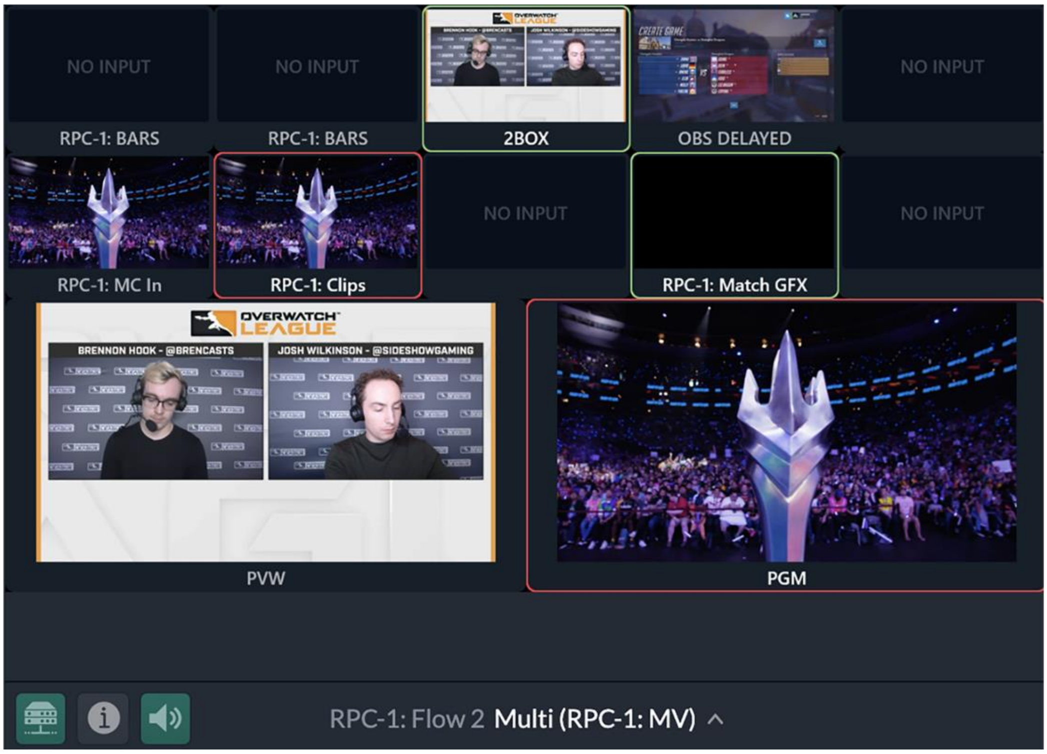


Figure 44: GV AMPP User Interface Example.

4.5.6 Component: LiveU LU800Pro

The LU800Pro [6] is a portable multi-camera all-in-one production-level field unit. Inside 5G-RECORDS project, it is used to encode and transmit video flows captured from cameras to 5G network. This unit has embedded an HEVC H.265 real time video encoder and 5G/4G cellular modems. This specific unit has connectors for external antennas, which came useful for the Ericsson 5G lab in Aachen as the lab policies mandated that the unit will be cabled into the network rather than transmitting on-air.

The LU800Pro can ingest up to 4 A/V streams, encoded them and transmit them simultaneously. It has up to 4 internal 5G modems embedded inside and can connect to two additional/other external modems via the ETH RJ45 connection, which came handy during the tests once the embedded Sierra Wireless modems failed to connect (solution validated in cycle 3 with SW & FW upgrades as well as the network upgrade). Headphones (speaker + mic) can be connected to the unit for audio features such as IFB and intercom, allowing the remote producer/director to be in touch with the field team cameraman over the cellular connection. The LU800Pro also support LiveU IP-PIPE, which is a bi-directional connectivity in parallel to the transmitted video for any type of IP communication. An IP device, or hub, can be connected to it via the ETH RJ45 connector. These capabilities were also validated.

Figure 45: LU800Pro connectors panels.

The LU800Pro encoder is H.265 HEVC. It is controlled by the LU800Pro SW to adaptively change its output according to the available bandwidth. The LU800Pro can then split the encoder output over multiple modems. In some cycle 3 tests in 5G-RECORDS this bonding capability was validated when using two modems operating on a single 5G SA lab network slice as well as when each of the modems operated on a different slice (UL-oriented slice and eMBB slice) as per each of the SIMs allowed registration.

The following picture shows the basic lab setup for this, using external 5G industrial router as the external 5G modem and at the same time cabled into the lab infrastructure for the LU800Pro embedded modems.

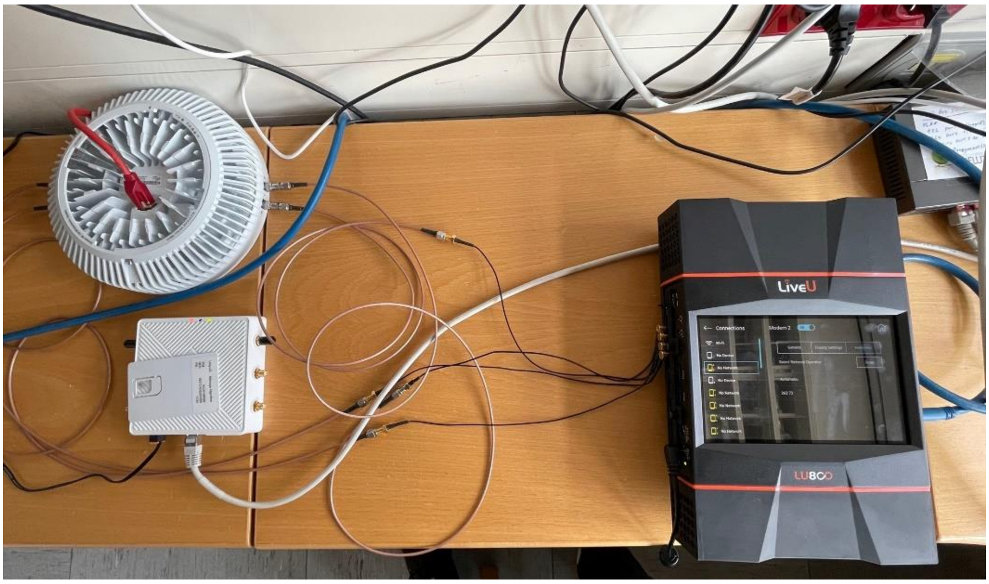


Figure 46: The LU800Pro connected in the Aachen 5G SA lab.

The picture below shows an exemplary screenshot of a transmitting LU800Pro, as seen in the LiveU LU-Central (LUC) cloud-based management system. In this specific case, the transmission is done over a single external modem connected via the LU800Pro Ethernet port.

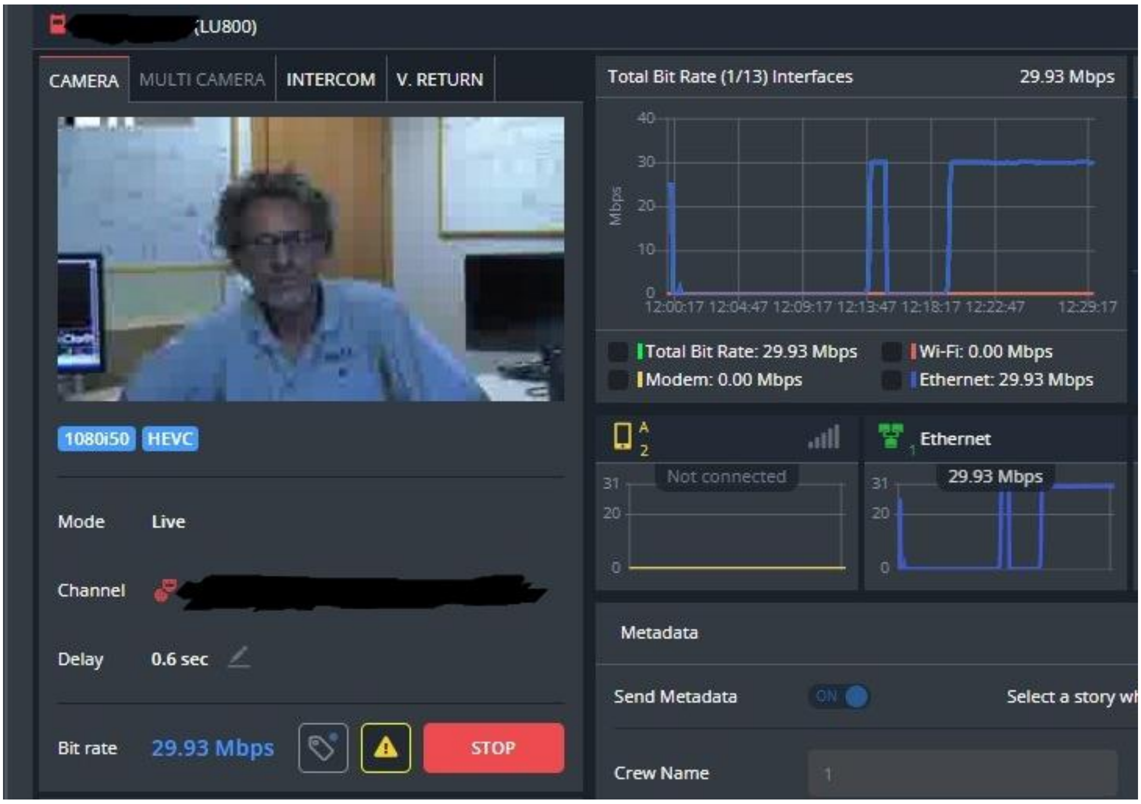


Figure 47: LU800Pro in operation.

4.5.7 Component: LiveU LU2000-SMPTE

LiveU LU2000-SMPTE [7] is a Bonded Video Decoder that receives the video from the LiveU field encoders-transmitters and outputs it over compliant SMPTE A/V out. The LU2000SMPTE received the IP packets containing the encoded-transmitted A/V packets from the LU800Pro over the cellular and then the public network, decodes them, outputs to the Eth using an integrated Nvidia RiverMax SMPTE board, synchs to the studio master PTP clock for this, connects to the LiveU audio server for the audio capabilities (IFB and intercom) and to any IP device that needs to use the IP-PIPE that it can establish with the field unit.

The LU2000SMPTE A/V output implement compliance with 2110-10, 2110-20, 2110-30. It does not support metadata over SMPE.

Figure 48: LU2000SMPTE rack-mounted at Rai studio.

In this special setup, due to RAI lab IT policies, the LU2000SMPTE needed to connect to 3 different IP subnets: *(a)* the internal network for the A/V packets incoming from the public network and the communication it has with the field unit *(b)* the internal network to which the Rai master PTP clock was connected *(c)* the subnetwork to which it had to output its SMPTE A/V packets.

4.6 Interfaces

Live scenario

Diagram 49 below illustrates the four key element for live production scenario (scenario 1) of UC2: Camera interface, 5G S-NPN RAN+CORE, Media- and control-gateway and finally Live Production backbone.

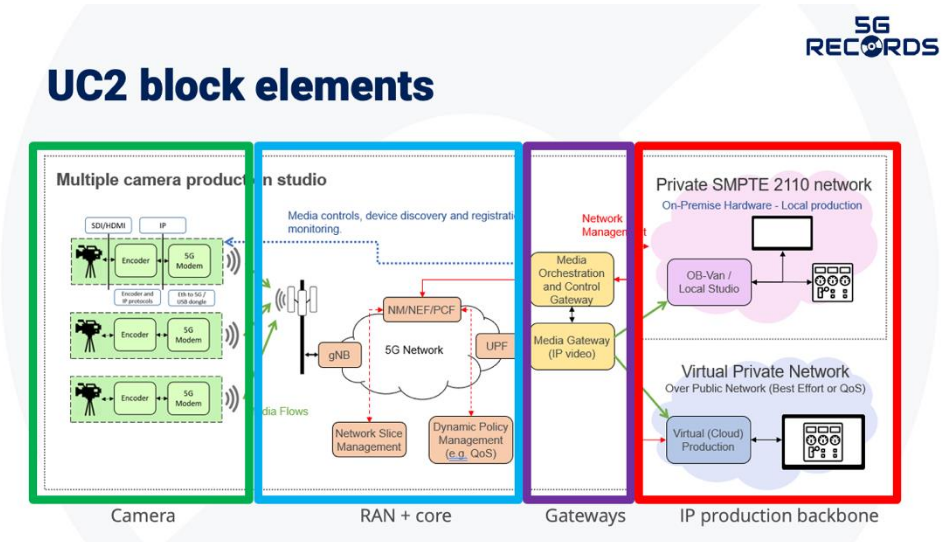


Diagram 49: Key element for live production scenario (scenario 1) of UC2.

For this multicamera production, two or more production-camera with encoder/decoder are connected to Fivecomm 5G modem. This connects to 5G S-NPN, in this first stage Ericsson lab-testbed in Aachen. Media- and Control-gateway function as the bridge between RAN+Core and IP production, to interface ST 2110 IP video and Media Orchestration/control.

Live Production can run on on-prem hardware as local production or with virtual private network (top right), as remote cloud production (bottom right), as shown in Diagram 50.

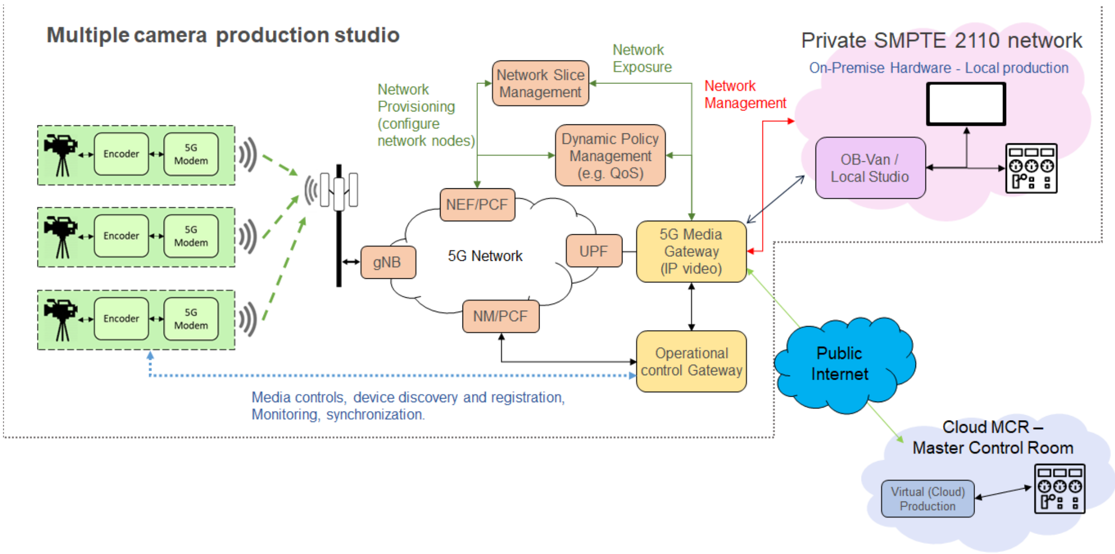


Diagram 50: UC2 Live scenario architecture.

Contribution scenario

The following diagram describes the interfaces of the remote production scenario (scenario 2) of UC2. The internal Aachen lab 5G SA network interfaces are already detailed elsewhere. As for the components' interfaces, the LU800Pro is connected via 1 to 4 SDI connectors to a fed (camera or video player using a splitter to create up to 4 streams from the single stream). A headset is also connected to the LU800Pro as well as a Cyanview IP controller, which in turn is connected to relevant cameras which it can control (iris, shading). The LU800Pro contains the embedded Sierra Wireless modems and was also connected via RJ45 to external modems and routers. These in turn are cabled into the lab network (not transmitting over the air). The network is connected via standard IT components and routing to the public internet. On the Rai studio side, the receiving video decoder is connected to 3 IP networks as depicted, for the various inputs/outputs per the lab installation and IT security policies.

Diagram 51 also shows the remote production scenario but with the inclusion of an UL load emulation component which transmits packets via a 5G router cabled into the 5G SA network in desired packet rates.

