H2020-ICT-2016-2
5G-MoNArch
Project No. 761445

5G Mobile Network Architecture

for diverse services, use cases, and applications in 5G and beyond

Deliverable D5.2

Final Report on Testbed Activities and Experimental Evaluation

Contractual Date of Delivery	2019-06-30
Actual Date of Delivery	2019-07-13
Work Package	WP5 - Testbeds
Editor(s)	Jose Enrique Gonzalez, Ignacio Labrador Pavón (ATOS)
Reviewers	Mehrdad Shariat (SRUK), Asterios Mpatziakas (CERTH), Ekkehard Lang (NOK-DE), Paul Arnold (DT)
Dissemination Level	Public
Type	Report
Version	1.0
Total number of pages	133

Abstract

This document is the final public deliverable D5.2 of Work Package 5 (WP5) and reports the work performed by this WP in the 5G-MoNArch project. It explains the testbed activities and the experimental evaluation carried out by the project. The document describes the deployment and the results obtained in the two real-world testbeds that have been used to verify the 5G-MoNArch technology: The Smart Sea Port testbed (deployed in the Port of Hamburg) and the Touristic City testbed (deployed in the Palazzo Madama museum in Turin). These two testbeds are representative for verticals being highly relevant for 5G: the industrial and the touristic sectors. To support the requirements of the respective verticals, multiple different network slices are deployed in each testbed to carry various use cases and applications. For the Smart Sea Port testbed these are: the improvement of the port traffic management, the usage of an AR application to enhance the port operations, and the control of the pollution in the port based on real-time environmental measurements. For the Touristic City testbed, tourists visiting Palazzo Madama get a completely new experience through a rich interactive VR visit to the museum. This document provides detailed information about the implemented use cases, their deployment as well as the experimental evaluation conducted and the main KPIs obtained.

Keywords: 5G, Testbeds, PoC, Demos, Smart Sea Port, Digital Tourism, Flexible Architecture, Network Slicing, Reliability, Resilience, Resource Elasticity, Artificial Intelligence, Security, NFV, Orchestration, Virtual Network Functions, AR, VR, Pollution Control, ENI, MANO

List of Authors

Partner	Name	E-mail
HWDU	Ali Ramadan Karthikeyan Ganesan	ali.ramadan@huawei.com karthikeyan.ganesan@huawei.com
NOK-DE	Peter Rost Lars Christoph Schmelz	peter.rost@nokia-bell-labs.com christoph.schmelz@nokia-bell-labs.com
HPA	Hendrik Roreger	hendrik.roreger@hpa.hamburg.de
SRUK	Mehrdad Shariat	m.shariat@samsung.com
UC3M	Marco Gramaglia Luca Cominardi Ginés García Cristina Márquez	mgramagl@it.uc3m.es luca.cominardi@uc3m.es gigarcia@it.uc3m.es mcmarque@pa.uc3m.es
DT	Markus Breitbach	m.breitbach@telekom.de
TIM	Alessandro Trogolo	alessandro.trogolo@telecomitalia.it
ATOS	Jose Enrique González Ignacio Labrador Pavón Rubén Trapero	josee.gonzalez@atos.net ignacio.labrador@atos.net ruben.trapero@atos.net
CERTH	Thanos Tsakiris Charalabos Georgiadis	atsakir@iti.gr ch.georgiadis@iti.gr
MBCS	Dimitris Tsolkas Odysseas Sekkas	dtsolkas@mobics.gr sekkas@mobics.gr

Executive Summary

This is the final deliverable of work package 5 (WP5) of the 5G-MoNArch project. It reports the activities carried out in the two 5G-MoNArch testbeds during the complete lifetime of the project. The objective of the 5G-MoNArch testbeds has been to serve as test and demonstration platforms to verify and showcase the main technical concepts and innovations of the 5G-MoNArch architecture model, such as: (i) *Network Slicing*, (ii) *Network Orchestration*, (iii) *Network Resilience and Security*, and (iv) *Network Elasticity*. This document presents the main findings and results, and the lessons learned derived from the experimental evaluation conducted in these testbeds.

A major highlight in this project is that developed technical concepts are not (only) implemented in a closed laboratory environment, but in two ***real-world testbeds*** deployed in two operational environments: The Port of Hamburg (Germany's largest sea port) and Palazzo Madama in Turin (an important touristic location in the Italian city). We call these testbeds the Smart Sea Port and the Touristic City testbeds, respectively. These deployments have been possible thanks to the involvement of ***real verticals*** (the Hamburg Port Authority and the Municipality of Turin together with the Fondazione Torino Musei), which represent actual potential customers of the 5G technology.

Each testbed has been used to showcase different innovations. The Smart Sea Port testbed is mainly devoted to evaluating the network resilience and security concepts, while the Touristic testbed is mainly devoted to evaluating the network elasticity concept and the orchestration techniques. To show these concepts, each testbed deploys different ***use cases*** and makes use of the ***network slicing*** to support these use cases in an isolated and controlled way. Within the Hamburg Smart Sea Port testbed, network slicing is applied in an actual industrial deployment to support different applications. In particular, three different properly isolated slices are deployed to support the following use cases relevant for the port area: (i) the *traffic lights management*, to improve the traffic flow for trucks within the port area; this is a service with high reliability and guaranteed latency requirements, and needs a tailored slice to this end, (ii) *interactive AR techniques* to support certain port operations using an eMBB service slice, and (iii) an *IoT service* to measure and control the pollution in the port with sensors installed on moving barges; exemplary for MTC, this service has high reliability and security requirements.

Within the Touristic City testbed, one of the main goals consists in showcasing the feasibility of using an "elastic" management and orchestration system based on Artificial Intelligence (AI) in order to smoothly adapt the resources allocation and usage. To this end, the following two slices have been deployed to provide the visitors an interactive VR visit of the Madama Reale chamber, one of the most impressive rooms in Palazzo Madama: (i) an *eMBB network slice* that serves 360° videos to a VR headset, and (ii) an *URLLC network slice* to handle all the other client-server communications (VoIP, real-time multi-user interaction or the 3D model's movement control).

To implement the required network slicing functionalities different sets of equipment and resources have been used in each testbed, based on developments provided by the manufacturers participating in the project. It is particularly worth highlighting that this equipment comprises (i) ***prototypes based on commercial equipment***, and hence developments that are potentially close to the market, and (ii) more ***advanced research prototypes*** with innovative features. More specifically:

- In Hamburg, the testbed comprises two base stations that cover the sea port area almost completely based on commercial equipment. Two different terminal types are used, the first based on Software Defined Radio (SDR) platforms enabling multi-connectivity and data duplication, and the second based on small-form-factor PCs providing multi-slice capable devices. In both cases commercial industrial antennas are used. For the data centres hosting the core network, commercial servers have been deployed to a local (edge cloud) and a remote (central cloud) commercial data centre of Deutsche Telekom. A specific network slice administration and management tool has been developed for the testbed.
- In Turin, the physical layer of the testbed is an SDR based system with reconfigurable parameters implementing a 5G radio interface based on a PHY/MAC and higher layers in compliance with the 3GPP standard. The Baseband Unit (BBU) has been implemented on an Intel x86 platform, and the radio interface uses various 5G New Radio (NR) functionalities aligned with NR Rel.15. The testbed also includes a set of Physical Network Functions (PNFs)

implementing the radio lower layers and a cloud infrastructure. The eMBB and URLLC slices use different Bandwidth Parts (BWPs) aiming at satisfying the different requirements for each slice in terms of latency and reliability. The orchestration of the VNFs is based on Open Source MANO (OSM) using OpenStack as Virtual Infrastructure Manager (VIM). A specific GUI has been developed to access the relevant service metrics.

These testbeds have served to (i) demonstrate the main technical concepts and innovations of 5G-MoNARch, (ii) provide useful and realistic use cases in real-world environments, and (iii) evaluate experimentally the performance of the provided network services. Among the ***measured KPIs*** and lessons learned from the testbeds, we can highlight the following:

- The Service Creation Time KPI has been reduced below 10 minutes in both testbeds, thus reaching the 5G-PPP target for this important KPI.
- The usage of the Network Slicing paradigm is successfully applied in both testbeds, and it has been shown that isolation between slides is effectively provided.
- The latency values have been reduced to the range of a few milliseconds: less than 5 ms for the Touristic City testbed and about 15 ms for the multi-connectivity setup in the Smart Sea Port testbed, in cells of up to 10 Km.
- The Smart Sea Port testbed has demonstrated how end-to-end reliability can be achieved based on the multi-connectivity concept.
- The Touristic City testbed has demonstrated how innovative AI algorithms can be actually applied to the VNFs orchestration based on the ETSI ENI approach.
- The Touristic City testbed has shown an innovative VNFs orchestration approach based on the VNFs "context" migration that is able to perform the live relocation of VNFs within very few milliseconds and avoid service disruptions.
- A Security model based on Security Trust Zones concept has been designed in the context of the Smart Sea Port testbed.

Both testbeds have been made available to real users and customers in their respective contexts: The Palazzo Madama visitors and the Hamburg port authority.

Intermediate versions of both testbeds have already received recognition in relevant public forums. The Smart Sea Port testbed was recognised with the ***GSMA Global Mobile Award (GLOMO)*** in the category "5G Industry Partnerships" at the 2019 Mobile World Congress [Mwc19], confirming the high relevance of the testbed as such and the implemented use cases for the launch of 5G mobile networks, and the excellence of the technology deployed within the testbed. The Touristic City testbed has been selected as ***Proof of Concept (PoC) for the ETSI*** Experiential Network Intelligence (ENI) ISG [Eni19] due its very high innovation potential and for the availability of different AI algorithms in the proposed management and orchestration mechanisms, contributing to the so-called "Elastic Resource Management and Orchestration" use case within ETSI ENI.

Acronyms and Abbreviations

2D	2-Dimensions
3D	3-Dimensions
3GPP	3rd Generation Partnership Project
5G	5th Generation mobile wireless communication system
5G-MoNArch	5G Mobile Network Architecture
AI	Artificial Intelligence
AIS	Automatic Identification System
API	Application Programming Interface
AR	Augmented Reality
BBU	Baseband Unit
BSS	Business Support System
CAD	Computer-Aided Design
CAPEX	CAPital Expenditure
CDF	Cumulative Distribution Functions
CN	Core Network
COTS	Common-Off-The-Shelf
CP	Control Plane
CP-OFDM	Cyclic Prefix Orthogonal Frequency-Division Multiplexing
CSS	Cascading Style Sheets
DL	Downlink
DoW	Document of Work
DT	Deutsche Telekom
E2E	End-to-End
eMBB	Enhanced Mobile Broadband
ENI	Experiential Network Intelligence
ESU	Environmental Sensory Unit
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
FM	Fault Management
gNB	5G base station
GPS	Global Positioning System
GST	Generic Slice Template
GUI	Graphical User Interface
GW	Gateway
HLS	HTTP Live Streaming
HMD	Head-Mounted Display
HPA	Hamburg Port Authority
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
HW	Hardware
IK	Inverse Kinematics
IoT	Internet of Things
IP	Internet Protocol
ISC	Intra-Slice Controller
ITS	Intelligent Transport System
KPI	Key Performance Indicator

LCM	Lifecycle Management
LTE	Long Term Evolution
M&O	Management and Orchestration
MAC	Media Access Control
MANO	Management and Orchestration
MEC	Mobile Edge Computing
ML	Machine Learning
MME	Mobility Management Entity
mMTC	Massive Machine Type Commination
MNO	Mobile Network Operator
NAS	Non-Access Stratum
NF	Network Function
NFV	NF Virtualisation
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NGMN	Next Generation Mobile Networks
NR	New Radio
NSI	Network Slice Instance
NSSI	Network Slice Subset Instance
NSMF	Network Slice Management Function
OSM	Open Source MANO
PBR	Physically Based Rendering
PC	Personal Computer
PCF	Policy Control Function
PDCP	Packet Data Convergence Protocol
PDU	Packet Data Unit
PHY	Physical Layer
PLMN	Public Land Mobile Networks
PNF	Physical NF
PS	Protocol Stack
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
RLC	Radio Link Control
RPC	Remote Procedure Call
RRC	Radio Resource Control
RRM	Radio Resource Management
RSRP	Reference Signals Received Power
RSRQ	Reference Signal Received Quality
RTMP	Real-Time Messaging Protocol
RTP	Real-Time Protocol
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
RUDP	Reliable UDP
SDN	Software Defined Networks
SDR	Software Defined Radio

SLA	Service Level Agreement
SMm	Security Monitoring Manager
STZ	Security Trust Zone
STZm	STZ Manager
SthD	Security Threat Detector
SthP	Security Threat Prevention
SthR	Security Threat Reaction
SW	Software
ThIntEx	Threat Intelligent Exchange
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TN	Transport Network
TOSCA	Topology and Orchestration Specification for Cloud Applications
TTI	Transmission Time Interval
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
UP	User Plane
UPF	User Plane Function
UPNF	User Plane (UP) Network Function
URL	Uniform Resource Locator
URLLC	Ultra-Reliable Low Latency Communication
USRP	Universal Software-Defined Radio Platform
UV	UV mapping ("U" and "V" are used to describe the horizontal and vertical axes of a 2D texture)
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtual NF
VNFM	VNF Manager
VoIP	Voice over IP
VPN	Virtual Private Network
VR	Virtual Reality
WLAN	Wireless Local Area Network
WP	Work Package
XSThIntEx	Cross-Slice Threat Intelligence Exchange

Table of Contents

List of Figures

List of Tables

1 Introduction

The main objective of this document is to provide a final report on the testbed activities and the experimental evaluation performed in this 5G Mobile Network Architecture (5G-MoNArch) project. Along with the testbed results, this deliverable reports the outcome of the work performed in the Work Package 5 (WP5). It is the continuation of the previous Deliverable D5.1 [5GM18-D51], which described the initial work conducted by WP5 and explained the testbeds setup and the main 5G-MoNArch technical concepts demonstrated on them.

The main objectives of the testbeds developed by 5G-MoNArch are described below:

1. Implement the functionality developed by the project in a real-life platform, in order to bring the concepts of the project into a practical setting and show that they are technically feasible and provide the foreseen functionality.
2. Confirm that the project functionality provides the desired behaviour, showing and evaluating the features of the developed prototypes.
3. Evaluating the Key Performance Indicators (KPIs) corresponding to the performance of the system, measuring them experimentally and contrasting them against the targets identified at the beginning of the project.
4. Involving the customers and end-users of the technology in order to show them the benefits they can get from it, getting their feedback and their level of satisfaction with the results of the project.

The present document has been structured in line with these objectives, reporting on the results that the project has achieved in the various fronts. In order to address the above objectives, 5G-MoNArch has developed two different testbeds, located in two different geographical locations and targeting different real-world use cases:

- The Smart Sea Port testbed, deployed in the Port of Hamburg (the largest sea port in Germany) and hosted by the Hamburg Port Authority [Ham19b] [Hpa19].
- The Touristic City testbed, deployed in the Palazzo Madama (one of the most relevant art museums in Turin) and hosted by the Municipality of Turin and the Fondazione Torino Musei [Pal19] [Fon19].

The above testbeds have been selected because they are representative of two of the most relevant use cases for 5G: an industrial environment (requiring high reliability requirements) and a touristic one (targeting end-users with media-entertainment devices). In the deployment of these testbeds, 5G-MoNArch has involved ***vertical partners*** playing the role of the customers of the technology; namely, the Hamburg Port authority of Hamburg and the municipality of Torino (along with the Fondazione Torino Musei). Their participation has been essential in order to gather their requirements and evaluate their level of satisfaction with the provided 5G services.

Besides involving real verticals, one of the highlights of the testbeds deployed by 5G-MoNArch is that they are not closed laboratory demonstrators, but rather ***real-world deployments*** involving the real operation in the respective environments. Thus, although testbeds are limited in scale by the nature of the project activity, they offer a good perspective on the validity and potential of these concepts in larger scale environments. Indeed, the Smart Sea Port has been employed for the ***real operation*** of the port in various fronts, while the Touristic City testbed has been deployed in a museum and offered to ***real visitors*** in the museum. The involvement of relevant verticals and end-users and the use in an operational environment make 5G-MoNArch testbeds very well aligned with the goals of the 5G-PPP phase 2, meeting and even exceeding the ***ambition*** of Phase 2 projects and being quite close to what is expected for the Phase 3 projects.

The use cases deployed by the testbeds have been carefully selected to satisfy the requirements of the respective verticals while showing the underlying functionality developed by 5G-MoNArch to support such requirements. To this end, in the Smart Sea Port testbed three different use cases have been deployed, requiring three different 5G network slices:

- *Traffic light control and management*: This involves the control of a traffic light in the Hamburg port with a dedicated 5G network slice. The purpose is to showcase the possibility of extending the control of all the traffic lights in the port using the 5G wireless technology in order to replace the system currently in use, which is about 30 years old and mainly based on a fixed network infrastructure performed by an analogue cooper-based technology.

- *Augmented Reality (AR) techniques*: To support certain port operations, AR can be very effective. By this, experts located in a central office can assist on-site maintenance personnel in distant port areas remotely and give support in certain specialised tasks. This is quite important in a wide-area industrial environment such as the Hamburg Port (the port covers an area of about 80 km^2).
- *Pollution measurement platform*: This is essential to support the pollution reduction in the port. It is based on an Internet-of-Things (IoT) application which collects measurements from mobile environmental sensors installed on certain ships in the port, which are connected through a specific network slice of the 5G mobile network.

The Touristic City testbed has focused on a use case involving an *interactive VR visit* to the Madama Reale chamber in Palazzo Madama. In the visit, users can interact with a virtual tourist guide, which assists and provides them with relevant information. Users also perform certain VR-enabled activities, such as instructional games or thematic tutorials. This involves the deployment of two dedicated slices, one with stringent latency requirements for the operations that involve a high level of interactivity, and another one with more relaxed requirements for the remaining operations.

The above use cases and applications serve to demonstrate the different innovations and technical concepts linked to the 5G-MoNArch network architecture. The main technical concepts evaluated by means of the testbeds include (i) Network Slicing, (ii) Network Orchestration, (iii) Network Resilience, Reliability and Security, and (iv) Network Elasticity. To demonstrate these concepts, the Smart Sea Port testbed is used to show how the network slicing concept is actually applied to industrial environments to instantiate isolated network slices that can provide reliable and resilient services, even on failure conditions. The Touristic City testbed shows how the elasticity concept is applied to media and entertainment use cases, allowing the network to efficiently support high-quality services while adapting to workload changes and dynamically allocating the necessary network resources through the appropriate orchestration functions.