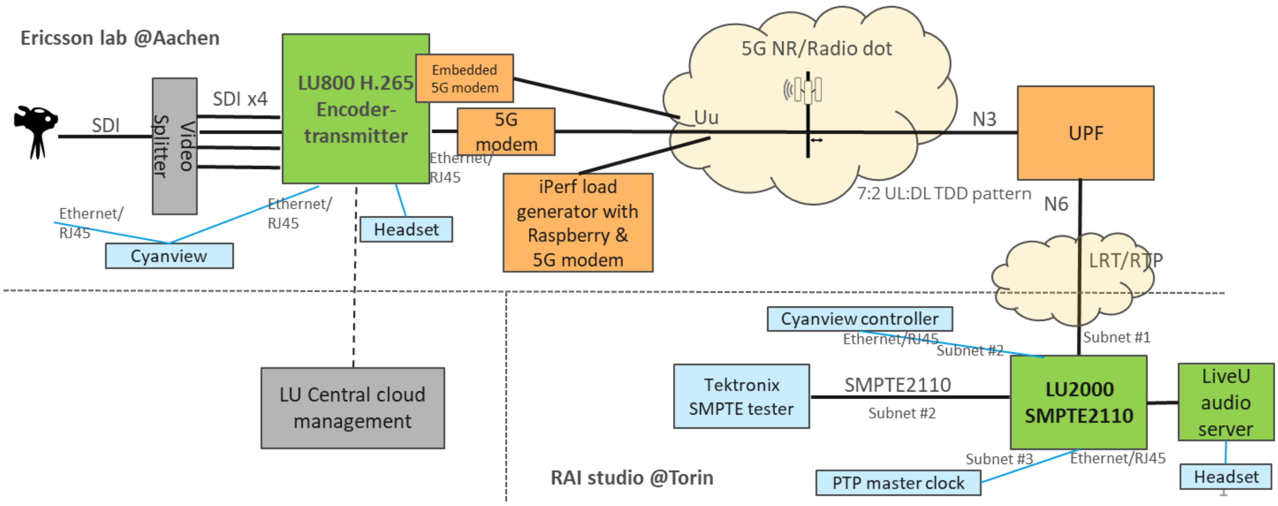


Diagram 51: Remote production (scenario 2), UC2 interfaces.

5 UC3 Components - Live Immersive Content Production

5.1 General architecture

The Use Case 3, for Live Immersive Media Production, comprises the deployment of the end-to-end video chain for a content production pipeline based on Free Viewpoint Video (FVV). This implies the capture of the scene with different cameras, the generation of a virtual view controlled via camera operation, the processing of the resulting live stream for its distribution, and the actual delivery to end users located in different geographical points. Additionally, the use case addresses the deployment of such chain, as well as the setup of its different elements, in a temporary way: which implies that all resources may be commissioned and decommissioned for the use case implementation. A full description of the rationale, scope, and requirements of the use case can be found in Deliverable D2.1.

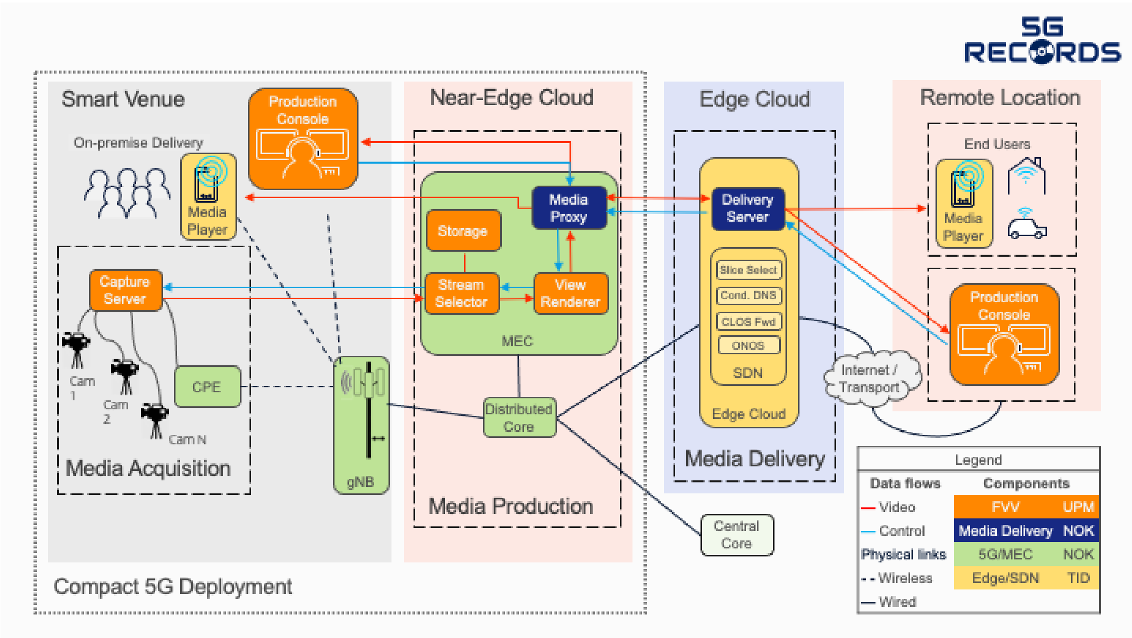


Figure 52: Use case 3 high-level architecture

Figure 52 describes the high-level architecture of the Use Case. It is split in several locations:

- **Smart venue**. The location where the live event takes place, with a deployment of 5G millimeter wave (mmW) RAN. It includes: *i)* the FVV Media Acquisition system, *ii)* the local camera operator, who operates the virtual camera offered by the FVV platform, and *iii)* end users accessing the live production of the event.
- **Near Edge**. It hosts the VNFs required to process the FVV streams (Media Production) and generate the virtual views which are produced.
- **Cloud Edge**. It hosts the VNFs required to deliver the produced streams to end users and other third parties (Media Delivery).
- **Remote locations:** where users will access the live production of the event either as end-consumers of the content or as remote producers that will include the media flow into an Edited Content Creation.

The use case is built on top of two media components (FVV system and Media Deliver) and two network components (5G/MEC and Edge/SDN).

The **FVV system (UPM)** is responsible of two media functions: acquisition and production.

- **Media acquisition:** A set of capture servers, each one receiving the stream from up to 3 stereoscopic cameras. The capture server generates the stream for each camera (RGB + Depth) and sends it, via the 5G network, to the media production VNFs. Capture server function is implemented in *baremetal* due to the tight coupling of the cameras and the processing.
- **Media production:** The View Renderer function implements an FVV renderer and encoder for live or on-demand requests: it creates a virtual view (similar to a virtual camera) whose trajectory is remotely controlled by a camera operator. Two additional VNFs are required to support the instantiation of multiple View Renderers: a Stream Selector, which receives the flows from all the cameras and distributes them properly to the View Renderers, and a Storage system for on-demand production.

The **Media Delivery (Nokia)** system receives the output of all the View Renderers and delivers it to third parties: content producers or broadcasters, in contribution quality, and end users, in streaming quality. It also includes a Media Proxy module.

The **5G and MEC** component **(Nokia)** provides two key enablers to implement the use case: 5G network (UE, gNB and core) with millimetre Wave (mmWave) access and Multi-access Edge computing. The use of mmWave is needed to provide enough uplink capacity for the FVV streams. The MEC platform provides GPU capacity to run the Media production functions. It is located close to the gNB to minimise end-to-end latency.

There are two different configurations of the 5G/MEC platform:

- **Public network** (“5G Theatre”). A reference configuration of the deployment using public 5G infrastructure. A 5G mmWave cell and a MEC have been deployed in Segovia (Spain), using Nokia infrastructure and Telefónica mobile network. This was the focus of the Phase 1 of the project.
- **Non-public Network** (“5G Festival”). A “compact” 5G deployment, that can be easily deployed ad-hoc in a temporal location where the live event of interest may happen. A full 5G private network, including a mmWave cell, and a MEC platform have been developed and tested in Madrid (Spain). It is connected to Telefónica transport network with a FTTH access. This was the focus of Phase 2.

The **Delivery Cloud & SDN (TID)** provides the edge cloud infrastructure required to host the Media Delivery VNF and deliver the video with guaranteed QoS (network slicing) to end users. This system comprises three main elements:

- An Edge Cloud deployment with computing, network, and storage capabilities.
- A set Software Defined Network (SDN) modules which implements the network slicing functionality.
- An experimental video player, which feeds application-level KPIs to the SDN to trigger slice selection functions.

The Delivery Cloud & SDN component is deployed in Telefónica Madrid-Peñuelas data centre, and it is integrated with the production transport network of Telefónica.

5.2 Data flows

5.2.1 Live Immersive Content Production

The following diagram show the main data flows considered in the system.

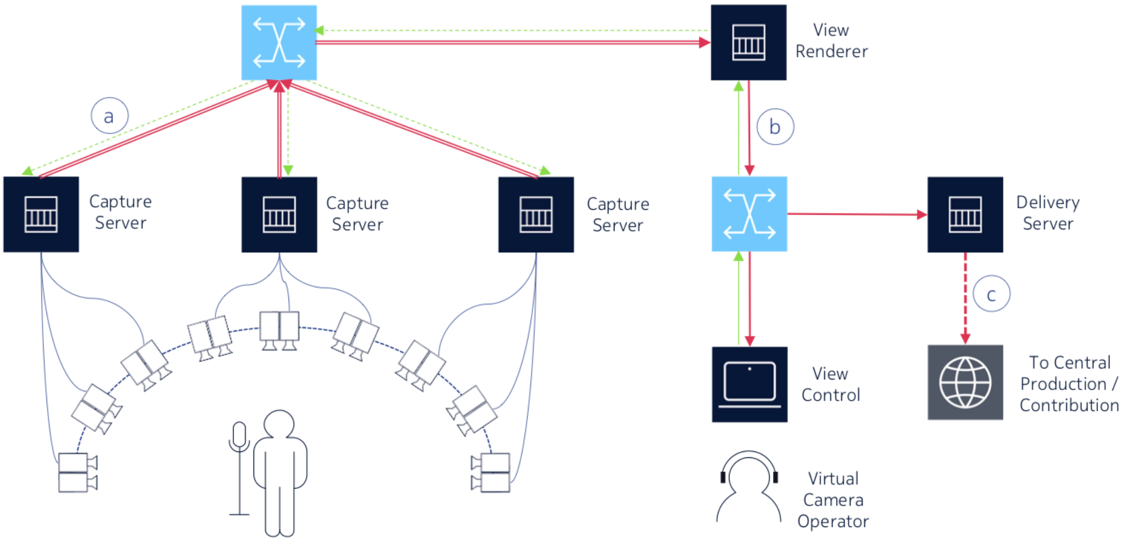


Figure 53: Free-Viewpoint Video production data flows.

a) Each capture server receives the stream from up to three stereoscopic cameras, and it generates one RGB+D(epth) stream per camera, which it sends to the view renderer. Once a frame is captured, the RGB data is encoded using an H.264 lossy scheme, packetised and transmitted over RTP. On the other hand, the corresponding depth map is computed and post-processed prior to its encoding and transmission. The post-processing stage consists of depth correction, 12-bit quantisation of the computed depth map, and bit re-arrangement allowing to fit a 12-bit depth map frame on an 8-bit YUV 4:2:0 frame (8 bits for Y, and 4 bits for U and V). The post-processed depth map is encoded using an H.264 lossless scheme, packetised and transmitted over RTP. Finally, live monophonic audio is captured, encoded and transmitted over RTP.

b) The view renderer is in charge of performing the computation of the virtual view according to the view control instructions. The view renderer receives as input the video streams (both RGB+D data) and the control messages associated to the desired position of the virtual camera to create a synthetic view. Once a frame has been synthesised, it is encoded and sent to the media delivery server over RTP. In parallel, the resulting frame is sent to the view control module so the producer or the virtual camera operator is able to monitor the view of the virtual camera. Using the view control, the camera operator can choose freely the desired virtual view at any moment and visualise the currently rendered view.

c) The delivery server delivers the rendered views to all the users of the system, both professional (typically content producers or broadcasters) and end users. The output of the delivery server is standard live streaming (HLS), and it can be sent to content producers or to end users.

In the project, this architecture is integrated on the network as described in Figure 54. Capture servers, view control and some end users' UEs are located on a trial site and connected to a 5G mmWave base station (gNB), including the radio unit (RU) and the baseband unit (BBU). Video processing functions (view renderer and delivery server) are virtualised and deployed as VNFs in the edge cloud. Content producers are located remotely. An end-to-end slice (shown in gold colour) guarantees QoS across all network segments, including Telefónica's nation-wide transport network

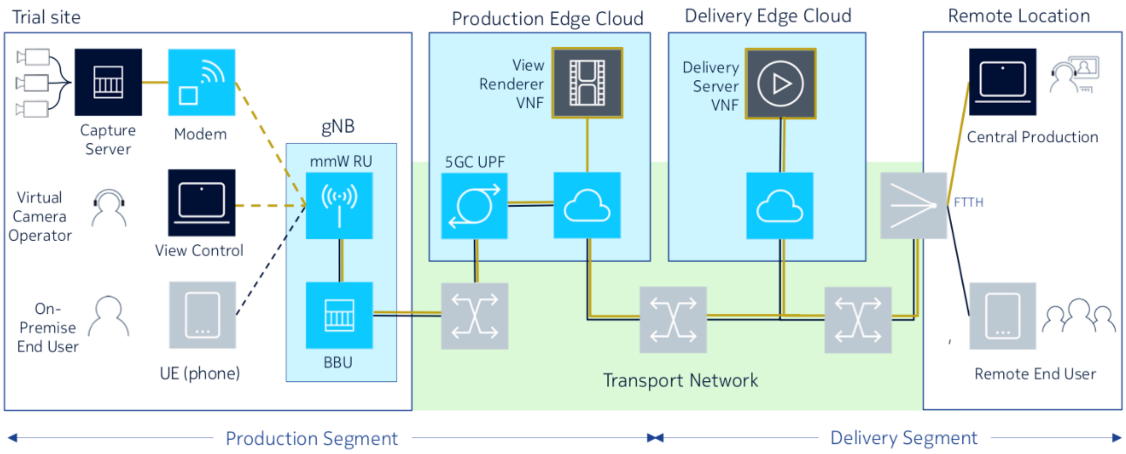


Figure 54: Free-Viewpoint Video production over a public 5G network.

In the second phase of the project, a modified version of this network architecture has been designed, to support a compact deployment of the 5G network and the production edge capacity over a Non-Public Network (NPN) with MEC capabilities. This is shown in Figure 55.

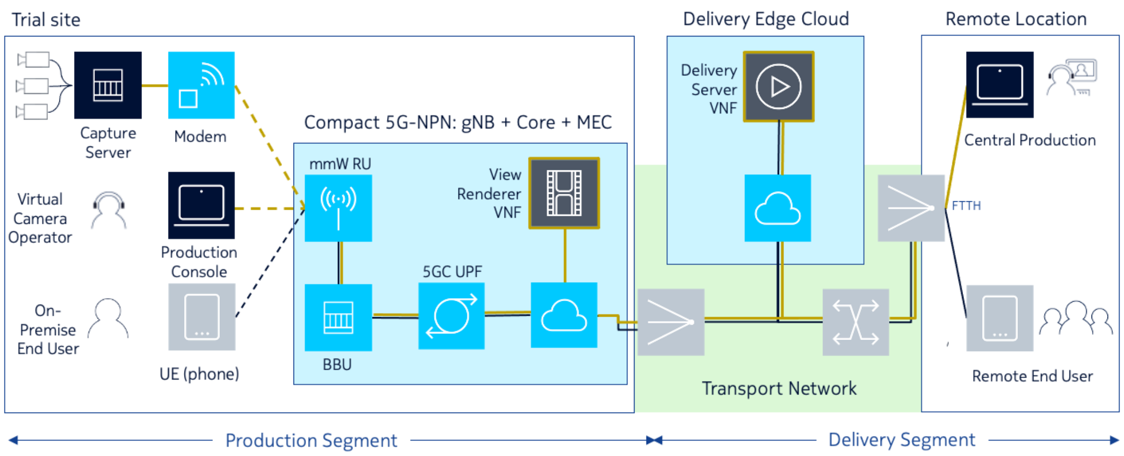


Figure 55: Free-Viewpoint Video production over a compact 5G-NPN.

5.2.2 5G end-to-end user plane protocol stack

The following figure shows the protocol stack for the video data flows.

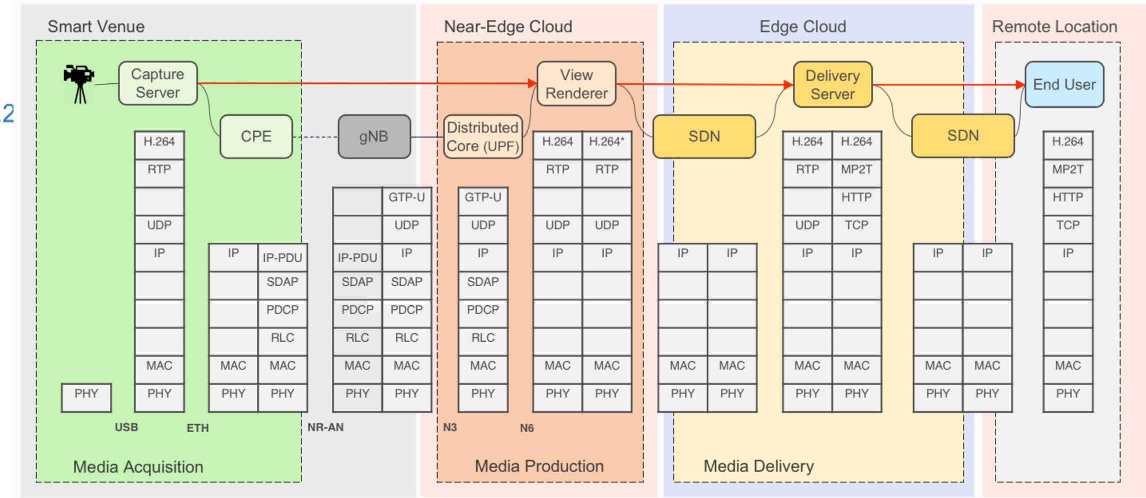


Figure 56: UC3 user plane Protocol Stack.