The two project testbeds have been presented in different public workshops, which were very successful and attracted not only numerous audiences, but also representatives of very relevant stakeholders from the 5G ecosystem, including vertical customers, end-users and application developers, among others. These events proved to be very useful to gather feedback from the different stakeholders and provided the project with very valuable information on the views of very relevant players for the project's technology. The Smart Sea Port testbed was presented on June 5th 2019 in a final dissemination event in Hamburg, including the presentation of the core research and innovation results as well as a live experience showcasing 5G-MoNArch technical solutions. The Touristic City testbed was made available to all visitors in the Palazzo Madama museum during the May 23rd and 24th, offering the VR experience to real visitors together with a detailed description of the 5G-MoNArch technical solution and innovations.

In this document, we provide complete information about the two testbeds, including details for the different use cases, hardware (HW) and software (SW) components, specific implementations, the set-up and deployment of the testbeds and the experimental evaluation focusing on the main project KPIs. The rest of this document is structured as follows:

Chapter 2 provides background information about the main 5G-MoNArch technical concepts, reviewing the overall architecture and its key enablers, and identifying those enablers and innovations implemented and evaluated in the WP5 testbeds.

Chapter 3 and Chapter 4 are the core part of this document: they give detailed information of the Smart Sea Port and the Touristic City testbeds, respectively. Both chapters follow the same structure, providing: (i) a detailed description of the implemented use cases, (ii) the implementation details of the various functionalities, (iii) the set-up and deployment details, and (iii) the experimental evaluation, measuring and analysing the relevant KPIs for each testbed.

Chapter 5 provides a summary of the experimental evaluation showing the main KPIs in a tabular form that compares, for each KPI, the target result expected by 5G-MoNArch against the result achieved by the testbed. This provides a global view of the outcome resulting from the testbed activities within 5G-MoNArch.

Finally, Chapter 6 summarises the most relevant facts and gives the main conclusions, and some pictures about the project public workshops are provided in the Annex.

2 Technical concepts

In this chapter we review the overall technical concept of the 5G-MoNArch project, consisting of the overall architecture and its key enablers and identifying the enablers and innovations that are being implemented and evaluated in the 5G-MoNArch testbeds. In particular we review the architectural concepts proposed in WP2 [5GM19-D23], WP3 [5GM19-D32] and WP4 [5GM19-D42], and then see how we map these concepts into the two testbeds deployed in the project (Touristic City testbed and Smart Port testbed). Note that, while the 5G-MoNArch concepts have been elaborated in WPs 2, 3 and 4, and are explained in detail in the corresponding deliverables, in the following we provide a brief summary of those concepts with the goals of (i) making this document self-contained, and (ii) see how these concepts are mapped in both testbeds.

Figure 2-1 shows a schematic view of the overall 5G-MoNArch approach, depicting the instantiation of the so-called "Enabling Innovations" and the common mobile network architecture (developed in WP2), and the deployment of "Functional Innovations" on resilience, security and resource elasticity respectively (developed in WP3 and WP4). The figure furthermore shows the interrelation between the verification and validation tasks (conducted in WP6) and testbeds, where the testbeds provide operational results in particular for the KPI verification.

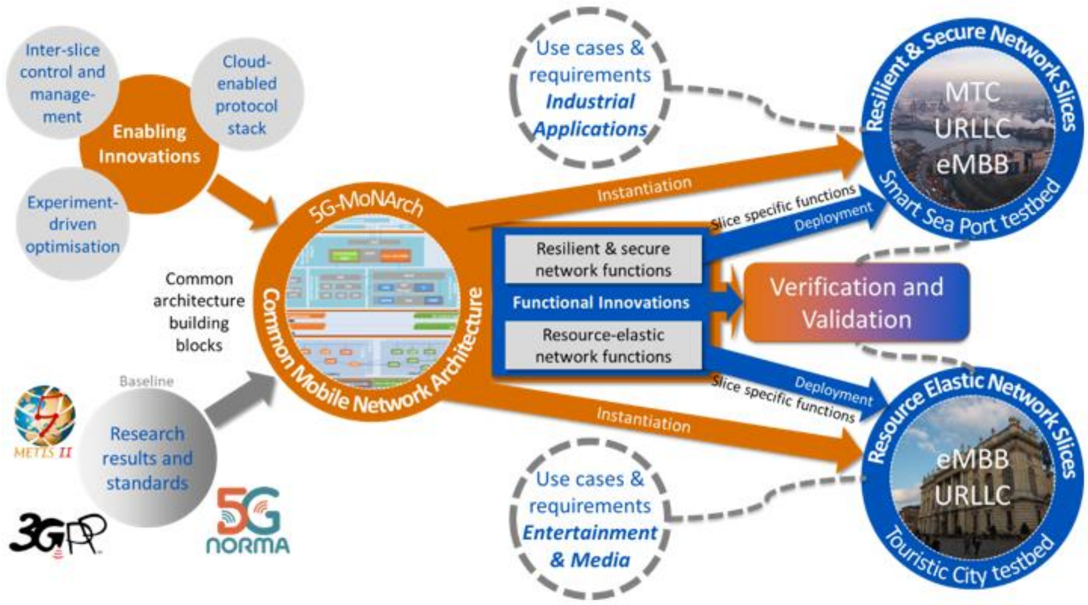


Figure 2-1: Schematic view of the 5G-MoNArch approach

Figure 2-2 depicts in more detail the 5G-MoNArch common mobile network architecture, including the extensions for resilience and security and for resource elasticity. The main components of this 5G-MoNArch architecture are the following:

- The **Service layer** functions interact with the Management and Orchestration (M&O) layer via the Communication Service Management Function (CSMF).
- The **Management & Orchestration layer** is composed by the M&O functions from different network, technology, and administration domains, e.g., the 3rd Generation Partnership Project (3GPP) public mobile network management, the European Telecommunications Standards Institute (ETSI) Network Functions (NF) Virtualisation (NFV) Management and Orchestration (MANO) reference [ETSI14-GSNFV001], the ETSI Multi-access Edge Computing functions [ETSI16-MEC], management functions of Transport Networks (TNs) and non-public enterprise networks. Further, the M&O layer comprises the End-to-End (E2E) M&O sublayer hosting the Network Slice Management Function (NSMF) and CSMF managing the network slices and the communications services, respectively, across multiple management domains in a seamless manner. In the so-called *Virtualisation MANO* domain,

the ETSI NFV MANO architecture for the Life-cycle Management (LCM) of Virtual Machines (VMs) is extended towards LCM of virtualisation containers (e.g., Docker) The management layer (as described in [5GM19-D23]) includes all the novel component defined by the 3GPP standard, enriched with the needed modules and interfaces needed to support the features required by the two testbeds, as described in Chapters 3 and 4.

- The **Controller layer** realises the software-defined networking concepts [ONF14], extends them to mobile networks, and therefore accommodates two controller types: (i) the Cross-Slice Controller (XSC), e.g., a Radio Access Network (RAN) controller for the control of Cross-slice network functions (XNFs) shared by multiple network slices, and (ii) the Intra-Slice Controller (ISC), e.g., a Core Network (CN) controller for Intra-Slice Network Functions (INFs) within a dedicated CN-Network Slice Subset Instance (NSSI). These controllers expose a northbound interface (NBI in the figure) towards control applications and a southbound interface (SoBI) towards Virtual and Physical Network Functions (VNFs and PNFs) in the Network layer.
- The **Network layer** comprises the VNFs and PNFs for both Control Plane (CP) (i.e., cVNF and cPNF) and UP (i.e., uVNF and User Plane Network Function (UPNF). NFs can include, for example, 3GPP CP functions and User Plane Functions (UPFs) (e.g., UPF or PDCP) or novel NFs developed in the project (e.g. for resource elasticity, resilience, and security).

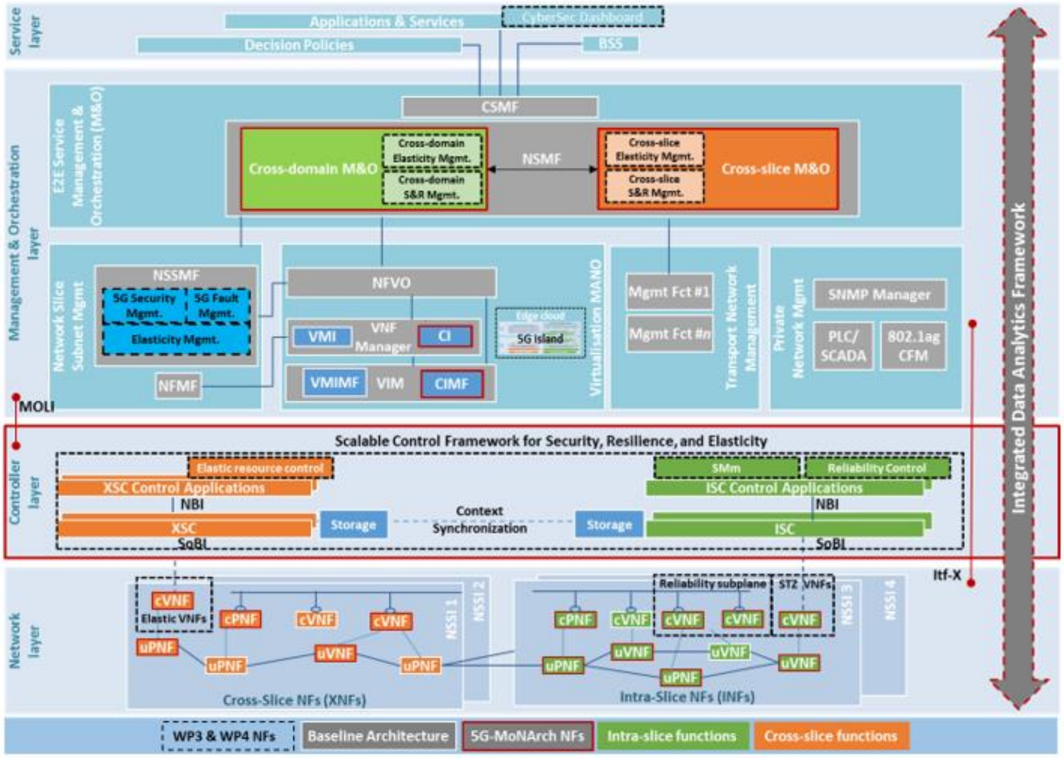


Figure 2-2: 5G-MoNArch final overall architecture [5GM19-D23]