Capture Server – View Renderer (per physical view)

Each camera produces two different streams that are sent in two separate RTP streams: one for the RGB data and the other for the depth data:

- RGB stream: H.264 AVC, low latency scheme: No B frames. GOP length of 300 frames. Target bitrate between 5 and 15 Mb/s. Two pass VBR.
- Depth stream: 12-bit depth encapsulated on a regular 8-bit AVC lossless encoding profile (no control of target bitrate). Low latency scheme: No B frames. GOP length of 300 frames.

The resulting NAL Units are mapped directly onto RTP packets. RTP MTU is configurable, with a maximum of 1472 bytes, to avoid IP fragmentation. RTP is send over UDP/IP/5G-RAN using unicast communication. Capture timestamps for each frame are used as the RTP timestamp after a 90 kHz clock conversion.

In addition, the audio stream is encoded using Opus codec and RTP encapsulated. RTP timestamps are generated using the same clock source as the video timestamps.

View Renderer – Delivery Server (per virtual view)

Each virtual camera is sent to the delivery sever in a single unicast H.264/RTP/UDP/IP stream. The encoder configuration parameters are as follows:

- Low latency scheme: No B frames. GOP length of 300 frames.
- Contribution quality. Target bitrate between 10 and 20 Mb/s. Two pass VBR.

The audio stream is sent to the delivery server in a single unicast Opus/RTP/UDP/IP stream. The encoder configuration parameters are as follows:

- Sampling frequency: 48 kHz
- Output bit-rate: 128 kbps

Delivery Server – End User (per virtual view)

Each virtual camera is sent to the media delivery in HLS. Transcoding is available for offline content. Target segment size is 6 seconds. The baseline transcoder configuration is:

Table 18: Associated delivery bitrate and reference bitrate per resolution

Quality ID	Resolution	Reference Bitrate (Mbps)
SD	960x540	1.5
HD	1280x720	3.0
Full-HD	1920x1080	6.0
3K	2560x1440	12.0
4K	3840x2160	24.0

5.2.3 Monitoring

Systems include a monitoring module to actively monitor system performance during operation. The architecture of system monitoring is common for all monitored elements:

1. Application logs are produced with the relevant KPIs and retrieved by a Telegraf service.
2. Additionally, Telegraf service retrieves performance logs from the system (CPU consumption, network logs, disk I/O stats, etc.), both at VM and at baremetal level.
3. Telegraf logs are sent to an InfluxDB + Grafana centralised service.

The following applications will generate monitoring logs: *(i)* capture server, *(ii)* media render, *(iii)* media delivery, and *(iv)* end client. Additionally, the near-edge and edge platforms will also be monitored. More details on the monitoring capabilities for this use case will be included in deliverable D4.2.

5.3 List of Components

5.3.1 5G-Ready FVV Live (UPM)

FVV Live is a real-time operation Free View-Point Video system that allows real-time navigation through a scene using a virtual camera. FVV-Live system has been modified to integrate with a 5G network and some of its components have been migrated to the cloud as VNFs. Also, new modules have been developed specifically to allow the system to work on a 5G and cloud-based environment. Figure 57 shows the general architecture and connection of the 5G-Ready FVV-Live modules. The main components of FVV-Live are:

- Capture Servers
- Capture Simulator
- View Renderer
- Production Console
- Stream Selector
- Storage and FVV Replay

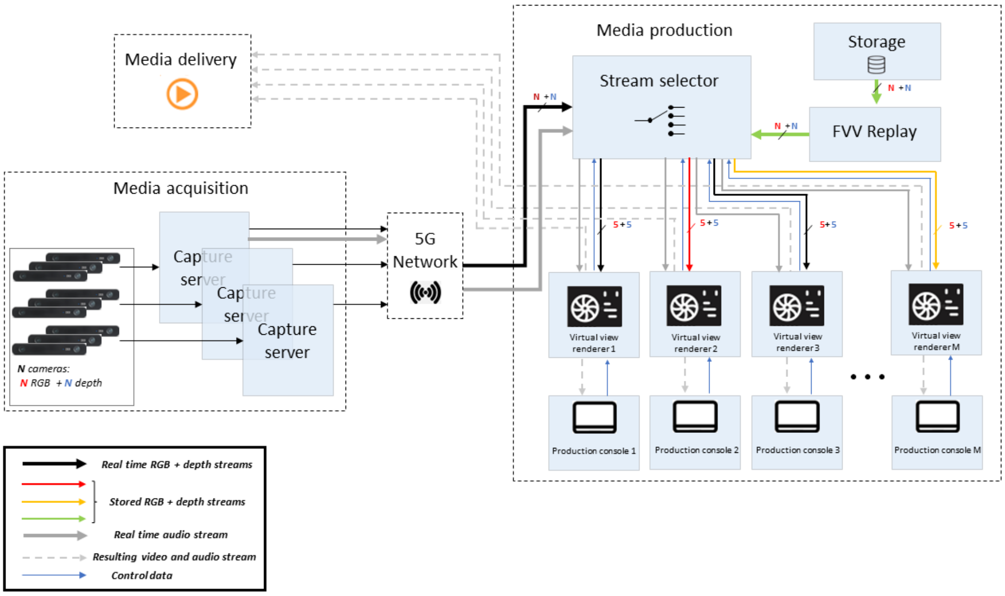


Figure 57: General architecture of FVV-Live integrated on the 5G and cloud-based environment.

5.3.1.1 Capture Server

Currently, FVV-Live video acquisition process is performed by 3 Capture Servers (CSs). These CSs capture colour and geometric information of the scene, specifically RGB+D(epth) information. The acquired information is encoded and transmitted so the View Renderer can process those video streams and compute the synthetic virtual view according to the desired virtual camera position.

The main characteristics of the CSs are:

- Intel Core i7 6850k @ 3.6Gz x 12
- 64 GiB
- 2xPCIe USB 3.0 controller
- NVIDIA QUADRO P4000 (mainly for encoding/decoding)
- NVIDIA GTX 1080 (mainly or depth maps processing)

Each one of the CSs manage 3 Stereolabs ZED cameras so a total of 9 RGB+D streams are delivered. Each stereo camera is composed by two 2.2 MegaPixels (MP) wide angle cameras with 110º diagonal FOV and f/2.0 aperture and a baseline of 120 mm. The cameras are wired to the CS by a USB3.0 cable.

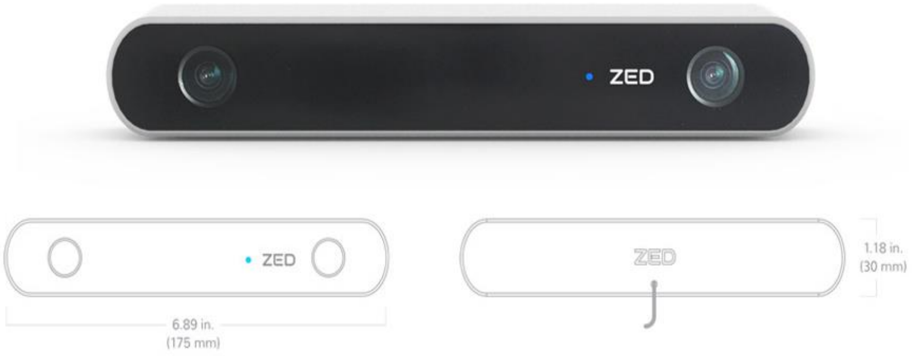


Figure 58: Stereolabs ZED camera.

Each CS processes data received from 3 cameras. There are two different pipelines depending on the type of data, i.e. whether data is RGB or Depth. On the one hand, RGB streams are H264 encoded and transmitted over RTP using for the encoding process the NVIDIA P4000 GPU. On the other hand, depth maps are computed using the GTX 1080 GPU and post-processed for a convenient transmission. This post-processing stage consists of two steps:

- 12-bit quantisation
- Bit re-arrangement algorithm that maps 12 bits depth data into an 8-bit YUV 4:2:0 frame (8 bits for Y and 4 bits for U and V)

Once the depth maps have been post-processed, the resulting frame is H264 lossless encoded and transmitted over RTP.

In addition, previously to the actual operation of the system, the acquisition stage of the system (Capture Servers + cameras) should be calibrated. The calibration process is divided in cameras calibration and background modelling tasks and both operations are performed using custom software developed specifically for the system needs.

Several developments have been conducted to integrate FVV-Live acquisition module on the 5G Network. First, regarding networking, a traffic shaping algorithm has been developed and the maximum transmission unit (MTU) size has been adapted to the 5G network. Also, the key frame period (length of the Group of Pictures) has been modified to be slightly different for each encoded RGB+D stream. These developments allow a better performance of FVV-Live on the 5G network, specifically:

- The traffic shaping algorithm spreads packet bursts associated to each NAL Unit, over the complete frame period, so network saturation is heavily reduced.
- Avoid IP layer fragmentation due to the MTU size adaption.
- Spread key frames (typically the biggest frames in terms of payload) over several frame periods so the aggregated bit-rate curve is flattened.

In addition to the video acquisition process, live monophonic audio acquisition has been implemented and integrated in the FVV live operation pipeline. The audio signal is acquired using a single microphone wired to one of the Capture Servers. The audio signal is acquired at a sampling rate of 48 kHz and PCM encoded using 16 bits per sample. The PCM signal is then encoded using the Opus codec and encapsulated in RTP packets. Finally, the RTP audio stream is sent to the Stream Selector so it can be delivered to the View Renderers that request the RTP audio stream.

Regarding FVV control messages, the previous scheme has been migrated to regular UDP and TCP-based messages to maximise the interoperability along the 5G network architecture. The system has been modified so now MPI messages are only used for the camera synchronisation process at the initialisation of the system. The rest of the control messages (start and ending messages, transmission of configuration files for each capture server and computed background on the calibration stage) are all regular UDP and TCP-based messages.

Finally, video recording capabilities have been added to the acquisition module so, in parallel to the live execution, each one of the cameras RGB+D streams is saved to file and stored in the Storage module within all the associated information of the FVV sequence, i.e. the configuration files and the background files. These files allow to the Capture Simulator and Storage modules in combination with a View Renderer instance, to playback the recorded RGB+D streams.

5.3.1.2 Capture Simulator

To increase the testing capabilities of FVV-Live over the 5G network a capture simulation module has been developed. This Capture Simulator allows to easily test the FVV-Live performance over 5G for different scenarios avoiding the deployment of the complete capture setup (cameras and Capture Severs). Moreover, the Capture Simulator working with the Storage Module, allows to a View Renderer instance to work offline and synthetise a replay using pre-recorded sequences.

The Capture Simulator is able transmit the previously stored and H264 encoded RGB+D sequences in the same terms as the actual Capture Servers do (RTP-based transmission) including the network related developments that have been explained on section 5.3.1.1, i.e. the traffic shaping algorithm and the MTU size adaptation.

The Capture Simulator allows two different configurations to meet different deployment requirements. On the one hand, an instance of a Capture Simulator is able to simulate one Capture Server, so three different instances are needed to simulate the complete FVV Live capture setup. On the other hand, a single instance on the Capture Simulator is able to simulate several Capture Servers at the same time. Finally, it has also been virtualised as a VNF and integrated in the 5G chain.

5.3.1.3 View Renderer

The View Renderer is in charge of performing the computation of the virtual view according to the control of the production console (section 5.3.1.4). The View Renderer receives as input the video streams (both RGB and depth data), from the Capture servers or the Storage module, and the control messages, from the Production Console, associated to the desired position of the virtual camera.

The View Renderer needs the resulting information of the calibration and background modelling processes to compute the virtual view. Additionally, the synthetic view is computed based on the information of the nearest real cameras (currently five cameras) to the current virtual camera position on a layered synthesis approach. Thus, at any moment, the view renderer needs to receive from the stream selector, at least, five RGB streams and five depth streams. The different synthesis layers are:

- Foreground: Based on the RGB and depth information captured by the cameras
- Background: Based on the background model (RGB and depth) computed on the calibration stage

Once a frame has been synthetised, it is encoded and sent to the Media Delivery module. Also, the resulting frame is sent back to the Production Console so the producer or the virtual camera operator is able to monitor the view of the virtual camera at any moment.

For a successful operation of the View Renderer over the 5G network, several modifications have been developed. Regarding to the network related developments, a packet reordering algorithm has been implemented. The main task of this algorithm is to mitigate the disproportionate effect that the out of order packets arrival (due to actual losses and subsequent retransmissions in the lower network layers) have on the video decoding process. Out of time received packets are marked as lost, so the incomplete packet batch is delivered to the upper layers.

The initial version of the View Renderer was designed to work as a bare-based server and use 2 GPUs one for the decoding and the other for the synthesis computation processes. To add flexibility to the configuration of the 5G-Ready FVV Live system, the View Renderer has been modified to been able to work with different number of GPUs, i.e. single and dual GPU operation, according to the GPU capabilities of the MEC in the 5G deployment.

As it has been said in Section 5.3.1.1, live monophonic audio acquisition has been integrated on the FVV live pipeline. The View Renderer receives the audio RTP stream and retransmits it to its associated Production Console and to the Media Delivery module, in parallel to the resulting synthetic RTP video stream. In this way the virtual camera operator is able to monitor both the live audio and synthetic video signals.

Additionally, mainly for monitoring purposes, logging functionalities have been implemented in the View Renderer. These logging functionalities allows to monitor the performance of different stages of the View Renderer, i.e. the synthesis frame period, the received bit-rate and the packet losses detected on the incoming RTP streams. These metrics are measured in real time and bounced in logging files.

Also, the View Renderer have been modified to been able to work with both, actual cameras (online operation) and the Capture Simulator + Storage modules (offline operation). This modification has been developed in a way that the View Renderer is fully decoupled of the previous steps, so its operation is exactly the same no matters if the streams source are the actual cameras or the Capture + Storage modules. These two operation modes add an extra layer of functionality to the system, been able to operate live content or a replay of pre-recorded content. Finally, the View Renderer has been virtualised as a VNF, using Docker-based container technology, so it can be deployed on a cloud-based environment. Also, the virtualisation of the View Renderer allows to deploy multiple instances working with the same streams so the production of a scene can be covered by multiple virtual cameras working in parallel.

5.3.1.4 *Production console*

The production console is the component that allows the producer/camera operator to choose freely the desired virtual view at any moment and visualise the current view that has been synthesised by the view rendered.

The previous versions of the virtual camera control (smartphone-based software tool) have been updated to the Production Console, that is a fully re-developed version of the virtual camera control tool. The Production Console is PC-based and can be easily installed and operated on a regular laptop. Each instance of the Production Console allows the user to configure and stablish a session with a given instance of a View Renderer. The Production Console allows the control of the virtual camera position using the PC keyboard and the corresponding UDP-based control messages are transmitted to the Virtual Renderer. The Production Console receives in turn the encoded resulting synthetised view form that position and the RTP audio stream.

Additionally, Motion-2-Photon measurement capabilities have been added to the Production Console so it is possible to measure the time span between a virtual camera movement command is generated on the console and the associated virtual view is rendered on the console screen. The methodology of the Motion-2-Photon measurement has been fully described in document D4.1.

The configuration of the session is performed through a GUI (Figure 59) and allows to configure the operation mode (Normal or M2P measurement).

Figure 59: Production Console GUI.

5.3.1.5 Stream Selector

The Stream Selector orchestrates the delivery of the FVV streams among the different components of the system, i.e. the delivery of the RGB+D and audio streams to the different instances of the View Renderer.

There are two different RGB+D streams sources depending on the operation mode of the current session:

- Live RGB+D and audio streams incoming from the capture servers on an online session configuration.
- Pre-recorded RGB+D streams incoming from the Capture Simulator + Storage modules on an offline session configuration.

The Stream Selector receives from each View Renderer instance a control message that indicates the RGB+D streams that are needed to compute the current virtual view based on the virtual camera position. These streams are those corresponding to the five nearest actual cameras to the current virtual camera position. These control messages are constantly sent from each View Renderer instance to the Stream Selector, so the Stream Selector is able to track which subset of cameras is associated to each View Renderer at any moment. The Stream Selector has been virtualised as a VNF using docker-based container technology.

5.3.1.6 Storage and FVV Replay

The Storage module and the FVV Replay adds to the system the offline operation capabilities. The Storage module stores pre-recorded sequences so they can be transmitted to a View Renderer instance. In this way, the system is able to work offline and synthesise a replay of past scenes. The RGB+D sequences are recorded by the Capture Servers and stored as H264 encoded streams. In addition to the RGB+D streams, configuration files and background files associated to that specific sequence are stored too. The Storage module has been implemented on a Virtual Machine running on Nokia's MEC, so both the RTP streams and the additional files are accessible by any element of the system.

The FVV Replay module is able to read the stored H264 RGB+D streams and transmits them to the Stream Selector. The Stream Selector resends the corresponding streams to each View Renderer as it has been explained in section 5.3.1.5. The FVV Replay module has been implemented using the functionalities provided by the Capture Simulator.

5.3.2 Compact 5G Network (NOK)

The 5G network deployment involves both RAN, Core, and MEC capabilities. Two different configurations are provided: public network, deployed in Segovia in Telefónica network, and compact NPN. They share most of their elements and therefore they are described jointly.

5.3.2.1 Hardware and platform

The complete gNB and MEC infrastructure can be deployed with a reduced footprint, comprised of *(i)* a Nokia 7250 IXR-e Router for connectivity, *(ii)* a Nokia OpenEdge MEC platform to support computing capabilities (5G core and VNFs) and *(iii)* a Nokia AirScale BaseBand Unit.

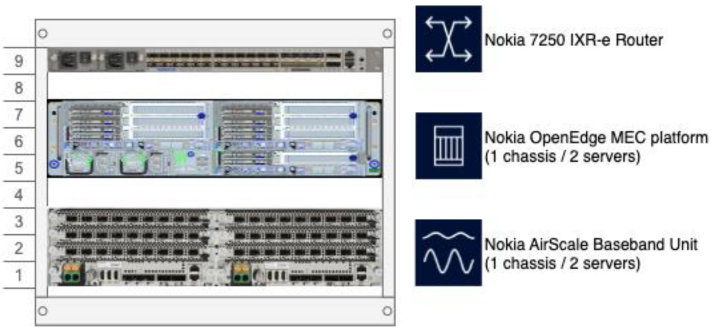


Figure 60: Compact 5G Setup – Hardware View.

The AirScale BaseBand Unit is composed of:

- AMIA cabinet.
- 5G module: ABIL base band module and ASIA controller.
- 4G module: ABIA base band module and ASIA controller.

The hardware platform for the MEC is the Nokia Airframe OpenEdge server, a compact and robust x86 server platform optimised for edge computing. The deployed system includes an OpenEdge 3U chassis, which can hold two 2U servers. Each 2U server has the following characteristics:

- Intel Xeon 20 cores, 192 GB RAM, 480 GB SSD
- 2x Tesla T4 GPU
- 25GbE dual port OCP NIC card CX5

The server runs Ubuntu 20.04 with an installation of MicroStack (a compact version of OpenStack) which is used to administrate and orchestrate the Virtual Network Functions.

OpenStack software controls large pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API. OpenStack works with popular enterprise and open-source technologies making it ideal for heterogeneous infrastructure. Microstack is a solution from canonical (Ubuntu) which consists in a simple hardcoded configuration type OpenStack installation based on snap. Snaps mounts a read only file, so no configurations changes are allowed. As a consequence, Microstack installation has been modified to support the specific functionalities required for the project, in particular, the visibility of GPUs from the VNFs.

In addition to rack-mounted equipment, the following elements are needed in the deployment:

- 5G radio head: Nokia AEUB AirScale MAA 2T2R 512AE n257.
- 4G radio head: Nokia AHEGB AirScale Dual RRH 4T4R B1/3 320 W.
- 5G UEs: Askey RTL6305. It supports mmW and Sub 6GHz for 5G NR NSA and SA core network, offering the flexibility of combining 5G NR and 4G LTE. It can be used outdoors, as it complies with IP67 waterproof standard. It offers gigabit Ethernet connectivity with the capture servers.

5.3.2.2 *Software and configuration*

Figure 61 shows the different systems and networks involved in the deployment.

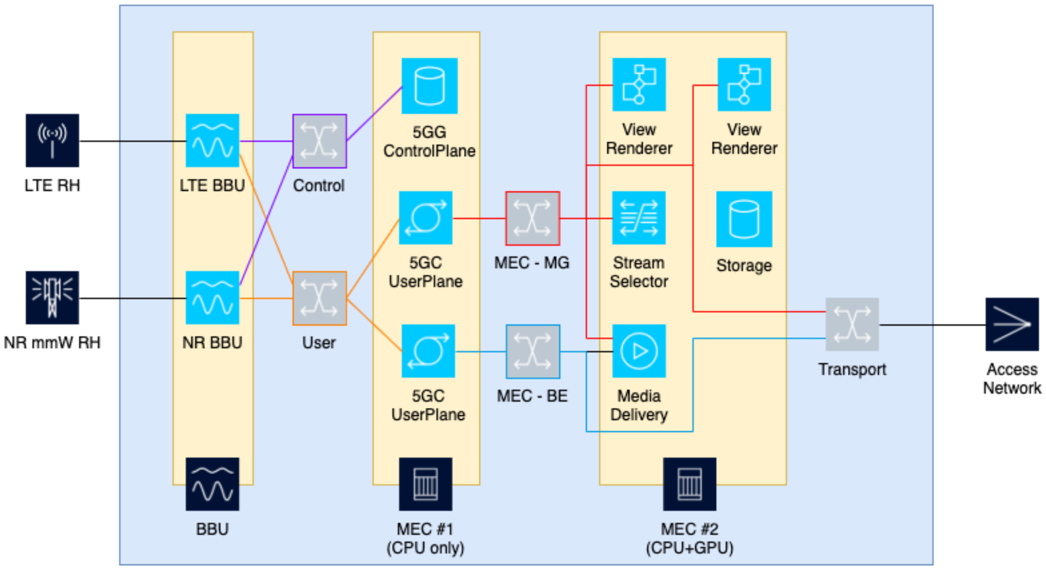


Figure 61: Compact 5G Setup – Network View.

The BBU contains a 5G gNB in NSA 3X configuration. It is based in E-UTRAN New Radio - Dual Connectivity (EN-DC), a technology that enables introduction of 5G services and data rates in a predominantly 4G network. UEs supporting EN-DC can connect simultaneously to LTE Master Node eNB (MN-eNB) and 5G-NR Secondary Node gNB (SN-gNB). This approach permits cellular providers to roll out 5G services without the expense of a full scale 5G Core Network.

An EN-DC enabled UE first registers for service with the 4G EPC. The UE also starts reporting measurements on 5G frequencies. If the signal quality for the UE is enough for a 5G service, the LTE eNB communicates with the 5G-NR gNB to assign resources for a 5G bearer. The 5G-NR resource assignment is then signalled to the UE via an LTE RRC Connection Reconfiguration message. Once the RRC Connection Reconfiguration procedure is completed, the UE simultaneously connects to the 4G and 5G networks. In this configuration, user data traffic directly flows to the 5G gNB part of the base station. From there, it is delivered over the air interface to the mobile device. It is also possible to forward a part of the data over the X2 interface to the 4G eNB part of the base station and from there to the UE.

The 5G core is installed in the one OpenEdge computing server. Two different core distributions are used. The Public Network deployed in Segovia is based on the Nokia Cloud Mobile Gateway (CMG), a multi-functional packet core mobile gateway that provides increased deployment flexibility, elastic scale, high reliability, and the capacity to deliver a full range of mobile and IP services. CMG implements UPF and provides SGi/N6 interface to the rest of the elements of the network, particularly to the MEC platform running the Media Production VNFs. The compact implementation, depicted in Figure 61, uses a full lightweight 5G core supported by Nokia Cloud and Network Services for pilot deployments.

The second OpenEdge computing server is enhanced with GPU capabilities, and it installs the Media Production VNFs for the FVV and the Media Delivery components.

Network slicing is implemented over the NSA configuration by defining two different SGi subnetworks, supported by two different instances of the 5G User Plane. UEs will be assigned to the Multimedia Gold or to the Best Effort QoS by the 5G core, depending on whether they are used for production purposes (Capture Servers, Production Console, VIP end users) or not (regular end users).

- Production/VIP users are assigned a high priority in the Radio Access Network (QCI/5QI = 6) and the Multimedia Gold SGi subnetwork.
- Regular users are assigned the standard priority in the Radio Access Network (QCI/5QI = 9) and the Best Effort SGi subnetwork.

5.3.3 Nokia 7250 IXR-e Router allows to define different VLANs with different DSCP priorities for each network, to guarantee that QoS policies are applied both in transport and in RAN.

Media Delivery VNF (NOK)

The Media Delivery VNF performs the delivery of the rendered views to all the end users of the project, both professional (typically content producers or broadcasters) and end users. It can optionally be deployed as Media Proxy to ensure proper application-level traffic replication and connectivity at the near-edge cloud.

Its architecture is depicted in Figure 62. It is comprised of three main modules: *(i)* real-time video router and processor, *(ii)* multi-purpose origin server, and *(iii)* configuration management.

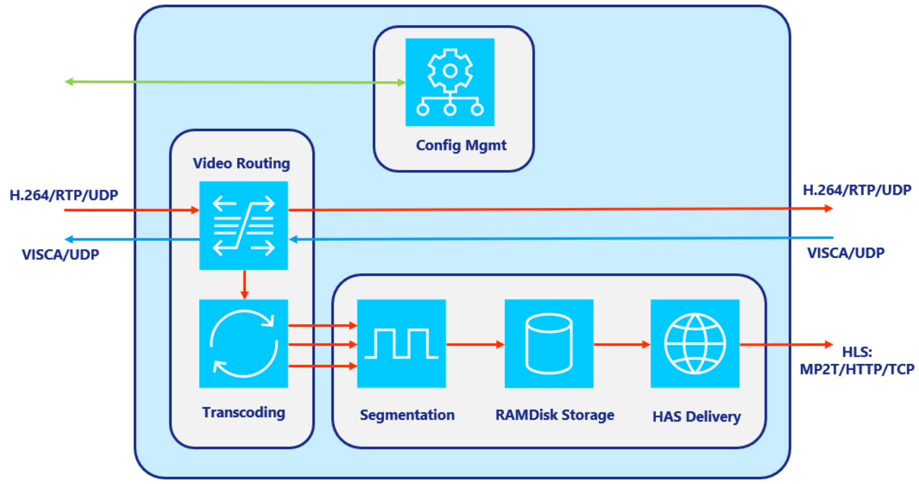


Figure 62: Media Delivery VNF.

Real-time video router and processor

It is composed of two main modules:

- Video routing: a low-latency application-layer video routing module, specialised in replicating, forwarding, and adapting video over RTP transport.
- Video processing: a flexible video processing chain based on Gstreamer, which can be used to transcode and remultiplex video. In our case, it is used to generate several qualities for the video stream, and to multiplex them in MPEG-2 TS so that they can be easily ingested in hydra module.

Multi-purpose origin server

It is composed, in a simplified manner, of the following subsystems:

- Live Ingest and segmentation: It receives the live video streams (using multicast MPEG-2TS/RTP/UDP) and generates a segmented video version in a mezzanine format (MPEG-2 TS segments and a private metadata manifest).
- Local Cache: The segmented video is inserted into a local cache (in disk or memory) that can store a sliding window of several minutes of content.
- HTTP Server: The segmented video is delivered using ABR protocols (e.g., HLS, DASH) to the end client. The video can be obtained from different sources:
    - ABR protocol from an external source (the CDN). Then the HTTP server acts as an efficient reverse proxy and delivers the content to the user terminal.
    - Live ingested into the local cache. Then the HTTP makes use of a processing library (“LibPers”) to dynamically generate the ABR content from the segmented mezzanine format.

Configuration management

A set of configuration files and monitoring scripts to handle the VNF.

5.3.4 Delivery Cloud & SDN (TID)

The Delivery Cloud & SDN component is an Edge Datacenter that Telefónica I+D has deployed in one Central Office in Madrid (Madrid Peñuelas), at Calle Torres Miranda Nº 8.

Figure 63: Madrid Peñuelas Central Office Building: edge cloud datacenter.

This Datacenter contains the following Hardware Elements:

- 1 x E1050 Pebble, 48x1G 4x10G (CUL), Hurricane2, 2C X86 BTL 2GB/4GB, Fixed 1xPSU 12V F2B, ICOS 3yr, Maintenance OS 3yr
- 6 x SDN Switches: Edgecore AS6712-32X 32x40G Trident2
- 3 x Nodes Intel Relion XO1132g Server from Penguin Computer
- 4 x Nodes AMD Altus XO1132g Server from Penguin Computer
- Open Vault (SAS12G JBOD/JBOF) ST7110G2-30A Wiwynn with over 100 TB SSD storage space
- OCP Bryce Canyon with over 150 TB with HDD storage

All these infrastructures are assembled in one 42-U rack and managed with Open Nebula Cloud management system software.

5.3.4.1 *Cloud infrastructure*

Physical architecture of the Edge Cloud is made up of a layer of networking, (6 switches in leaf-spine model), a compute layer, (7 OCP servers), and a storage layer (storage server and JBOD)

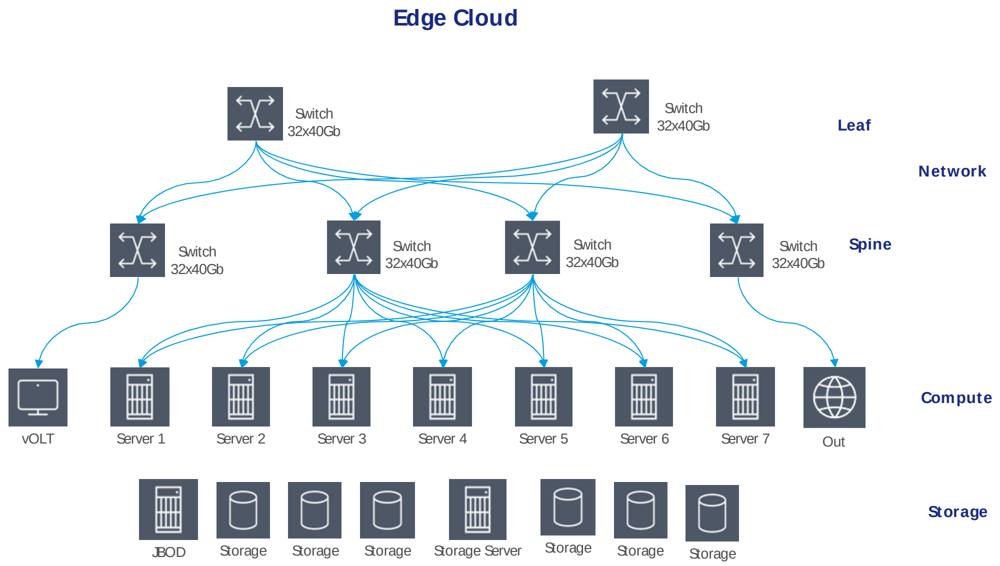
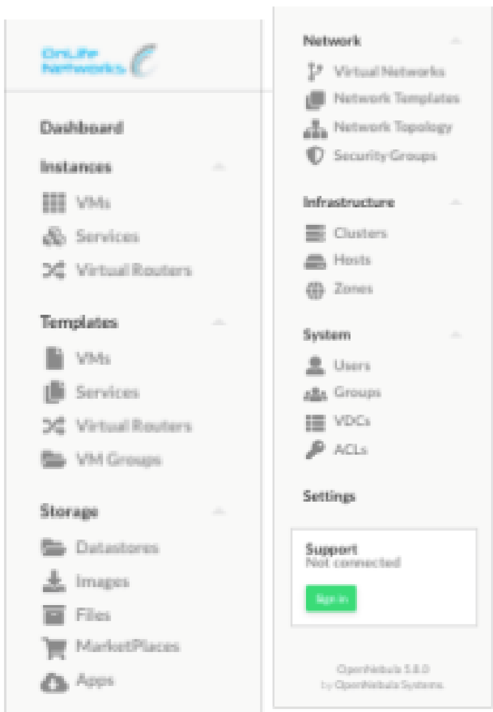


Figure 64: Leaf-Spine edge compute architecture.

Edge platform is based on three main modules: *(i)* compute virtualisation Subsystem based on OpenNebula, *(ii)* storage virtualisation subsystem based on GlusterFS, and *(iii)* network virtualisation subsystem based on SDN ONOS.

Compute virtualisation

It is an Opensource platform based on OpenNebula software for cloud computing. It is oriented to distributed and heterogeneous datacenters, providing virtual infrastructure to build private, public, and hybrid clouds. Further, it is Opensource under Apache License, and in particular, we use Linux KVM, as virtualisation manager.



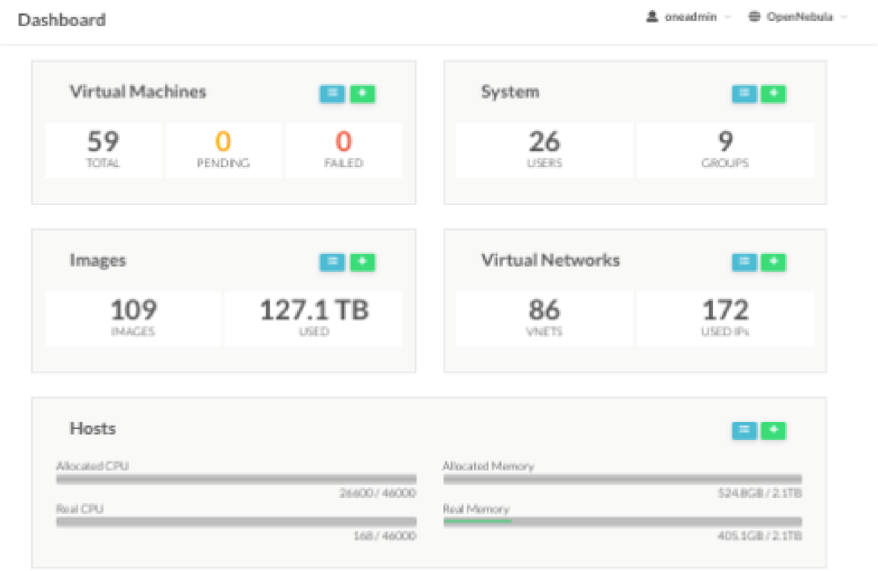


Figure 65: OpenNebula dashboard.