Figure 2-2 also shows NFs (emphasised with dotted line frames) that specifically implement the solutions developed in WP3 and WP4. They can be found in all layers:

- On the Service layer, the Security Dashboard enables efficient security management by human operators or via Application Programming Interfaces (APIs) as well as intuitive security status monitoring.
- On the M&O layer, Cross-slice M&O incorporates WP3 **Cross-slice Security and Resilience (S&R) Management** functions for addressing jointly increased security and resilience requirements. Furthermore, to enable flexible orchestration across elastic slices, Cross-slice M&O can integrate WP4 Slice-aware and Orchestration-driven **Cross-slice Elasticity**

Management functions (see Section 4.3.2.3). Cross-domain M&O can include WP3 **Cross-domain S&R Management** that manages security and resilience issues within a single slice, but between different management domains. In addition, Cross-domain M&O can integrate WP4 Computational and Orchestration-driven **Cross-domain Elasticity Management** functions if intra-slice flexible resource management is required (see Section 4.3.2.4). The M&O layer also hosts the Big Data Analytics module defined in WP4 (as part of the **Integrated Data Analytics Framework in the MANO**), which can support the elasticity functions located in the Cross-slice M&O and Cross-domain M&O (see the Measurement tool described in Section 4.5.1.1). Further, the WP3 **5G Fault Management (FM)** and **5G Security Management** and the WP4 **Elasticity Management** functions address network alarms, security procedures and elasticity performance, respectively, on the granularity level of NSSIs. Edge Cloud infrastructure nodes can host so-called "5G Islands", i.e., mobile networks with minimal functionality that can operate autonomously and do not require connectivity to central data centres.

- The Controller layer can host the WP3 **Scalable Control Framework for Security, Resilience, and Elasticity**, which automatically scales the controller nodes with respect to the underlying traffic in the network in order to enhance network scalability but to ensure high availability of controllers. For slicing elasticity, dedicated **ISC and Cross-slice Controller (XSC)** have been developed in WP4. These controllers can provide an 'inner-loop' control of NFs, to enforce elasticity at fast time scale. Exemplary controller applications include Reliability Control and Security Monitoring manager (SMm) from WP3 or Elastic resource control from WP4.
- The Network layer can be enhanced with WP3 **Reliability sub-plane** functions for multi-connectivity (data duplications) and network coding for improved RAN resilience. For further customisation, the Network layer may host WP3 **Security Trust Zone (STZ)** VNFs, which enable to guarantee a required level of security and trust in a dedicated area of the infrastructure, and VNFs with enhanced robustness and elasticity characteristics in case of unforeseen fluctuations of resource availability.

Finally, an **Integrated Data Analytics Framework** has been developed to allow the exchange of data and analytical services across all layers of the 5G-MoNArch architecture. In Table 2-1 we explain how the two project testbeds leverage 5G-MoNarch technology and designs and incorporate novel technical concepts, focusing on the technological pillars that are the base of 5G-MoNArch innovations. It is worth to highlight that, together, both testbeds jointly cover all the key innovation concepts and enablers of the project.

Table 2-1: Mapping of the 5G-MoNArch architectural concepts and enablers into the testbeds

<table>
<tr><th>5G-MoNArch concept / enabler</th><th>Touristic City testbed</th><th>Smart Sea Port testbed</th></tr>
<tr><td rowspan="2">Network slicing</td><td colspan="2">One of the goals of the testbeds is to prove the feasibility and functionality of network slicing in real network operations. Network slices allow achieving greater flexibility and efficiency, as well as supporting different services with different characteristics and requirements. The use cases envisaged for both testbeds are designed to consider different types of network slices, such as Enhanced Mobile Broadband (eMBB), massive Machine Type Communication (mMTC) or Ultra-Reliable Low Latency Communication (URLLC) slices.</td></tr>
<tr><td>The Touristic City testbed incorporates implementation of research prototypes, including 5G NR for the connectivity between one user and the network, supporting E2E network slicing.</td><td>Pre-commercial equipment has been modified to support network slicing as well as to include resilient and reliable NFs. Based on the work conducted in WP2, fundamental aspects such as slice blue print and slice LCM functions were designed and implemented.</td></tr>
</table>

Network Orchestration	5G-MoNArch provides orchestration and management platform and applies customised orchestration algorithms which are demonstrated in both testbeds. This allows flexible deployment of relevant Network Functions (NFs) across the network when needed and according to the network conditions.	
NFV	The Touristic city testbed has virtualised several functions in the user-plane (such as the higher layer of the RAN and the UPF) which are moved by the orchestration in order to provide the desired behaviour.	
Network resilience and reliability: RAN multi-connectivity		The testbed implements multi-connectivity[1] (see also Section 3.3.1 on the multi-connectivity terminal description) in the radio part to achieve the reliability requirements of the URLLC slice. Data duplication in combination with multi-connectivity [5GM19-D32], being a main research result of work package 3, further increases link reliability to prevent service interruption during mobility events (handover).
Network resilience and reliability: Fault Management		To improve resilience, a Fault Management (FM) approach is used. FM techniques identify, trouble shoot and isolate occurring network faults. Implemented in the testbed are monitoring tools for the detection of changes, potential problems and anomalies in network behaviour, the root-cause analysis enabling the localisation of the actual problem, and problem isolation such that the propagation of the fault effects and impact to the rest of the network can be minimised.
Network elasticity	Network elasticity (especially inter slice and orchestration driven, as defined by WP4) and efficient use of the resources is a crucial topic that is analysed in context of 5G-MoNAch and poses several technical challenges. The outcome and the result of the investigation concluded in WP4 is implemented in this Touristic City testbed (see Section 4.3.2.4). The use cases designed for the Touristic City allow taking the advantage of provided solutions for elastic orchestration and resource allocation strategy.	
AI-based data analytics	The orchestration algorithms developed for Touristic City testbed rely on AI and Machine Learning (ML) techniques. The goal is to design specific algorithms that would determine the best VNF placement	

[1] Multi-connectivity here refers to Multi-Radio Dual Connectivity as described in [3GPP19-37340]. The term multi-connectivity is used here to prevent from confusion with the dual-connectivity feature introduced with 3GPP Release 12.

	between central and edge clouds and design elastic NFs capable of resource adaptation and utilisation depending on the current network conditions (see also description of the Measurement tool in Section 4.5.1.1).	
Orchestration-driven Elasticity	This innovation focuses on the ability to re-allocate NFs within the heterogeneous cloud resources located both at the central and edge clouds, considering service requirements, the current network state, and implementing preventive measures to avoid bottlenecks. The algorithms implementing the orchestration-driven elasticity need to cope with the local shortage of computational resources by moving some of the NFs to other cloud servers which are momentarily lightly loaded. This is particularly relevant for the edge cloud, where computational resources are typically more limited and more cost efficient than in the central cloud. Similarly, NFs with tight latency requirements should be moved towards the edge by offloading other elastic NFs without such tight timescale constraints to the central cloud servers. More details on the implementation of this enabler are provided in Section 4.3.2.3	
Computational Elasticity	The goal of exploiting computational elasticity is to improve the utilisation efficiency of computational resources by adapting the NF behaviour to the available resources without impacting performance significantly. Furthermore, this dimension of elasticity addresses the notion of computational outage, which implies that NFs may not have sufficient resources to perform their tasks within a given time. To overcome computational outages, one solution is to design NFs that can gracefully adjust the amount of computational resources consumed while keeping the highest possible level of performance. RAN functions have been typically designed to be robust only against shortages on communication resources; hence, the target should be directed at making RAN functions also robust to computational shortages by adapting their operation to the available computational resources. An example is a function that chooses to execute a less resource-demanding decoding algorithm or number of iterations in case of resource outages, admitting a	

	certain performance loss. While this is not a focused implementation of the Touristic City testbed, the implementation of e.g., the RAN higher layers allowed us to assess the performance of this enabler in WP4.	

In the following chapters, we describe in detail the deployments of the Touristic City testbed and the Smart Sea Port testbed respectively. Those deployments leverage on the concepts described above, implementing these concepts in real-life platforms and evaluating their performance experimentally. In this way, the experimental effort conducted in 5G-MoNArch serves to validate all the key concepts devised in the project.

3 Smart Sea Port testbed

3.1 Introduction

In the Smart Sea Port testbed, 5G-MoNArch concepts are tested in a realistic industrial environment, under the constraints imposed by such an environment. In industrial applications, communications are very critical for a number of operations and the failure of the communications systems may result in severe problems for such operations, e.g., leading to production downtime. Availability, reliability and security are therefore mandatory requirements for the usability of 5G networks in many industrial applications.

To demonstrate that the network slicing principle is technically feasible, three slices for use cases with different requirements have been setup on a shared infrastructure in the Smart Sea Port testbed: (i) one slice is used to control a traffic light as part of an Intelligent Transport System (ITS), (ii) a second slice is used for enhancing maintenance activities of field engineers by remote support through AR applications, and (iii) a third slice is used to transmit the measurements of sensors mounted on barges for improved pollution control.