Storage virtualisation

For this layer we use GlusterFS software. This software is installed and configured into server hosts directly. For storage network GlusterFS replication is done using two separate networks. First network managed by SDN. And a second auxiliary network using the management switch. Each network has a different VLAN assigned. Also, Gluster configuration is done using at least two cluster for load balancing purposes.

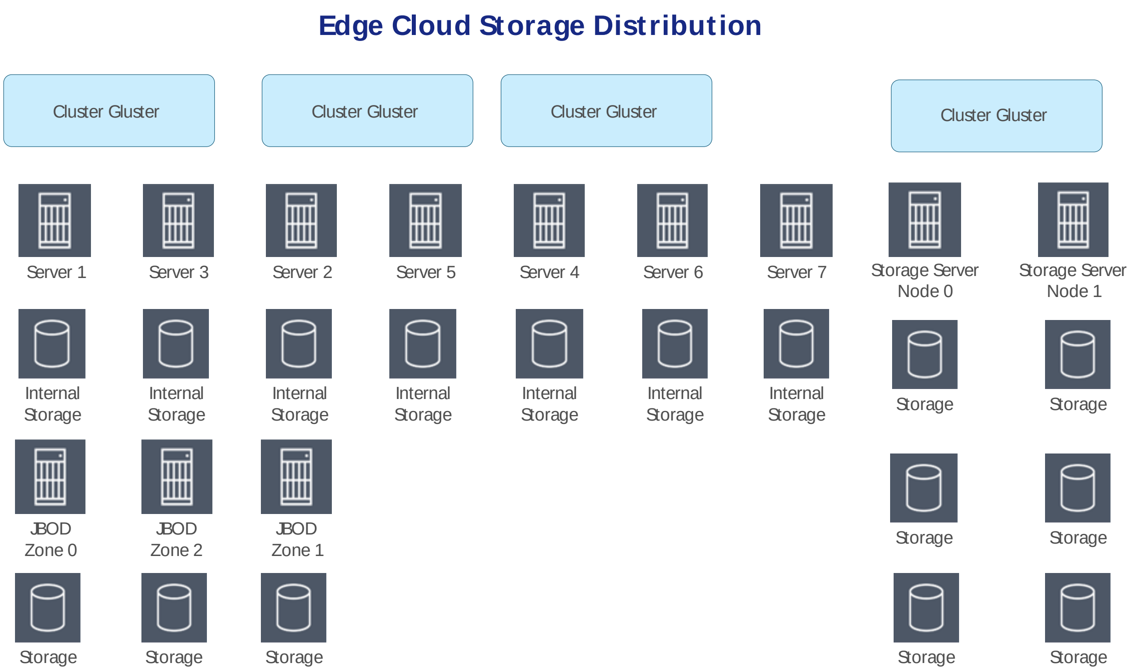


Figure 66: UC3 Edge Cloud Storage Distribution.

Network connectivity

It is a subsystem based on ONOS, Open Networking Operating System, ONOS provides the control plane for a software-defined network (SDN).

In the Edge cloud network solution, ONOS is used as the SDN controller, in charge of managing the switch fabric. ONOS provides a series of functionalities and software models to provide communication services to end hosts and networks.

The team is using an ONOS application called CLOSfwd, that is responsible for managing the CLOS fabric of Edge Cloud switching. In the case of the switching fabric, it is sought to create paths between pairs of end nodes. In Figure 67, the ONOS SDN GUI can be seen.

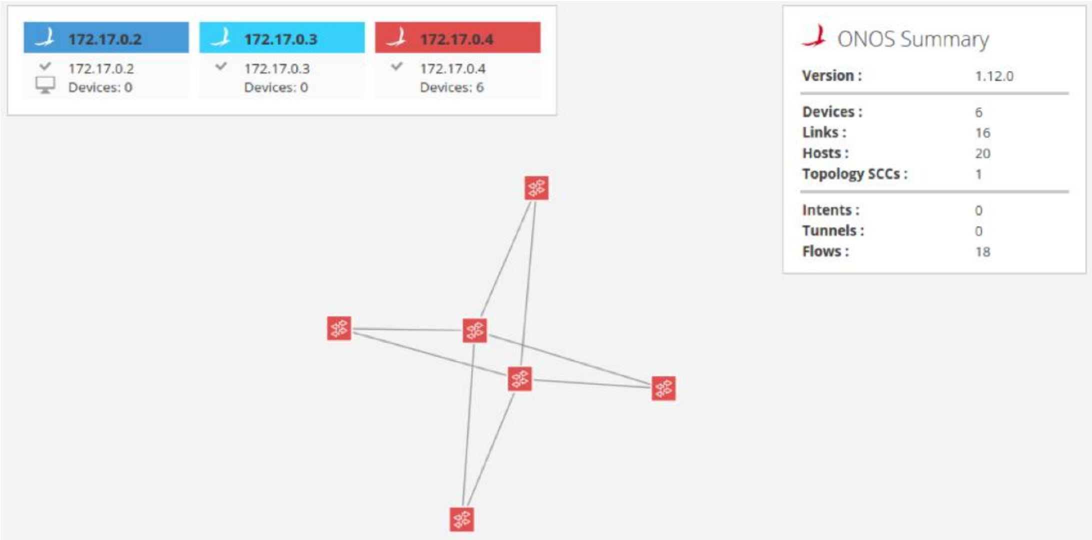


Figure 67: SDN GUI.

5.3.4.2 *End-to-end SDN*

Architecture of E2E SDN is defined based on the building blocks shown in Figure 68:

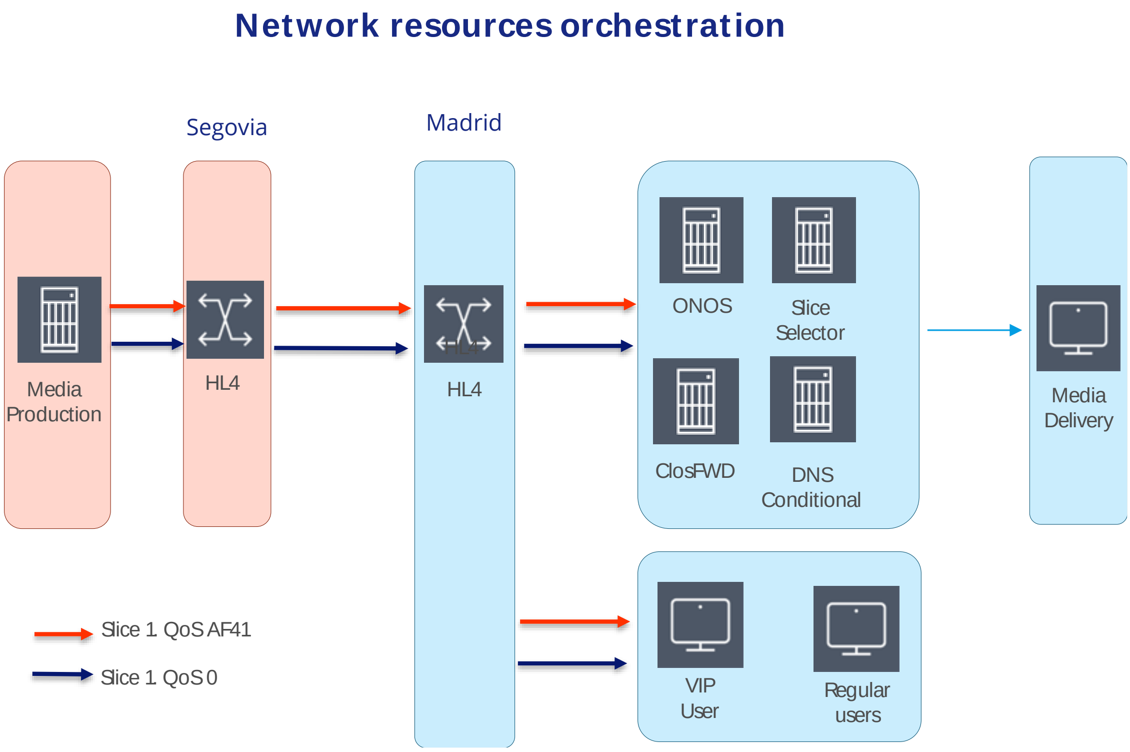


Figure 68: Network resources orchestration.

- **ONOS**: used as the SDN controller, in charge of managing the switch fabric, as described in section "Network connectivity subsystem".
- **ClosFwd**: application of the ONOS environment is responsible for managing the CLOS fabric of Edge Cloud switching.
- **Slice Selector**: Software component based on NGINX servers acting as reverse proxy with capability to redirect request to the correct slice based on the IP accessed and the URL. Combining server instances listening on different public interfaces and different VHOST to segment the traffic along the correct slice. It is connected to slices gold and best effort internal and external, and to the internal video delivery.
- **DNS Conditional**: implemented using opensource software bind and several views configuration to respond correctly. That means that responses for VIP users will be different that for regular users.

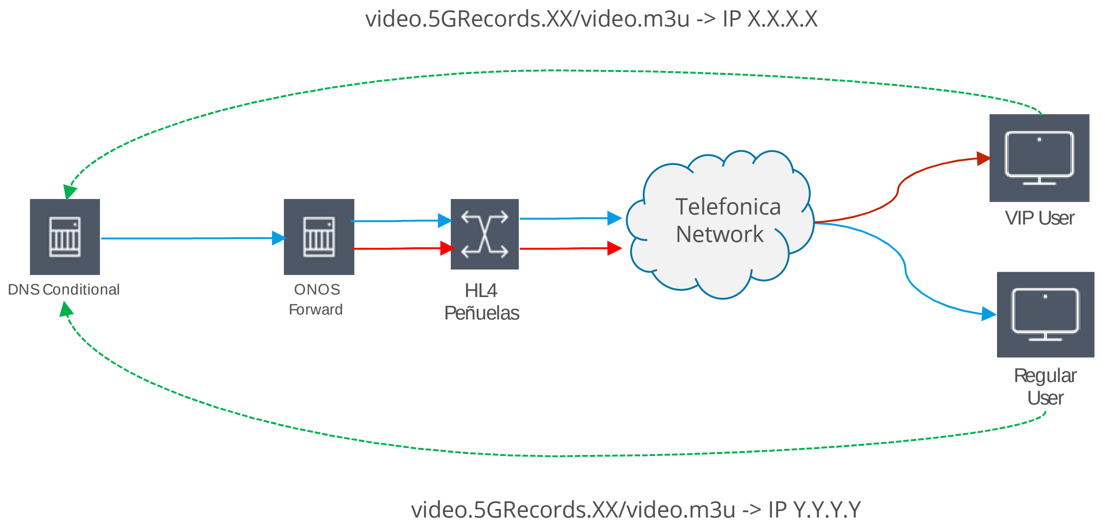


Figure 69: DNS conditional.

Experimental video player

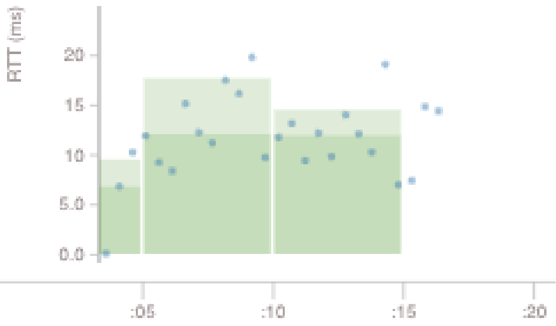
For the playback of the video by the users, a specific video player will be used.

It is a simple React APP that contains a web page for video Testing. It based on OpenSource HTML5 player videojs (https://github.com/videojs/video.js).

The most interesting part is the Plugin (extended from https://github.com/spodlecki/videojs-event-tracking) that reports user QoE in term of quality metrics during the KPI. Specifically, it adds following metrics:

- selectedQuality: the name of the selected video quality in a multi-quality video stream.
- qualityIndex: the index of the selected video quality.
- qualityId: the id associated with the selected video quality.
- width: the width of the selected video quality.
- height: the height of the selected video quality.
- bitrate: the bitrate specified in the selected video quality.

This experimental video player is instrumented: it records video KPIs and sends them to the monitoring infrastructure. Details are described in D4.1. In Figure 70, on top the RTT can be seen in blue dots and the video preview on the bottom.



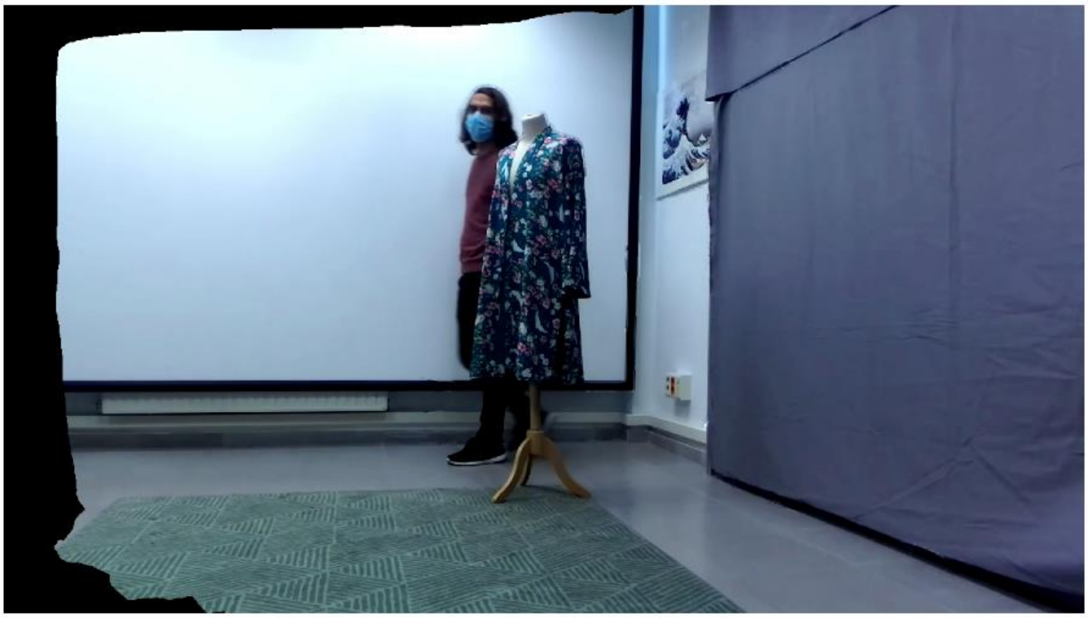


Figure 70: Web Video Player.

5.4 Interfaces

Figure 71 shows the interfaces involved in UC3.

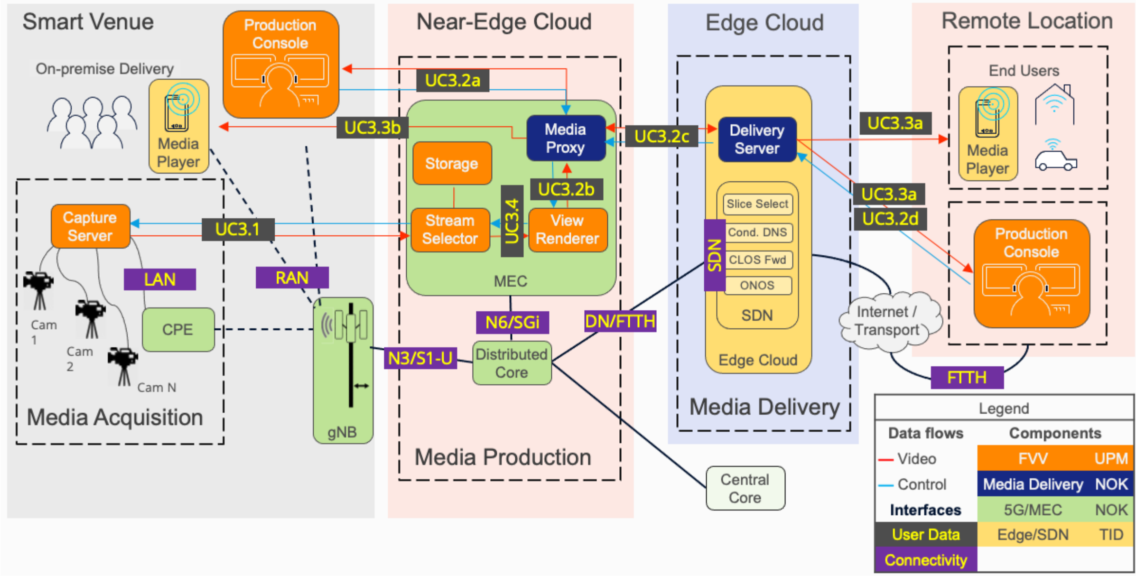


Figure 71: UC3 interfaces.