To deploy these use cases with a common network infrastructure, both roads and waterways in the port area need to be covered by a radio cell. To this end, the Smart Sea Port testbed comprises two radio cells with a maximum range of approximately 8 Km, covering a large area of the Port of Hamburg. Leveraging on legacy signal processing and RF built on Long Term Evolution (LTE), both RAN and CN have been enhanced with the novel 5G functions relevant for the addressed use cases, including among others the functions to provide reliability and to enable the network slicing (e.g., network slice selection). Terminals have been extended by the possibility to connect to multiple slices in parallel and by Multi-Connectivity with packet duplication to reduce the latency in handover situations and to increase the reliability of data transmission. A Lifecycle Management tool has been developed by which slices can be instantiated and activated in very short time. Furthermore, the sea port of Hamburg has been used as example case for a simulative study on security measures.

This chapter is structured as follows. Section 3.2 recaps the key use cases and corresponding requirements to support various applications in the Hamburg port. Section 3.3 provides an overview of the implementation of terminals and infrastructure, along with the network slicing implementation and operation. Section 3.4 explains the deployment aspects along with the logical network architecture. Section 3.5 presents some key experimental evaluations on absolute latency of SDR and air-interface modem, latency of Central vs. Edge deployment and Multi-Connectivity performance within the deployed testbed as well as considerations on security measures. Finally, Section 3.6 describes the deployment of a simulated sea port environment to demonstrate the security capabilities

3.2 Use cases

5G-MoNArch intends to improve the operation of the Hamburg port by leveraging the 5G technology to support various use cases or applications that are useful for the port's operation. As the nature of these applications is quite different, this technology needs to provide support for service diversity. As the port operation would be severely harmed in case any of these applications did not work properly, the main requirements behind the various use case are:

- *Reliability*: It is essential to provide a guaranteed network connection that maintains a good quality.
- *Resilience*: It is also essential to guarantee availability all the time, even in the case of failures.

Specifically, the Smart Sea Port testbed incorporates the following three use cases, which are deployed in the real operation of the Hamburg port:

1. *Better traffic flow* (Intelligent Transport Systems (ITS) through traffic light control): The testbed includes the monitoring and control of an important traffic light at one of the main truck entrances to the port area, which is connected to the central traffic control through the 5G mobile network with a dedicated slice.
2. *Enhanced maintenance experience:* The testbed includes AR applications for the port's engineering teams connected via mobile broadband. Engineers can access construction plans

and information for buildings and assets and get remote live assistance by experts using AR applications on tablets or goggles. This equipment is connected to a central application server through the 5G mobile network, using a dedicated slice.

3. *Improved pollution control:* The testbed includes sensors installed on mobile barges operated on behalf of the Hamburg Port Authority (HPA). These sensors provide real time environmental measurements on the current air quality within the port area. The sensors are connected through the 5G mobile network with a dedicated slice.

In the following, each of the above use cases is described in detail.

3.2.1 Use Case 1 – Better traffic flow

Today, traffic light management in the Hamburg port area is largely based on a fixed network infrastructure. Monitoring and controlling the status and cycle times of traffic lights thereby represents an important cornerstone of road traffic steering, which aims at an optimised routing of road traffic to reduce traffic jams, emissions, and improve the overall port efficiency. Road traffic steering, in particular for trucks, is thereby part of an overall ITS system within the port area. Traffic lights today either run completely standalone, i.e., based on fixed pre-programmed cycle times, or their management is still performed using analogue legacy technology based on copper lines which is up to 30 years old. This ageing infrastructure needs to be replaced in the next years, and furthermore additional traffic lights shall be connected to the central traffic management system in order to enable road traffic steering within larger parts of the port. The traffic lights' connection could be done by optical fibre. However, this poses the following issues: (i) the necessary fibre infrastructure is not available in all port areas; (ii) expansion of fibre infrastructure is a long-term project with high investments; (iii) digital connection requires replacing all legacy copper cabling, and (iv) Capital Expenditures (CAPEX) and time efforts can be very high – CAPEX for connecting a single traffic light with fibre are up to 100 k€[2], including the equipment and the construction works for laying the lines. For temporary constructions sites mobile traffic lights are frequently used, but they are not connected to the central traffic management system, which typically leads to a clear increase of traffic jams due to the missing alignment with the regular traffic lights.

The 5G network slicing solution developed within 5G-MoNarch allows to setup a URLLC network slice. While low latency is not required for traffic light control, the high reliability of the network slice makes it an excellent opportunity to easily connect traffic lights in a safe, secure and high-performing manner, but with manageable CAPEX. This allows to integrate all types of traffic lights – already existing fibre-connected, existing non-connected, and temporary traffic lights – into the sea port's central traffic management system. To demonstrate the feasibility the control of one of the traffic lights in the port has been implemented, which is shown in Figure 3-1.

Figure 3-1: Location and picture of the testbed's traffic light

[2] This amount was provided by HPA based on a quotation of the Hamburg Traffic Authority.

In particular, 5G-MoNArch deploys a network slicing solution that provides high reliability and security and serves the following aspects of traffic light management: (i) monitoring the status of the traffic lights; and (ii) optimising traffic flow by connecting each traffic light in the Hamburg port area to the central traffic management system.

To demonstrate this use case, it is shown that the deployed system allows for managing the traffic lights with a separate network slice that can be operated independently from other parts of the port's network infrastructure carrying less critical traffic. It is shown that even in extreme situations, e.g., in terms of radio channel degradations or peaks of network traffic within the sea port, the reliability of the network slice handling the traffic light related network traffic is not compromised. This is demonstrated and evaluated by injecting artificial traffic to the network, which would overload the mobile broadband slice while avoiding side-effects on the isolated slices (c.f. Section 3.5.5 for corresponding results).

3.2.2 Use Case 2 – Enhanced maintenance experience

With more than 100 engineers employed, the Port of Hamburg engineering team is the largest in northern Germany. Today, construction works, maintenance and operations at all kind of technical installations (for example, water gates, bridges, railway tracks and signals) within the port area uses offline data or even paper documents such as blueprints and construction plans, and often require on-site assistance by the engineering team. Engineers have to visit sites to discuss different construction options with technicians, stakeholders, or construction workers, or a maintenance team working on a technical installation needs further assistance through an expert in order to solve a certain issue. In case the on-site available offline documentation is insufficient, or further assistance through an expert is required, such documentation or the expert has to be brought physically to the site, which is time-consuming and expensive. Since the sites are typically located in areas without a fixed network connection, the 5G-MoNArch mobile network slicing solution enables those engineers and technicians in the field to use digital applications and solutions with a dedicated service quality and security.

In particular, solutions enabling the safe and secure remote access to document databases, including access to pictures and video material, and Augmented Reality (AR)-based remote assistance including live video streaming shall clearly increase the efficiency of the engineering team. In particular, mobile network connected AR-headsets (Microsoft HoloLens [Mic18], see Figure 3-2) shall provide considerable operational benefits and efficiency gains.

To this end, the Smart Sea Port testbed deploys a fast and flexible implementation of AR and video streaming system to support the engineering teams remotely or directly on-site. The following operations are demonstrated by this use case, which rely on a network slice specifically supporting the Quality of Service and Quality of Experience (QoS/QoE) requirements of AR and video streaming applications:

- AR headsets connected through a mobile network connection and a corresponding network slice to HPA's databases.
- Engineers and experts are enabled to visualise Computer-Aided Design (CAD) construction plans along with real-time or simulated data.

Figure 3-2: AR headset (Microsoft HoloLens [Mic18])

It has been shown that insufficient data throughput and too high latencies have a negative impact on the perceived quality of the above applications and use cases, and therefore in order to realise this use case a network slice providing sufficiently high data throughput combined with low latencies is required.

3.2.3 Use Case 3 – Improved pollution control

The HPA has prepared a climate protection concept that contains climate protection objectives, areas of action, activities to reduce emissions and suggestions on how to monitor and evaluate currently planned measures. The HPA’s climate protection objectives are based on the regulations of the Free and Hanseatic City of Hamburg, which foresee a 40 percent reduction in carbon emissions by 2020 and an 80 percent reduction by 2050, based on 1990 as reference year [Ham19]. To ensure the continuous reduction of adverse impacts on the environment, a systematic environmental management system in line with the criteria of the international environmental management standard was introduced in 2011. The relevant impacts on the environment are analysed and quantified to identify further areas of action and develop appropriate measures.

To implement the above measures, more sensors for continuously monitoring the air quality need to be deployed in the Hamburg port. Besides installing sensors at fixed locations in the port area, measuring the air quality at changing locations, and in particular on the waterways, has been defined by HPA as a necessary means to achieve the intended objectives. However, by today, there does not exist a reliable infrastructure for mobile sensors. To overcome the current limitations, environmental sensors located on HPA ships (Figure 3-3, and further details on the sensor boxes in Section 3.3.1) are being connected through the 5G mobile network testbed using a dedicated network slice, to continuously transfer real time environmental data measurements to the HPA IoT cloud. This involves the following challenges:

- Get fast, stable and secure transmission of sensor measurement data to the corresponding sea port IoT cloud through stable and secure connectivity.
- There shall be no interruption of the sensor measurement data transmission due to interruption of the mobile network connection, e.g., due to shielding behind large buildings or ships
- Moving sensors need to hand over between mobile network cells reliably.
- Visualisation of data, relationships, and identification of events and incidents.