5.4.1 User Data interfaces

User data interfaces describe the connection between the different elements of the data path. Video streams are shown in red. Control streams are depicted in blue.

5.4.1.1 [UC3.1] Camera contributions

This is the interface between the Capture Server and the View Renderer/Stream Selector. It comprises:

Video (per physical view)

Each camera produces two different streams that are sent in two separate RTP streams, one for the RGB data and the other for the depth data. The texture streams are regular H264 encoded and transmitted over RTP. The depth streams are regular lossless H264 encoded and transmitted over RTP.

Audio

A single monophonic audio RTP stream is sent to from one Capture Server to the Stream Selector. The audio stream is encoded using Opus codec.

Control (per capture server)

These control messages imply two main categories:

- Initialisation control messages (which only occur at the beginning of the session): configuration files for each capture server, computed background on the calibration stage, and synchronisation messages between capture servers.
- Ending control messages (which only occur at the end of the session): Signal that the current session has been finished. This message can be sent by the capture servers to the view renderer, because the total number of frames to capture (configurable parameter) has been reached, or by the view renderer to the capture server because the producer/camera operator has decided to actively finish the session.

The communication between the different modules is performed using UDP-based messages except for the synchronisation messages between capture servers that are MPI messages.

5.4.1.2 [UC3.2] Virtual camera

This is the external interface of the View Renderer, which offers the interface of a virtual camera to the Production Console (UC3.2a). It also provides the contribution quality for the Media Delivery. The Media Proxy (UC3.2b) and Media Delivery (UC3.2c) infrastructure can forward the traffic from this interface to remote producers (UC3.2d).

Video (per virtual view)

Each virtual camera output is sent to the camera operator in a single unicast H.264/RTP/UDP/IP stream.

Audio

A single monophonic audio RTP stream is sent to from the View Renderer to the Production Console.

Control (per virtual view)

The virtual view control message indicates the virtual camera position chosen by the producer/camera operator, from the Production Console to the associated View Renderer. This message is based on UDP containing the camera movement command. The virtual renderer receives the message and synthetises a new virtual view according to the new position.

5.4.1.3 [UC3.3] Content delivery

This is the delivery interface. Each virtual camera is sent using HTTP Adaptive Streaming to the end users, which might be remote (UC3.3a) or local (UC3.3b).

Video (per virtual view)

Each virtual camera is sent to the media delivery in HLS.

5.4.1.4 [UC3.4] Stream selector

This is the interface between the stream selector and the view renderer.

Video (per physical view)

The stream selector receives all the RGB+D streams from the Capture Servers or the Capture Simulator + Storage modules and transmits the subset requested to each view render instance. All the communications are RTP based.

Audio

The Stream Selector receives a single monophonic audio RTP stream from a Capture Server module and transmits it to the View Renderer. All communications are RTP based.

Control (per virtual view)

The virtual view render control data indicates the subset of cameras needed to perform the view synthesis for each frame. These cameras are the nearest actual cameras to the chosen virtual camera position. These messages are based on UDP.

5.4.2 Connectivity interfaces

5.4.2.1 [LAN] CPE interface

A Gigabit-Ethernet physical interface.

5.4.2.2 [RAN] mmWave 5G

5G Radio Access Network protocol stack, either in Standalone (SA) or in Non-Standalone Mode (NSA), in FR2 (mmWave).

Due to the unavailability of SA-compatible UE chipsets for FR2, only NSA will be used during the project

5.4.2.3 [N3/S1-U, N6/SGi] 5G Core interfaces

Standard 5G Core interfaces are used, either in NSA (S1-U, SGi) or in SA (N3, N6) configuration. As already mentioned, only NSA configuration is used in the project.

5.4.2.4 [DN/FTTH] Contribution link

Connectivity between the Media Production Near-Edge cloud infrastructure and the Media Delivery Edge Cloud infrastructure.

Depending on the deployment type, two different options are considered:

- Public Network (Phase 1, “5G Theater”). In Segovia trial location, the Near-Edge Cloud is located in a Data Center in Telefónica network. Connectivity through Telefónica Transport Network is provided using 10GbEth links.
- Non-Public Network (Phase 2, “5G Festival”). In Madrid trial location, the Near-Edge Cloud is included in the Compact 5G Deployment in Nokia premises. Connectivity to Telefónica Transport Network is provided using a 1Gbps FTTH access.

5.4.2.5 [FTTH] Delivery link

Remote users have access to the Delivery Server through a commercial 1 Gbps FTTH access provided by Telefónica. The FTTH access is able to keep DSCP packet signalling, and therefore to support end-to-end slices.

5.4.2.6 [SDN] Software Defined Network interfaces

The flow between the different SDN elements is described in the figure:

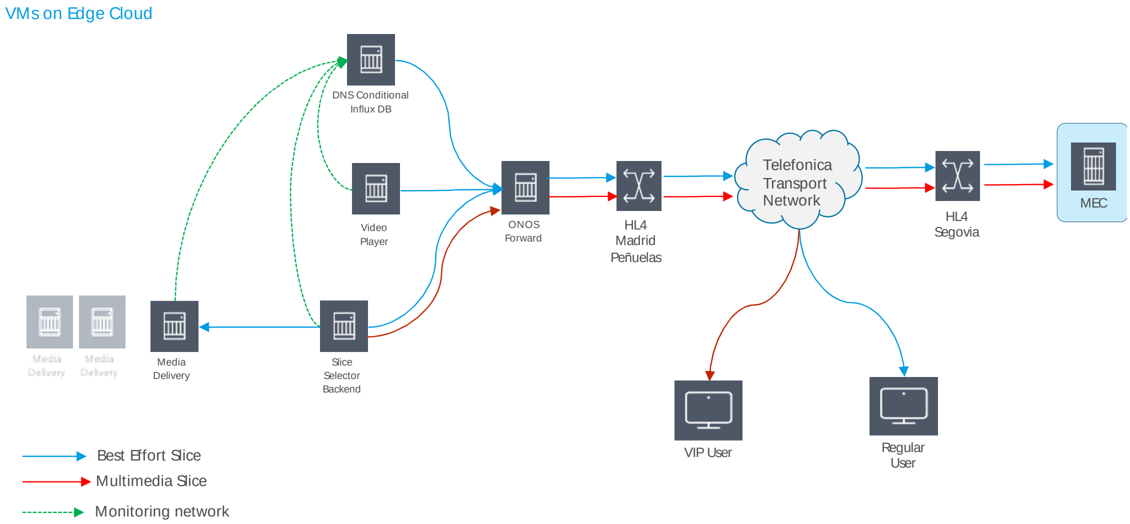


Figure 72: Network management architecture

Interface Orchestration is done thanks to a communication between the Influx DB as monitoring component and two other elements Conditional DNS and slice selector. An alert, previously configured (TBD) will be trigger two actions, change a DNS record inside DNS and internal tables in Slice selector. All the communication is TCP/IP in a dedicated and private monitoring network.

The relation between components is:

- **SDN:** ONOS is used as the SDN controller and it´s connected to all networks.
- **Slice Selector**: Software component based on NGINX servers acting as reverse proxy with capability to redirect request to the correct slice based on the IP accessed and the URL. Combining server instances listening on different public interfaces and different VHOST to segment the traffic along the correct slice. It is connected to slices gold and best effort internal and external, and to the internal video delivery.
- **DNS Conditional** connected to best effort slice. Response will be different based on network status and type of user.
- **Video Player** connected to best effort slice. Only to download the player.
- **Media delivery:** delivery of the video, connected to the internal video network.

6 Multiple UC shared component: Media Operational Control Gateway (MOCG)

6.1 Task purpose

Task T3.4 is concerned with providing a joined-up approach to “Media Orchestration and Control” (MOC) functions that allow effective use of media devices on 5G networks. These functions include:

- Media device registration
- Discovery of media devices
- Connection of media devices
- Configuration and Control of media devices
- Monitoring of media devices
- Synchronisation of media and data flows
- Authentication and authorisation
- Resource management and provisioning

6.2 Architectural Approach

An important activity within T3.4 has been to explore how to access these functions for 5G-RECORDS’s use cases in a way that fits in with the wider production architecture, and the Media Orchestration and Control Gateway is a logical concept to enable this. The following figure shows the MOC Gateway in context of use case 2; the presence of the Media Gateway means this is the most complex use case for the gateway and is therefore where the project has concentrated its effort.

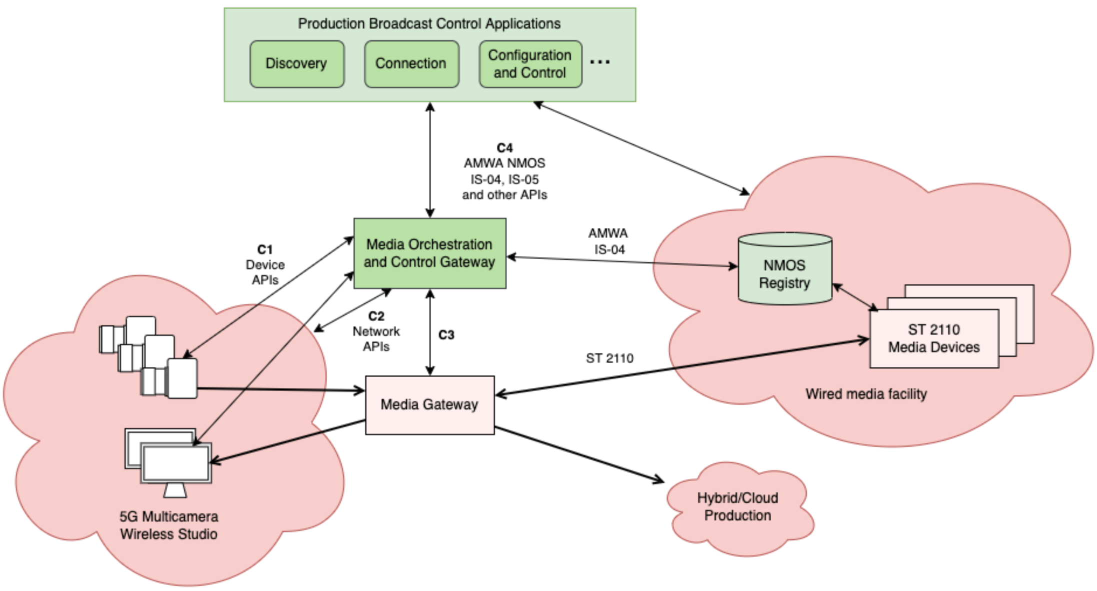


Figure 73: Media Orchestration and Control Gateway interfaces (UC2).

The project has documented many aspects of the four interfaces C1..C4 and developed a prototype implementation of the MOC Gateway to demonstrate the most important functions for UC2, as described in the following sections.

UC1 does not require a Media Gateway so uses a similar but simpler approach as interface C3 is not required.

6.3 Media Orchestration and Control Functions

6.3.1 Registration and Discovery of Media Devices

To make a camera, or other media device, available for use in a production, it first requests access to the 5G network and establishes a data link, after which it will be available to communicate with the MOC Gateway.

For UC2, the camera connects via a Media Gateway to a ST 2110 network. The Joint Task Force on Networked Media recommends (in TR-1001) the use of AMWA IS-04 NMOS Discovery and Registration Specification with ST 2110, and so we desire the camera to appear in the NMOS IS-04 registry. However, it might not be possible, or desirable, for the camera, for the camera to use IS-04 directly:

- Few cameras support IS-04 directly at this time.
- IS-04 has mostly been used with uncompressed video, rather than the compressed stream formats typically used on wireless networks (although, with input from 5G-RECORDS partners, this is starting to change).
- In its current form, IS-04 requires a media device to expose aa Node API. This requires the camera to run an HTTP server, which might not be possibly for some implementations or for some network scenarios or might not be permitted by the network operator.

For such cases, the project has defined an MQTT protocol within C1 to register a media device. “Request” and “response” MQTT topics to provide a general mapping of a REST API to MQTT. This approach can provide a general approach to interface AMWA NMOS APIs where the media device cannot run an NMOS Node API, and instead the gateway runs this, in effect acting as an “NMOS proxy”:

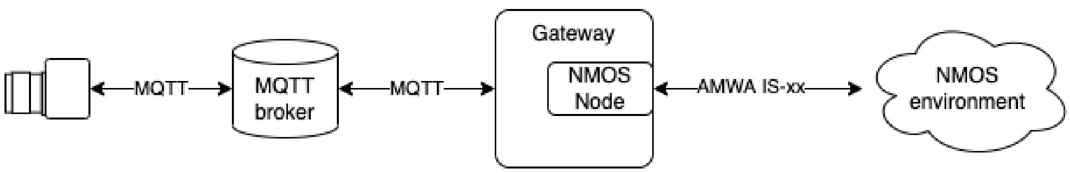


Figure 74: MQTT and NMOS.

For initial testing, BBC and Bisect have implemented a simplified version of the MQTT protocol using Node.js (and the Node-RED flow wiring toolset).

A more detailed description of registration and discovery, and the MOC Gateway’s role in this is presented in D3.1.

6.3.2 Connection of Media Devices

Once the camera or other media device is registered, a control application can connect it to a destination in the media facility, which in simplified form involves the following:

- The Media Gateway runs a pipeline to receive from the camera and send ST 2110.
- The MOC Gateway uses MQTT to communicate with the camera (see above) to communicate the Media Gateway's receive endpoint
- The controller queries the NMOS registry to find information about the camera and the ST 2110 sender on the Media Gateway that is handling the (converted) camera streams, and about the ST 2110 receiver in the wired facility.
- The controller invokes AMWA IS-05 NMOS Connection Management to make a connection between the sender and receiver.

Figure 75 shows two examples of NMOS control applications.

A more detailed description of this sequence and the MOC Gateway and Media Gateway roles in it is presented in D3.1.

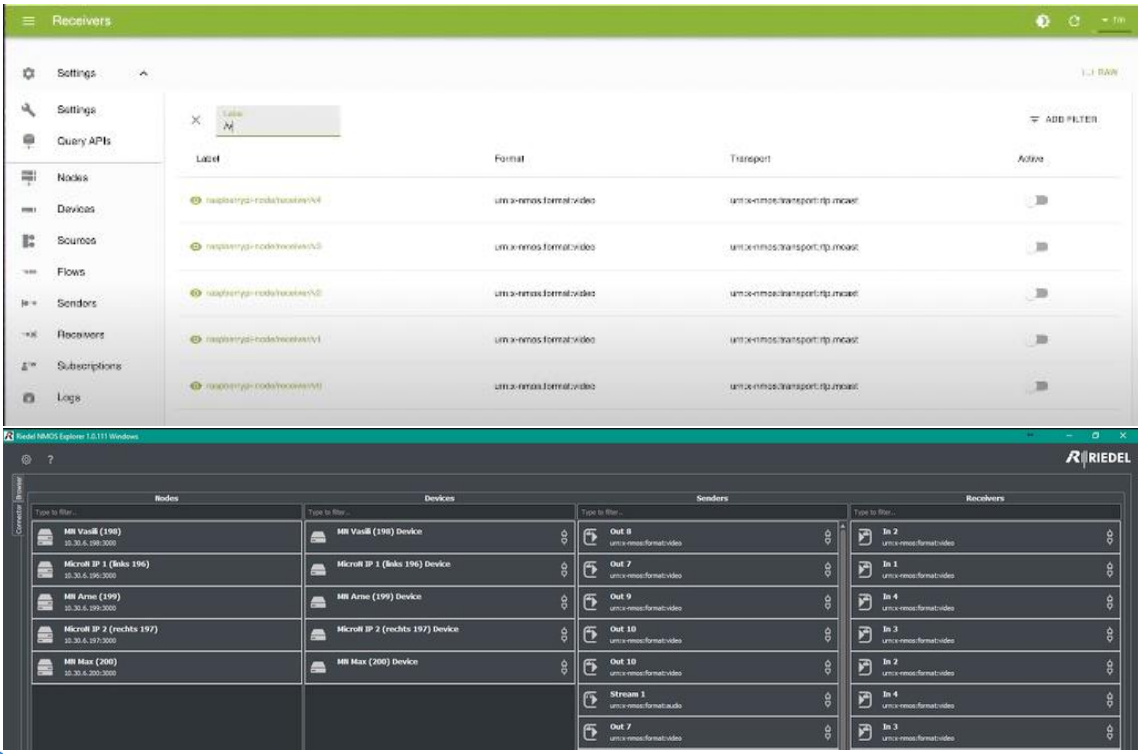


Figure 75: Two examples of NMOS control applications.

6.3.3 Network Control and QoS

The project has investigated requirements for prioritisation of different flows used by the production, based on their sensitivity to latency (see Figure 76 below) and the available APIs (PCF and NEF) to define QoS flows to meet these requirements in 3GPP releases 15, 16 and 17. While the UC2 tests so far have used default QoS settings, the MOC Gateway architecture can be extended to use the APIs to setup suitable QoS flows, as discussed in D3.1.