The HPA plans to connect many more sensors on moving vehicles (not only barges) in future, which clearly accentuates the relevance of Use Case 3.

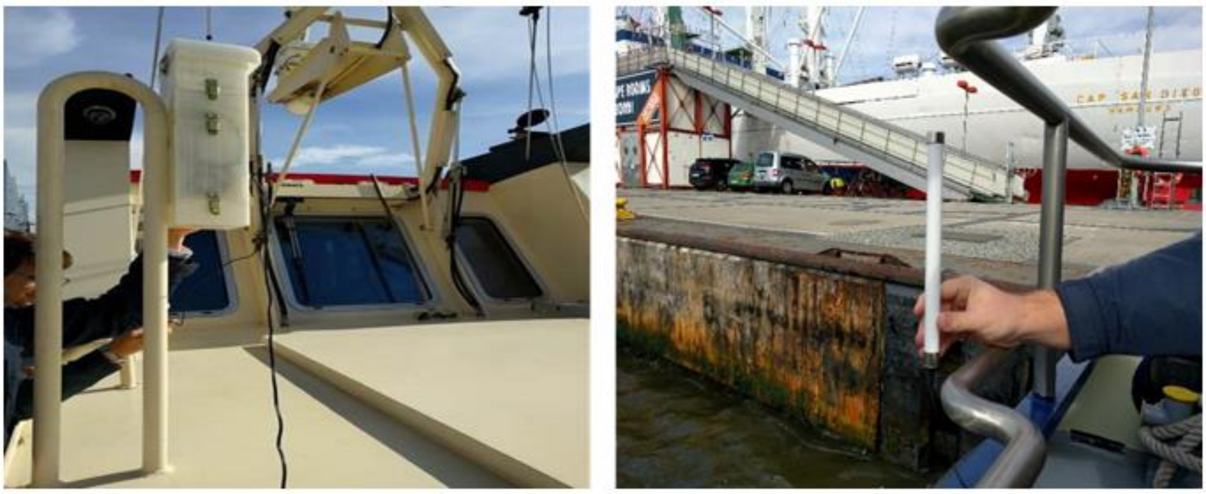


Figure 3-3: Environmental sensors located on HPA ships

3.3 Testbed implementation

This section describes the implementation of the different components to provide the functionality required to address the use cases explained above. In Section 3.3.3, we explain how the different components (including the individual terminals and the network equipment) have been deployed in the Hamburg testbed.

In particular, the different components that have been implemented in the Smart Sea Port testbed include the following:

- The terminals and sensors, which have been implemented and deployed in the testbed (Section 3.3.1)
- The network-side equipment, which includes novel 5G functions (Section 3.3.2).
- The network slicing functionality, which comprises both the terminals and the network side equipment (Section 3.3.3).

It is worth highlighting that the network-side equipment builds on commercial devices which have been updated to include the novel functions required for this testbed. Such an implementation provides a strong basis for the adoption of these novel functions in future commercial products.

3.3.1 Terminal implementation

To implement the required network slicing functionality as well as the reliability and resilience features, the testbed uses two different types of terminals:

- A multi-slice mobile terminal which builds on top if legacy commercial router boards with Long Term Evolution Advanced Pro (LTE-A Pro, according to 3GPP Release 14 specifications) compliant modems.
- A multi-connectivity mobile terminal which builds on Software Defined Radio (SDR) terminals and small-form-factor Personal Computers (PCs).

Only the above two terminal types allow to exploit the full set of features required for the three slices of the Smart Sea Port testbed.

The main reason for using different terminal types is the following. The multi-connectivity feature can only be realised through the SDR terminals. However, these SDR terminals have a certain lack of reliability due to their software-based design. This is an obstacle, in particular for the implementation of the traffic light management, where a high reliability of the terminal is required due to the integration of the traffic light into the port's central traffic management system. Therefore, in order to be able to count on the multi-connectivity and have reliable terminals at the same time, the two different types of terminals are needed.

Depending on the installation site (traffic light control switch box, or barges), different commercial antenna types have been used to connect the terminals to the base stations. In particular for the barges, the individual components have been mounted in a shielded box in order to protect the equipment from the harsh environment (e.g., very low and high temperature, high humidity, permanent and abrupt movement, salty water, etc.).

In the following, we describe in detail the different types of terminals employed in the testbed, including the two mentioned above as well as the sensor boxes.

Multi-Slice Mobile Terminals

The Multi-Slice Mobile Terminals used in the Smart Sea Port testbeds have been built on top of the Mikrotik RouterBoard [Mik18] commercial devices, which are compliant with LTE-A Pro and use an embedded OpenWRT Linux [Wrt19]. These terminals have been customised in order to provide the desired capabilities (including the multi-slice functionality).

The multi-slice capability, i.e., a terminal being connected to multiple slices at the same time, is achieved through modification of the operating system and providing a dedicated slice configuration and deployment system on each mobile terminal. Because Common-Off-The-Shelf (COTS) terminals are applied, the solution is proprietary and not directly integrated in the Non-Access Stratum (NAS) signalling of the mobile network. Figure 3-4 shows a picture of the terminal as it used on each barge. However, the implementation of network slicing on Operation System layer allowed for utilising existing COTS modems, which are proven to be sufficiently robust and scalable in the implemented use cases, i.e., where permanent reliable connectivity in a very large network area is required.

Figure 3-4: Mikrotik RouterBoard used in the testbed

Multi-Connectivity Mobile Terminals

One of the key novel functionalities evaluated in the Smart Sea Port testbed is that of multi-functionality. This functionality is essential in order to provide the desired reliability levels in the testbed. In order to provide this functionality, Multi-Connectivity Mobile Terminals have been implemented on top of COTS hardware.

The multi-connectivity approach is illustrated in Figure 3-5, schematically showing two cells implemented in the Smart Sea Port testbed (cf. Section 3.3.2). When a mobile terminal starts moving in Cell 1 (upper) towards the cell border, it will at some point in time add Cell 2 (lower) as a secondary base station (Point 1). From this moment on, the mobile terminal is able to receive duplicated data via both base stations. As the mobile terminal continues moving, the signal strength of the secondary base station will be at some point stronger than the one of the master base station, and target and source base station change the roles (Cell 2 will become new master base station and Cell 1 secondary base station), Finally, the mobile terminal will only be connected to the new master base station's Cell 2 (Point 2). This approach is described in detail in [5GM18-D31] and [5GM19-D32].

The multi-connectivity terminal uses two Intel NUC computers [Int19] for RAN protocol processing, and two Ettus B210 Universal Software-defined Radio Platform (USRP) devices [Ett19] for Radio Frequency (RF) processing. The applied SW has been developed by Nokia and focuses on enabling intra-frequency multi-connectivity[3]. To this end, all layers had to be modified, i.e., physical layer (PHY) to enable co-existence of bearers in the carrier and still perform reliable time synchronisation, Media Access Control (MAC) layer to enable co-ordinated scheduling across multiple base stations, Radio Link Control (RRL) and Radio Resource Control (RRC) to enable multiple bearers per user device (duplication), and RRC to control the setup of multiple bearers and control the establishment/release of the multi-connectivity feature.

[3] Multi-connectivity in the context of the implementation of the Smart Sea Port testbed refers to the Multi-Radio Dual Connectivity feature as described in [3GPP19-37340]. This standard is still not fully completed, in particular the definition of Multi-Radio Dual Connectivity in combination with the 5G Core has only been introduced in v.15.4.0.

Note that the developed SDR terminal is fully compliant with 3GPP LTE-A Pro PHY, but supports upper layer 5G protocol stack, in addition to proprietary features enabling multi-connectivity[4] in combination with data duplication. Although the HW and SW was originally only meant to be used in a lab-environment, the implementation performed could be used to connect the SDR terminals reliably at a distance of about 8 Km over the air by using an additional amplifier and selected antennas. Figure 3-6 shows a picture of the terminal as it used on each barge.

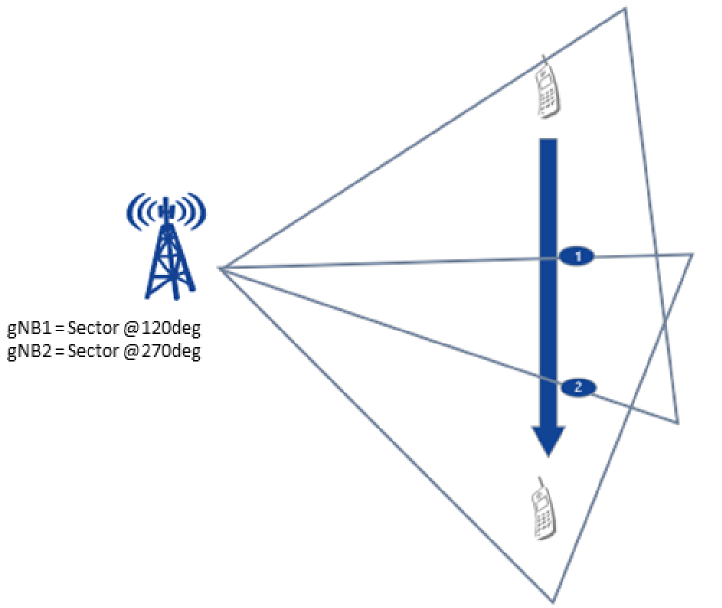


Figure 3-5: Illustration of multi-connectivity approach implemented in the Smart Sea Port testbed

Furthermore, an administration and management tool has been developed also, which permanently monitors the multi-slice and multi-connectivity terminal in order to track the performance and to intervene in the case of failures because, during the time of operation, only remote access to the individual terminals is provided.