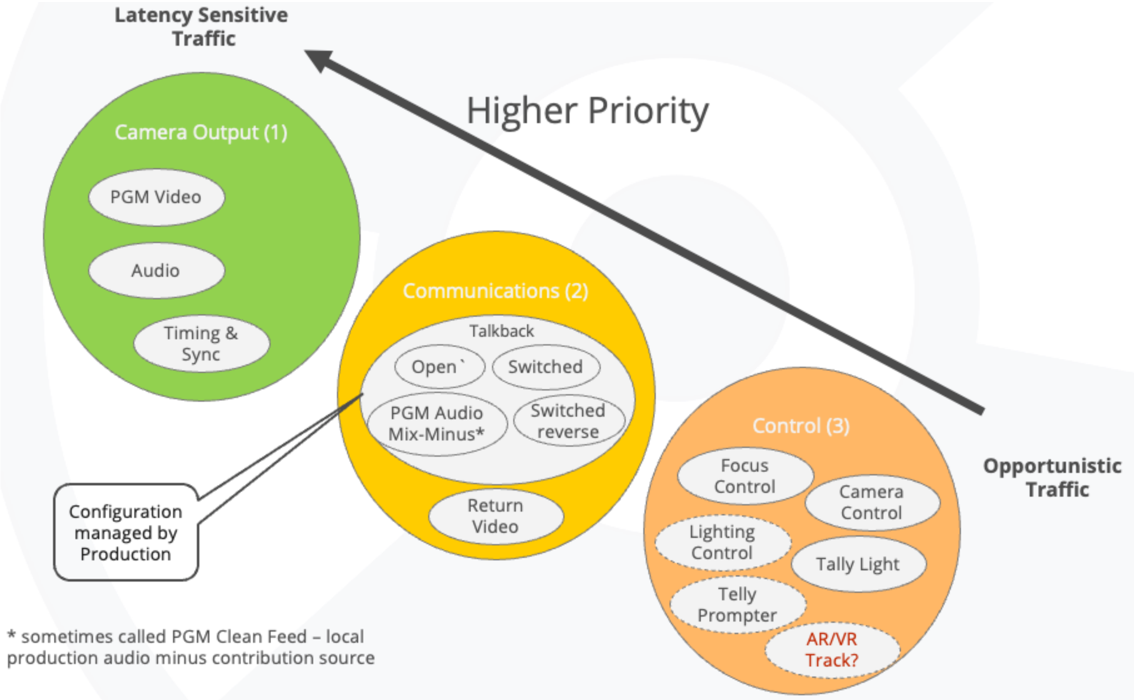


Figure 76: Prioritisation of typical production flows.

6.3.4 Camera Control

The project has investigated a possible solution to Remote Camera Control and has implemented a Proof-of-Concept application to control the camera. Based on what we described in the previous deliverable D3.1 we developed a novel way to send time stamped camera control messages through NMOS IS-07 specification.

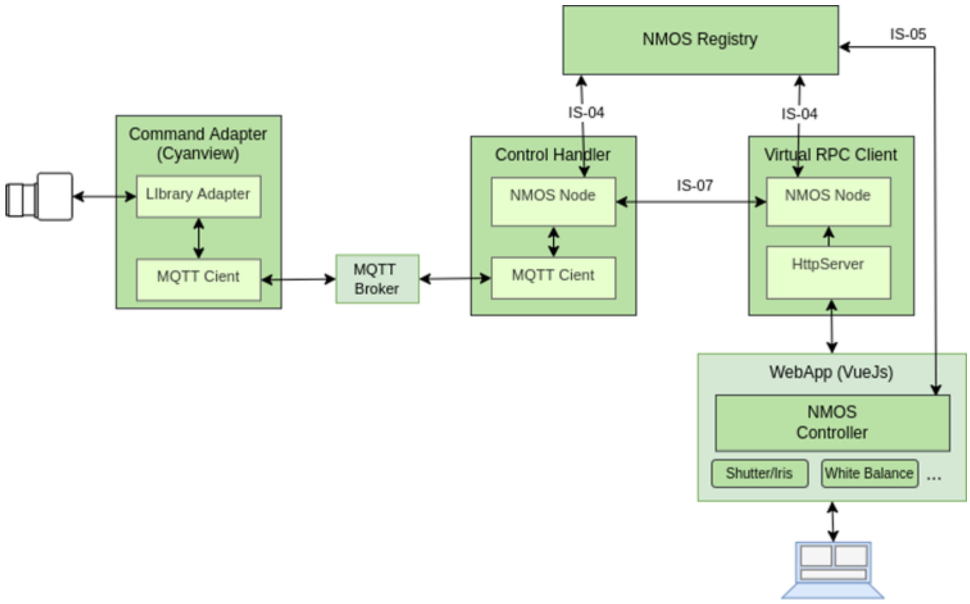


Figure 77: Camera Control proposed architecture.

The two core components developed by RAI team in the proposed architecture are presented in Figure 77 and are:

- Nmos-control-sender: this node has the http endpoints to get the commands from the control WebApp
- Nmos-control-receiver: this node receives the IS-07 messages and translate it into MQTT messages to forward proper commands to Cyanview system

The two nmos nodes has been adapted for our scopes, the baseline implementation used for our development was the Sony NMOS Opensource implementation.

A control WebApp has been developed using VueJS framework, starting from a template available on GitHub called Vuestic. In the dashboard panel it is possible to control basic camera parameter like Exposure and Gain.

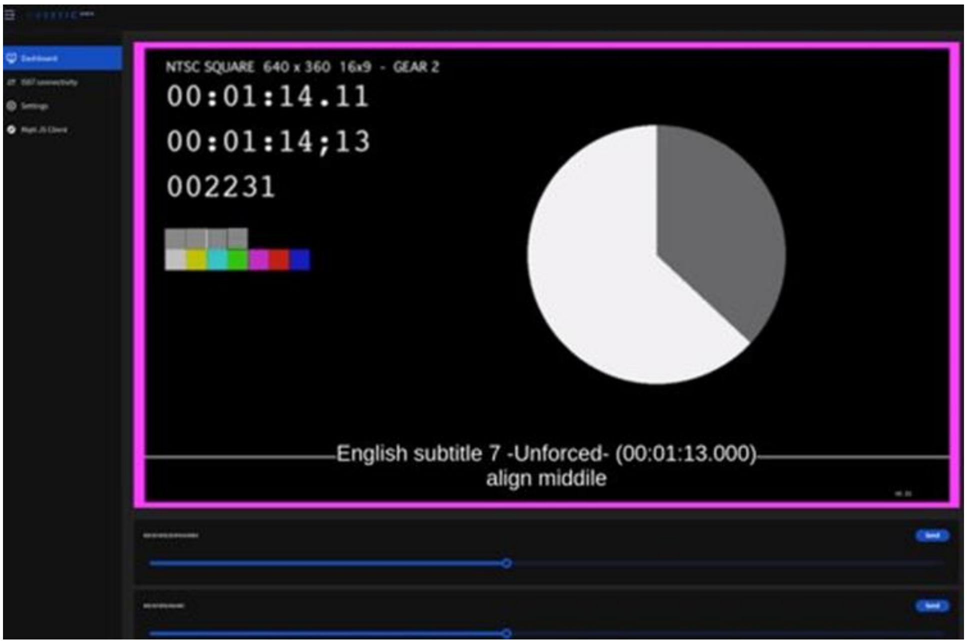


Figure 78: Control dashboard.

A basic NMOS controller has been implemented. With this simple Web NMOS controller it is possible to discover every nmos-control-receiver and nmos-control-sender registered to the NMOS registry present in the local network. It is also possible to connect an IS-07 receiver to an IS-07 compatible sender.

The settings webpage (Figure 79 below) is where it is possible to setup the web application. It is possible to setup the NMOS registry address and the nmos-control-sender HTTP endpoints (i.e., gain 10.45.1.12:9999/gain). Lastly, it is possible to setup also the video stream endpoint, if available, coming from the Media Gateway developed by project partners, allowing to check directly on screen the result of the remote control.

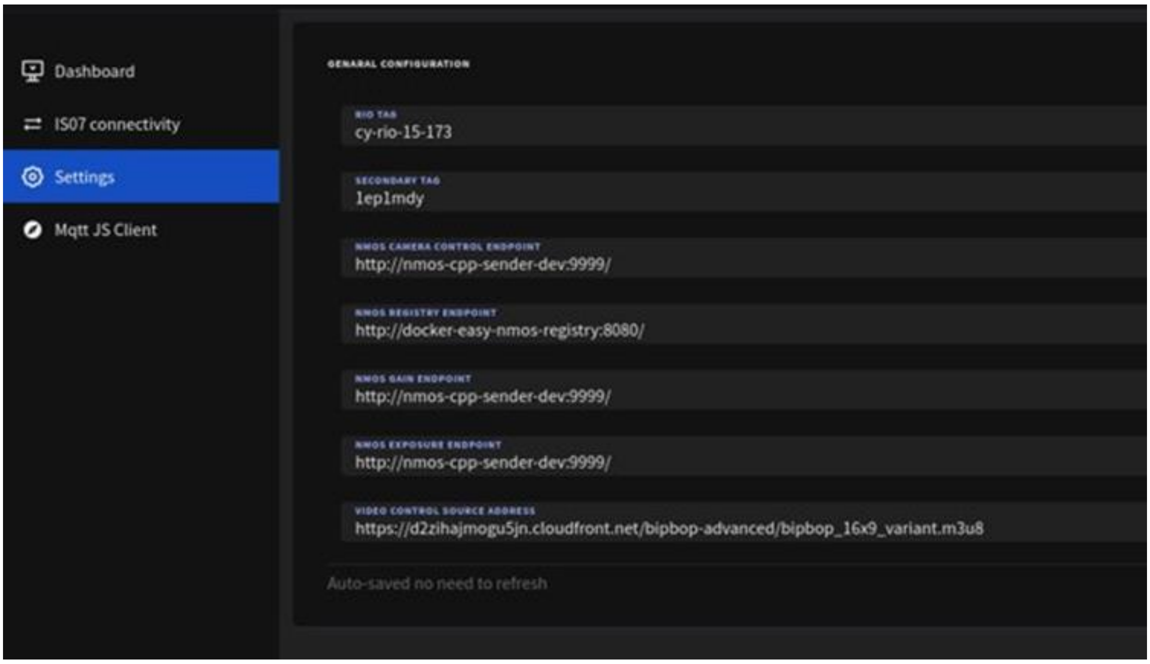


Figure 79: Settings view.

Thanks to Cyanview adapters, it has been possible to control a Sony camera with its legacy protocol. In the Figure 81, the Cyanview RCP (where the MQTT server runs and where it is possible to control the client device) and the RI0 (the blue device near the Sony camera, responsible for the translation of MQTT messages into camera proprietary control protocol) are visible.

Cyanview has defined a specific MQTT topic hierarchy for the operational control purpose. For example, to control a camera tally, a numeric value is published to an MQTT topic such as the following (simplified) example:

/<id>/camhead/set/tally

where <id> uniquely identifies the camera.

The functions developed are packaged as 3 containers present on docker-hub:

- /nmos-cpp-camera-control:sender (Virtual RCP client in the draw)
- /nmos-cpp-camera-control:receiver (Control handler in the draw)
- /camera-control-gui:latest (WebApp (VueJs) in the draw)

Thanks to this modular and containerised design, it is possible to deploy these three containers dynamically into a generic remote orchestrator like K8S or Openshift. In the context of the project an OpenStack cluster has been provided by BBC, where our three developed application runs relying on two Virtual Machines (VM).

Thanks to Wireguard described in the next paragraph it has been possible to test the remote deploy at BBC premises with a Sony camera and Cyanview devices from Turin at Rai CRITS-Labs premises.

6.3.5 Remote Camera Control POC

A Remote Camera Control POC has been developed. The proposed solution uses the developed components to remote control a camera through this NMOS IS-07 novel approach.

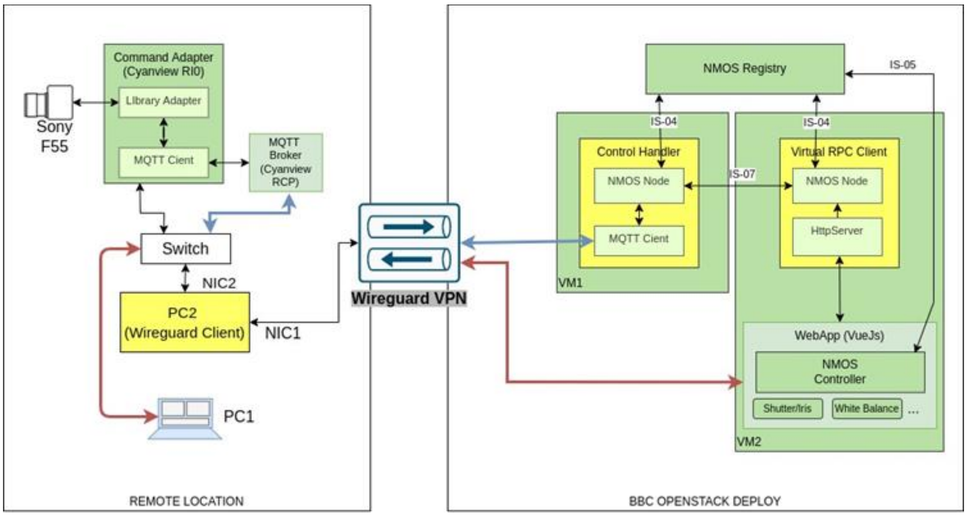


Figure 80: POC architecture.

In the Figure 81 the two VM (VM1 and VM2) deployed in the BBC openstack accessible via Wireguard VPN by using a PC as a client (PC2) can be seen. An unmanaged switch for the local remote connection of the devices has been added for convenience. The setup was completed by another PC (PC1) used as access for the remote-control GUI, the red arrow represents the connection to the WebApp through the browser. The MQTT traffic produced by nmos-cpp-camera-control:receiver is represented by the blue arrow in Figure 81.

The proposed POC deployment is visible in Figure 81. We can see the Sony F55 camera with a viewfinder the two Cyanview devices and the workstation used to route the traffic through the Wireguard VPN (green eth cable).

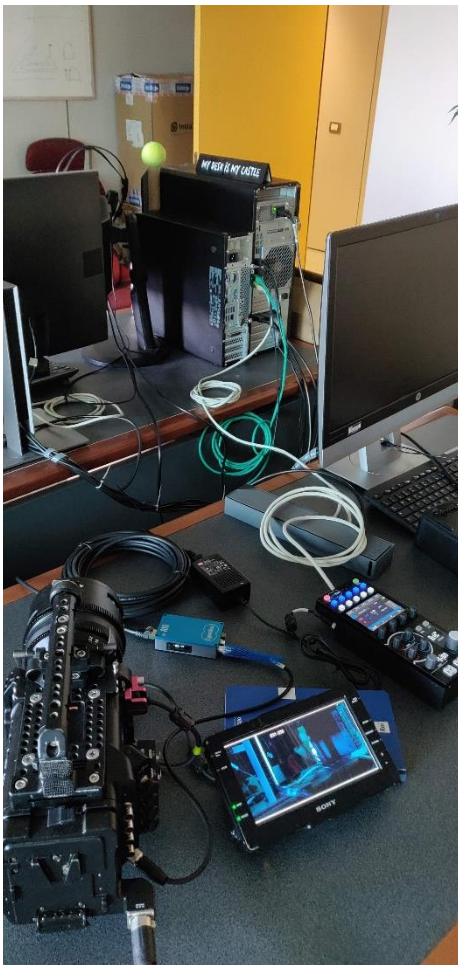


Figure 81: POC deploy at RAI CRITS-Lab.

6.3.6 Authentication and Authorisation

A media device's modem is authorised on the 5G network as part of its initialisation. By their nature, the MOC Gateway and Media Gateway provide a mechanism of controlling access beyond the network, and the MOC Gateway could be extended to support the AMWA IS-10 NMOS Authorisation Specification, so that it could present credentials when registering the media devices with IS-04.

6.3.7 Provisioning, Resource Management and Monitoring

The project has investigated the use of widely used open-source tools for provisioning and configuration of software components used for typical UC2 scenarios:

- BBC R&D provides the project with an NMOS registry, MQTT broker and GStreamer H.265 endpoints hosted in virtual machine instances on its OpenStack cloud at its London site. A WireGuard VPN provides controlled connection to these from project partners' locations. The instances are provisioned using the open-source infrastructure-as-code tool Terraform and the applications and VPN are configured using the open-source automation tool Ansible.

- BBC has also investigated the use of Ansible for installation of software on Jetson Nano and Xavier devices required for streaming and control.

6.4 Interfaces

The MOC Gateway functionality as described in Chapter 6 includes four main interfaces used by its Device Handlers. The section numbers referenced below provide more information , and a more complete description of the interfaces and message sequences is provided in sections 6.3.1 and 6.3.2 of D3.1.

6.4.1 C1: Device Interface

BBC, Bisect and Ericsson have agreed a generic approach to use MQTT to map to NMOS or NMOS-like REST APIs, to enable registration, configuration and connection of cameras and other networked media devices. A simplified version of this approach has been tested by BBC and Bisect. See the sections above for more on the motivation for this. This simple approach could be extended to a more general mapping of the AMWA NMOS IS-04 and IS-05 messages to MQTT.

RAI have extended the MQTT concept to include mapping of common control functions, for example pan, tilt, zoom and iris. This is based on the approach of AMWA NMOS IS-07 for carrying time-sensitive event data as often the time at which camera control actions occur is important.

6.4.2 C2: Network Interface

BBC and Ericsson have outlined the possible use of NEF and PCF to enable QoS flow prioritisation.

6.4.3 C3: Gateway Interface

Bisect have developed a REST HTTP interface to enable the MOC Gateway to instantiate a content handler on the Media Gateway and communicate source and destination address and port information between the MOCG and MG, as part of the connection sequence (described in more detail in section 6.3.2 of D3.1).

6.4.4 C4: Northbound Interface

C4 represents a set of interfaces with production control plane applications and services. For discovery, connection and camera control this uses, and is based on AMWA NMOS IS-04, IS-05 and IS-07 respectively, and the approach can be extended more generally to and other control-plane APIs.

Annex A: FVV capture system calibration

The capture system has been designed to be portable and based on stereo consumer electronics cameras. These cameras are placed at fixed (discrete) points around the scene using variable height stands, aligned to cover the desired field of view, and wired to the capture servers using USB cables. Each capture server manages three cameras and performs the operations described on section 5.3.1: synchronisation, RGB + depth data capture, depth post-processing, encoding and transmission.

To correctly render synthetic views, two offline tasks must be performed previously to the actual capture process: (i) a complete cameras calibration and (ii) background modelling. Both operations are performed using custom software developed specifically for this purpose.

(i) Calibration

Once the system is deployed, to correctly synthesise views, a complete and accurate characterisation of both the internal and external parameters of the reference cameras is necessary to know where the cameras are placed in a common reference system. The internal parameters of each camera are characterised, using the common pinhole camera model, by the intrinsic matrix:

$$A = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

where f_x and f_y describe the focal length, and c_x and c_y the principal point of the camera (all of them in pixel units). The external parameters of the camera are characterised by the extrinsic matrix, which describes the location of the camera in the 3D world and the direction it is pointing to:

$$(R|t)$$

where R is the 3x3 rotation matrix, and t is the 3x1 translation vector.

This set of parameters is necessary to unproject points from the image plane to the 3D space and then project those 3D points back to the image plane of the virtual camera.

To obtain the described parameters, a reference object/pattern calibration technique has been chosen for accuracy. The calibration process is performed by placing a pattern with known geometry in front of the camera. Different 3D or planar patterns can be used to obtain matching points between the images captured by the cameras. For this specific case, a checkerboard (Figure 82) has been chosen as the calibration pattern to ease point detection and obtain multiple matches between a large set of captured images, hence leading to the creation of a dense 3D point cloud.

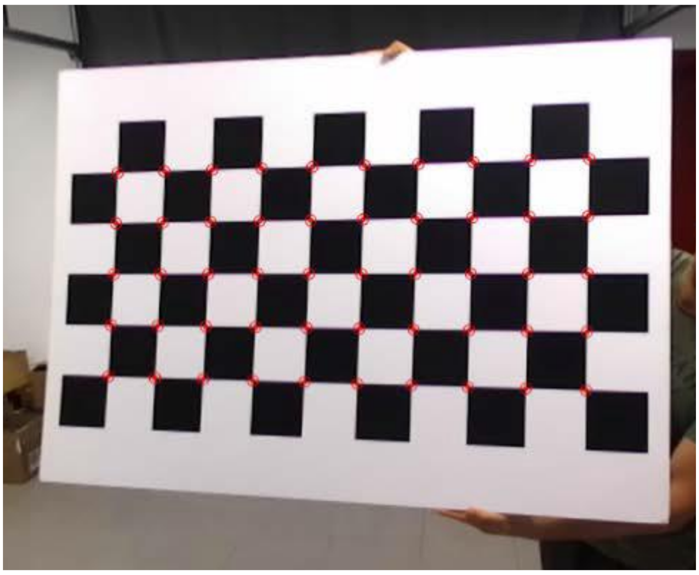


Figure 82: Calibration checkerboard

The first stage of the calibration process implies obtaining a point cloud by detecting the corners of the pattern (red points of Figure 83). The complete checkerboard needs to be viewed by several cameras at the same instant so that geometric constraints remain consistent among all different views. Our checkerboard pattern has 6 rows and 11 columns, and with a square edge of 101.6 mm (it must be that big to fill a significant portion of the image).

The estimation of the initial camera parameters results from computing a Euclidean calibration. 3D points are first organised in equal-sized triplets of aligned points (one triplet per checkerboard column, thus, some detected points are discarded). Figure 83 shows the points (in red) and triplets (in yellow) used in the optimisation step.

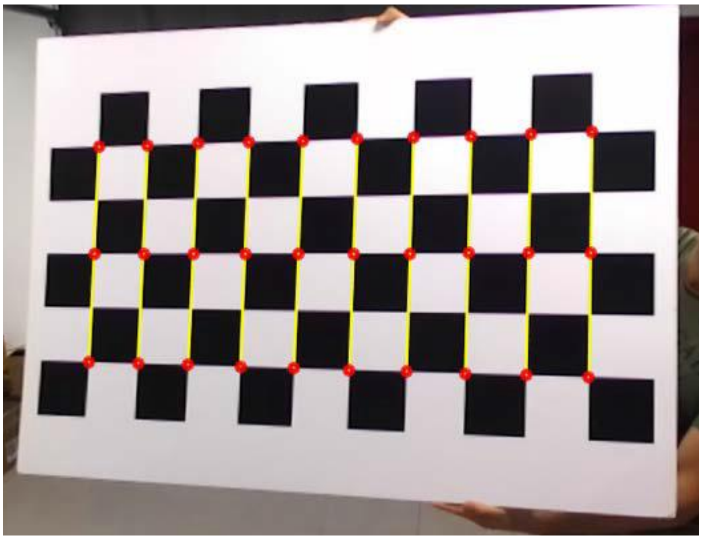


Figure 83: Detected points (red) and triplets (yellow) over the calibration checkerboard

Once the initial parameters have been obtained, a bundle adjustment optimisation (specifically, the partitioned Levenberg-Marquardt) algorithm is used for the minimisation of a cost function that includes the reprojection error and the variance in the triplet lengths. In practice, it is preferable to give more weight to the adjustment of the triplets, to prioritise their correct 3D reconstruction.

Figure 84 shows two examples of results obtained with the calibration procedure, using different arrangements of the cameras. The positions of both the cameras and the triplets are plotted.

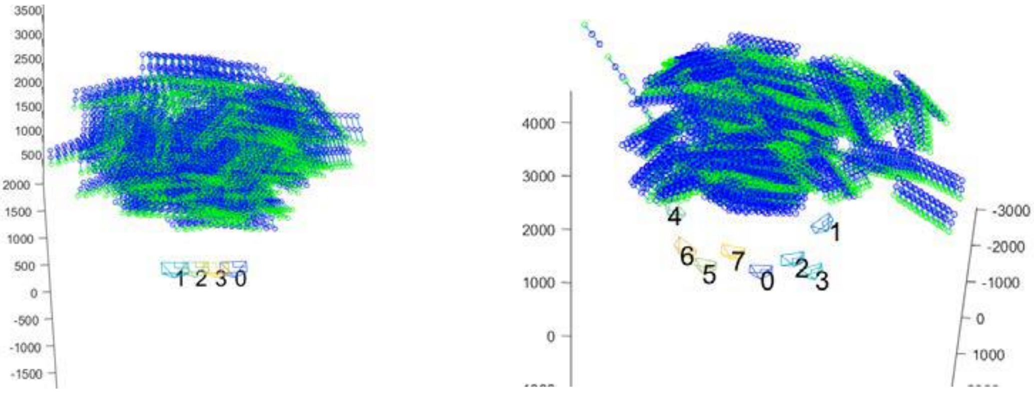


Figure 84: Calibration results example: detected points (green and blue) for different camera arrangements

In addition to the described computation of camera parameters, the calibration procedure allows to refine (to some extent) the depth data provided by the cameras, during the capture process, during the post-processing stage.

(ii) Background modelling

Once the cameras calibration process has been performed, the next step implies a background depth modelling. Assuming a controlled environment, the background of the sequences remains stationary; we take advantage of this to transmit just the foreground (i.e., the part of the image that changes every frame) to reduce data traffic. Therefore, a pre-calculated background model for both colour and depth is needed at the receiver, and a foreground segmentation must be performed at the transmitter in real time.

The creation of the background model is performed offline after the calibration process and the depth correction. Due to the noise inherent to the depth estimation of the cameras, a single/joint depth background model is calculated to have a complete and unique depth image that will provide a stable background for the synthesis. As for the colour background model, it is not unique, since one is created per camera, independently of the rest, by capturing several frames of the background (empty room) and obtaining the mean for each pixel. The colour models are used for both foreground segmentation (performed on-line during the capture process) and depth modelling.

The depth model of the background is acquired by using the structure from motion (SfM) paradigm that permits recovering the 3D structure of the scene by using 2D information from the different captured images and using the calibration parameters previously computed.

The first step is to extract key points from the colour images of the background. To enhance the input images so as to have the feature point extraction find in them more key points, a retinex filter is used. Then, key points are detected, and feature points extracted in the filtered images using the Accelerated-KAZE features extractor. With all the key points that have been detected on the input images, a 3D point cloud can be obtained using a regular SfM pipeline.

However, since the cameras are calibrated, a simpler and faster triangulation algorithm can be used: a point on an image, together with the calibration parameters of the corresponding camera, determine a line in 3D space; key points that are identified to be the same on several images ideally determine pencils of lines with a common 3D point, so the set of such common points would be the 3D point cloud wanted. In practice, due to errors both in the estimation of the camera parameters and the location of the key points, rays do not usually intersect, and the location of each 3D point is chosen such that it minimises the sum of squared distances to the possible candidates.

Direct estimation of the implicit surface of the scene can be obtained because multi-scale retinex with colour restoration and A-KAZE key point extractors are used to build the point cloud, yielding denser point clouds than usual SIFT-based workflows. After an initial estimation of the surface 0, it is refined using a multi-view stereo variational algorithm based on photo-consistency constraints 0, yielding the final mesh (see Figure 85).

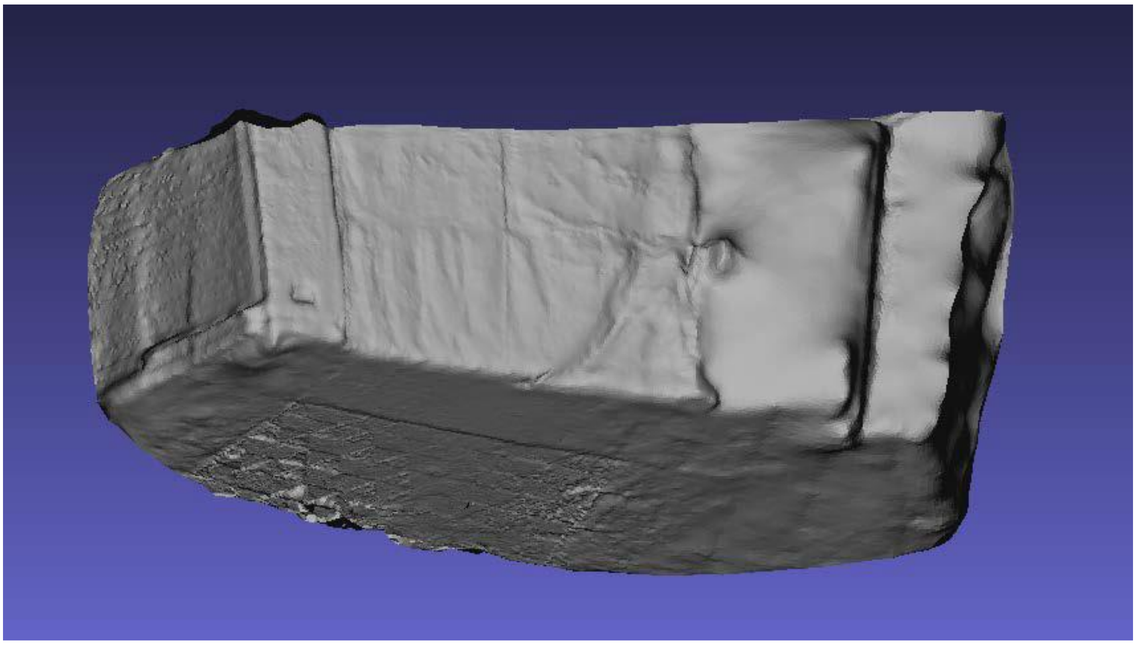


Figure 85: Background mesh example.

Computer graphics pipelines typically include a “hidden surface removal” stage to determine the visibility of objects in a scene, commonly known as Z-buffer. The idea is that all the geometric elements in the scene are projected onto the screen and the closest element to each pixel is recorded, as well as its distance to the camera. A computer graphics rendering engine is used to project the 3D reconstruction of the background model to each of the cameras. Therefore, a depth map of the background is generated for each camera, and these depth maps are used in the edge server for the view synthesis rendering

7 References

[1] 5G-RECORDS project, “D4.1 - Integration of 5G components (phase 1)”.

[2] 5G-RECORDS Project, “Deliverable D2.1 - Use-cases, requirements and KPIs”.

[3] EURECOM, “openair-CN-5G Docker deployment,” 22 03 2022. [Online]. Available: https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_HOME.md. [Accessed 27 05 2022].

[4] 3GPP group, “Technical specification (TS) 23.501 - System architecture for the 5G System (5GS)”.

[5] IEEE, “1588-2019 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems”.

[6] “LU800,” [Online]. Available: https://www.liveu.tv/products/field-units/lu800.

[7] “LU2000-SMPTE,” [Online]. Available: https://www.liveu.tv/products/decoders/lu2000-smpte.

[8] (JT-NM), Joint Task Force on Networked Media, “JT-NM Website,” 4th September 2015. [Online]. Available: https://www.jt-nm.org/reference-architecture.

[9] “OAuth 2.0 Website,” [Online]. Available: https://oauth.net/2/.

[10] AMWA, “AMWA IS-05 NMOS Device Connection Management Specification: Overview,” [Online]. Available: https://specs.amwa.tv/is-05/v1.1/docs/1.0._Overview.html.

[11] AMWA, “Networked Media Open Specifications: Introduction,” [Online]. Available: https://specs.amwa.tv/nmos/.

[12] AMWA, “Networked Media Systems. The big Picture.,” March 2021. [Online]. Available: https://static.amwa.tv/networked-media-systems-big-picture-2021-03-05.pdf.

[13] AMWA, “NMOS. A family of open, free of charge specifications,” [Online]. Available: https://www.amwa.tv/nmos-overview.

[14] Github/BBC , “NMOS Registry Registration API Service,” [Online]. Available: https://github.com/bbc/nmos-registration.

[15] EBU, “Tech 3371 - Minimum User Requirements To Build And Manage An IP-Based Media Facility,” July 2020. [Online]. Available: https://tech.ebu.ch/publications/tech3371.

[16] VSF - Video Services Forum, “SMPTE ST 2110 over WAN,” [Online]. Available: https://vsf.tv/SMPTE_ST_2110_over_WAN.shtml.

[17] AMWA, "NMOS IS-07: Message types," [Online]. Available: https://specs.amwa.tv/is-07/v1.0.1/docs/2.0._Message_types.html.

[18] Github/Sony, "An NMOS C++ Implementation," [Online]. Available: https://github.com/sony/nmos-cpp.

[19] Github/Sony, "An NMOS Client in JavaScript," [Online]. Available: https://github.com/sony/nmos-js.

[20] LiveU , "Tally Light," [Online]. Available: https://www.liveu.tv/ip-solutions/tally-light.

[21] "LU800," [Online]. Available: https://www.liveu.tv/products/field-units/lu800.

[22] Xilinx Inc., "Zynq UltraScale+ MPSoC ZCU106 Evaluation Kit," [Online]. Available: https://www.xilinx.com/products/boards-and-kits/zcu106.html.

[23] Scalagent, "www.scalegent.org," [Online]. Available: http://www.scalagent.com/IMG/pdf/Benchmark_MQTT_servers-v1-1.pdf.

[24] Confluent.io, "https://www.confluent.io/," [Online]. Available: https://www.confluent.io/blog/kafka-fastest-messaging-system/#latency-test.

[25] Muetsch.io, "https://muetsch.io/," [Online]. Available: https://muetsch.io/basic-benchmarks-of-5-different-mqtt-brokers.html.

[26] Flespi.com, "https://flespi.com/," [Online]. Available: https://flespi.com/blog/benchmarking-popular-mqtt-json-implementations.

[27] 3GPP, "TS 23.502 - Procedures for the 5G System (5GS)".

[28] 3GPP, "TS 23.501 - System architecture for the 5G System (5GS)".