To configure the terminal(s) with slice-level information and individual profiles for the AR applications within the Hamburg port, a life cycle management service is utilised that empowers any kind of enterprise to customise and automate the enrolment of terminals and to simplify the out-of-the-box experience with a large range of configuration options. This solution facilitates a scalable long-term solution to manage and configure several devices across the site. Two editions can be supported by the platform:

- **Setup Edition:** This edition supports the one-time configuration of enrolled devices and settings. To change a device configuration (after initial policies are set), devices must be factory reset and the new policies must be re-applied. The Setup Edition has been used within the testbed for initial terminal configuration and for re-configurations where basic policies had to be changed.
- **Dynamic Edition:** This edition allows deploying and updating enrolled device policies as many times as needed without a factory reset. The Dynamic Edition has been used within the testbed for minor updates of policies during terminal operation.

Once a profile is configured, it can be assigned to specific managed devices. Device assignment is the final step in the configuration process, though devices can be customised post deployment. Additionally, each managed device has its own device details and device log screens to assess its

[4] Due to the standardisation work in progress, and the development work on the data duplication feature in work package 3 [5GM19-D32], the implementation of the feature in the testbed (terminals, base stations and core) had to be performed in a proprietary way.

current configuration and events history that can be periodically reviewed to assess a selected device's events history and configuration.

Figure 3-6: Picture of one multi-connectivity device used in the testbed

Sensor-Box

In addition to the above terminals, the Smart Sea Port testbed also requires sensor devices that are capable of collecting the required data for the Smart Sea Port Use Case 3 (Improved Pollution Control). To this end, an Environmental Sensory Unit (ESU) has been designed specifically to meet the requirements of this use case.

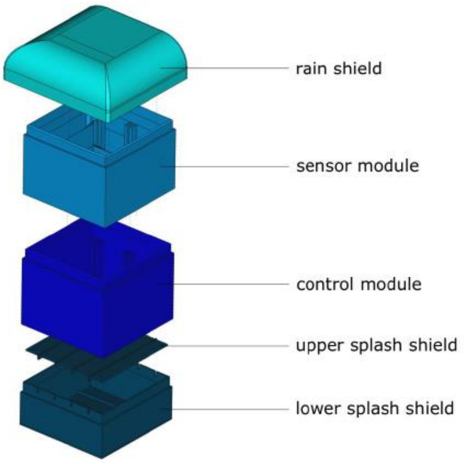


Figure 3-7: Sensor-Box design

Each of the mobile barges installed for the purpose of the testbed is equipped with the measurement ESU, that contains four sensors that provide six measurements, namely: air temperature, air pressure, carbon dioxide, nitrogen oxide, and fine dust (Particulate Matter PM 2.5 and PM 10, according to US National Air Quality standards), as well as a Global Positioning System (GPS) for the position. The ESU consists of four main parts (see Figure 3-7), listed from top to bottom:

- Rain shield: protects the electronics from rain and water from above while allowing air to be sucked in.

- Sensor module: housing all sensors which are installed on a sensor board to read the sensor data and send it to the control module. A fan guarantees a constant air flow into the box for the sensors to measure actual concentrations.
- Control module: contains the Raspberry Pi single-board computer for the overall control of all components, the Automatic Identification System (AIS) module, the GPS module and a switch. All power and data connections are plugged in here with watertight connectors.
- Splash shield: consists of two meshes, which allow air to flow out but protect all components from water entering from below. Furthermore, it provides mechanical protection for electrical components.

Each barge setup contains one multi-slice terminal, one multi-connectivity terminal, and one sensor box with four sensors.

For testing reasons during the initial phase of the testbed deployment (where the actual sensor boxes were not yet available), an additional simple off-the-shelf multi-sensor has been installed on each barge (three sensors providing GPS, temperature, humidity and gyroscopic information) and connected through the multi-slice terminal. It was finally decided to leave these multi-sensors on the barges to get additional load in the testbed.

3.3.2 Network-side implementation

In addition to the terminal equipment, the Smart Sea Port testbed also requires network-side equipment implementing novel 5G functionality. In this section we give an overview of the different network-side components implemented by 5G-MoNArch, while in Section 3.3.3 we focus specifically on the network slicing implementation and operation.

All the network-side equipment used in the testbed relies on commercial products from Nokia. While the original devices were conformant to the previous 3GPP standard (Rel. 14), the 5G features relevant for the addressed use cases have been implemented on top of this equipment. As they are already built on top of commercial devices, such implementations provide a strong basis for their possible adoption in future products.

Base stations

The testbed comprises two base stations with one antenna each (see Section 3.4.1 for further details) that together cover the area of the Hamburg port almost completely. These base stations have been built based on commercial Nokia AirScale base stations [Nok19] which have been modified to support the 5G functionality required for the testbed.

Specifically, the standard SW of the base stations has been modified in order to support E2E network slicing and the multi-connectivity and data duplication features (cf. Table 2-1) developed within work package 3. For this purpose, besides PHY, modifications to MAC, RLC and RRC have been applied. The AirScale have the following basic technical data:

- Radio standard:
  - LTE-A Pro/3GPP Release 14 compliant.
- Radio parameters:
  - Frequency Division Duplex (FDD)
  - Single Carrier, bandwidth 10 MHz
  - Downlink (DL): 713 – 723 MHz
  - Uplink (UL): 768 – 778 MHz

Core Network

The CN is based on a Nokia product, which has been modified to support multiple network slices, separate core and data planes. The Core network equipment provides the necessary support for network slicing, including the support for VMs as well as a VIM that manages the infrastructure, as well as management interfaces necessary to interact with the life cycle management tool. A detailed description of deployment and slicing procedure is provided below in Section 3.3.3.

Life Cycle Management tool

Nokia provided for the testbed a proprietary life cycle management tool with a Graphical User Interface (GUI), which has been developed for 5G-MonArch (cf. [5GM18-D51]). This tool allows for preparing slice blueprints, and to commission and decommission network slices on the fly. The tool is based on a web-interface in order to allow for an easy use.

3.3.3 Network slicing implementation and operation

As explained in Sections 3.3.1 and 3.3.2, one of the key functionalities that has been implemented on the terminals and network-side equipment is the support for network slicing. In this section, we describe with further detail the functionality implemented and their operation.

The section is structured based on the different lifecycle phases: for each phase, we describe the operation of network slicing along with the implemented functionality for such operations. In particular, the next three subsections follow the lifecycle phases of a Network Slice Instance (NSI) as defined in 3GPP TR 28.801 and shown in Figure 3-8 [3GPP18-28801]. They describe how network slices for the use cases in the Smart Sea Port testbed can be designed and instantiated and how applications can be set up using these network slices. A fourth subsection deals with SW aspects in the terminal devices as well as in the NFs on the infrastructure side.

The implementation of the Hamburg testbed does not make use of the full degrees of freedom offered by a fully automated orchestration and management system. The reason is that the environment itself is constrained by the limitations of the commercial devices employed in the testbed, which did not allow to integrate all the functionality of the 5G-MoNArch architecture. However, the testbed still implements the parts of 5G-MoNArch architecture that are relevant for this testbed, and it helped to better understand the processes, to investigate their applicability and to explore the data formats for their input and output information.

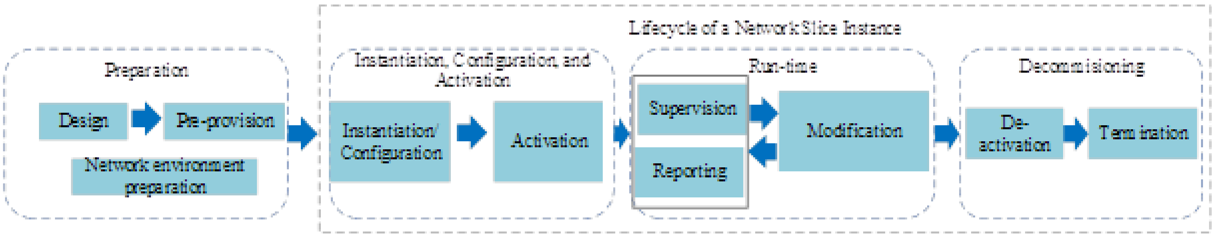


Figure 3-8: Lifecycle phases of an NSI [3GPP18-28801]

3.3.3.1 Slice preparation

The preparation phase serves to create a slice blueprint describing how the slice shall look like, which resources shall be used and how it shall be set up. During this phase, the NSI does not yet exist, and no resources are allocated. As shown in Figure 3-8, the preparation phase consists of slice design, pre-provisioning (onboarding) of the NSI blueprint and the network environment preparation.

Slice design

The objective of the slice design is to capture the requirements of the slice vertical and compile them into a slice blueprint that meets these requirements with the available resources. There are two main inputs: the demands and requirements of the vertical, and the available resources of the infrastructure platform.

To describe the vertical's demands and requirements, the GSMA has developed the Generic Slice Template (GST) concept [GSMA18-5GNS]. Following this approach, 5G-MoNArch has prepared a questionnaire comprising a set of parameters. To validate the suitability of this questionnaire, it has been filled for the use cases envisaged for the Smart Sea Port testbed. The detailed questionnaire is provided in Annex 7.1.

Some observations have been made in this process:

- Initially, the questionnaire asked for a detailed description of the required NFs. For the slices of the testbed, HPA (as vertical) did not fill in this information - for them as for probably the majority of verticals, only the quality of the provided network service matters, but not how it is provided. To improve the questionnaire, the default functionality of cellular networks was removed and only vertical-specific service features such as multicast or broadcast are asked for.

- Verticals may be not familiar with traffic profile parameters and the actual values of these parameters in an envisaged use case. These parameters allow the slice designer to assess the traffic load and its impact on the resource demand.
- Some parameters were specified too coarsely by the vertical. In practice, this means that filling the questionnaire is likely a joint effort in which a Mobile Network Operator (MNO) must assist the vertical to obtain suitable input for the slice design.
- Finally, the exercise helped to improve the questionnaire. Some parameters were found to be missing, e.g. the peering point connecting the CN to the vertical's private network. These parameters have been added in the meantime.

Based on this input, the slice design process must prepare a blueprint for a slice that meets the vertical's expectations. Slices can differ in three aspects:

- Application-specific functionality.
- Application -specific network slice topology and function placement.
- Quality of network service (i.e. data rate, latency, reliability) according to Service Level Agreements (SLAs).

The possibilities in the design are bounded by the available infrastructure as described in Section 3.4.1 and by the available functional structure in Section 3.4.2. Figure 3-9 shows the typical user traffic path from the UE to the peering point where the data are handed over to the vertical's network domain, and the configuration possibilities for this path.

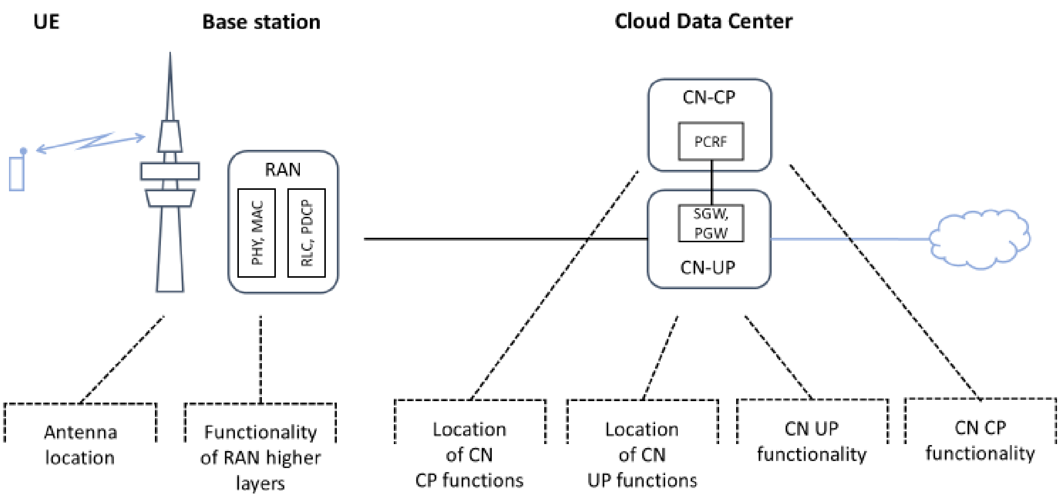


Figure 3-9: User traffic path

The relatively small size of the testbed limits the design options, and thus design decisions have been much simpler than they would be in a real slicing environment, e.g., the base stations could be assigned to the slices without using a network planning tool. In a real production environment, however, a tool would be needed to select the base stations based on their radio coverage and to assess the impact of the traffic in an additional network slice on the available radio resources. In the same way, CN and TN resources could be assigned manually, as there is no resource shortage in the testbed. In a life system, this may be different.

The network slices have the following deployment characteristics:

- **Common NFs across slices:** To utilise radio resources most efficiently, they are shared across all network slices. Therefore, PHY and MAC are common for all slices. Furthermore, also the CP functions that supervise the use of these radio resources, namely Radio Resource Management (RRM) and Mobility Management Entity (MME), are treated as common functions. The RAN functions PHY, MAC, RRM are located in the base stations; MME as a CN function is located in the edge data centre in Hamburg. Figure 3-10 depicts the functional architecture for the Smart Sea Port testbed. The common NFs are depicted in orange colour.

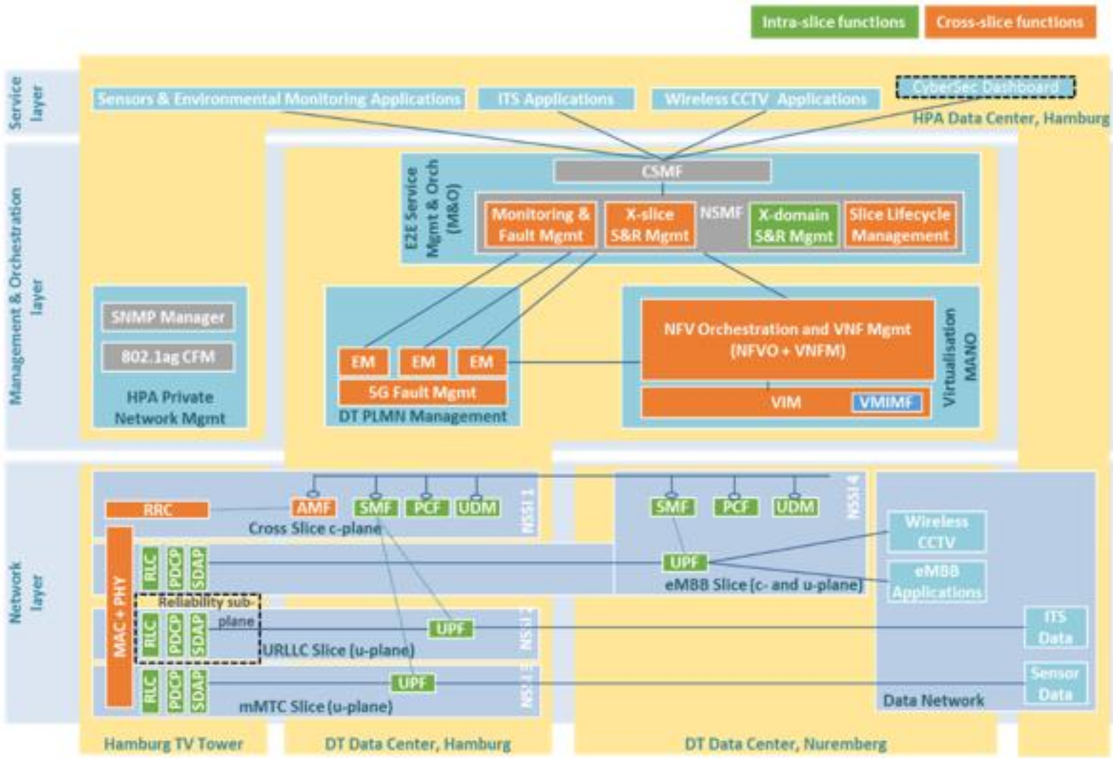


Figure 3-10: Targeted functional architecture for the Smart Sea Port testbed [5GM18-D51]

- **eMBB network slice:** This network slice is utilised to carry the AR traffic (e.g., enhanced maintenance experience for HPA service staff) as well as providing eMBB services such as Internet access or video streaming to cruise ship tourists.
  - Antenna sites: Both antenna sites are included in the slice.
  - RAN higher layers: RLC and PDCP layers are slice-specific due to customisations reflecting service requirements and make use of the common NFs MAC and PHY.
  - CN UP functionality: CN UPFs are service-specific and hence must be dedicated.
  - CN CP functionality: Policy and Charging Rules Function (PCRF) is dedicated to the slice.
  - Location of CN functions: All slice-specific CN functions run in the testbed's central cloud data centre in Nuremberg.
  - Furthermore, the AR applications (and other eMBB-like applications) in the Data Network that process the incoming user data are also hosted in Nuremberg. From there, user traffic is routed to the vertical's data centre (i.e. to HPA).
- **URLLC network slice**: This network slice is utilised for ITS applications in the sea port area, in particular for the traffic light control. The design of this slice is characterised by the edge deployment of NFs to minimise the latency, by applying a priority to the traffic to avoid side-effects by the eMBB slice, and by high-reliability using conservative configuration in RAN.
  - Antenna sites: Again, both antenna sites of the testbed are included in the slice.
  - RAN higher layers: RLC and PDCP layers are slice-specific due to customisations reflecting service requirements and make use of the common NFs MAC and PHY.
  - CN UP functionality: The CN UP uses a dedicated and customised UPF instance.

  - CN CP functionality: The PCRF is shared among the slices deployed in the local edge cloud, i.e., URLLC slice and mMTC slice. An alternative deployment option would comprise separate PCRFs for each of the two slices.
  - CN location: Due to latency requirements for traffic light control, all CN functionality for CP and UP is deployed locally.
  - Access Management: As for each slice, the user terminals admitted to the slice can be configured.
  - The ITS Data applications in the Data Network are operated in the local HPA Data Centre in Hamburg; hence user traffic is forwarded directly from the testbed's edge data centre to the vertical's premises.
- **mMTC network slice**: The slice is used to carry traffic from a group of sensors deployed in the Hamburg sea port, particularly from the barges patrolling within the port area. As previously described each barge is equipped with a set of sensors (air quality, temperature etc.) and GPS which need to deliver data at high frequency (granularity in the range of few seconds) to a central IoT cloud.
  - Compared to the previous slice, this slice uses a modified RAN protocol to enable intra-frequency multi-connectivity and data duplication as described in [5GM19-D32].
  - Antenna sites: Again, both antenna sites of the testbed are included.
  - RAN higher layers: Proprietary RAN implementation to provide 3GPP Release 16 features to the testbed.
  - CN functionality: As for the previous slice.

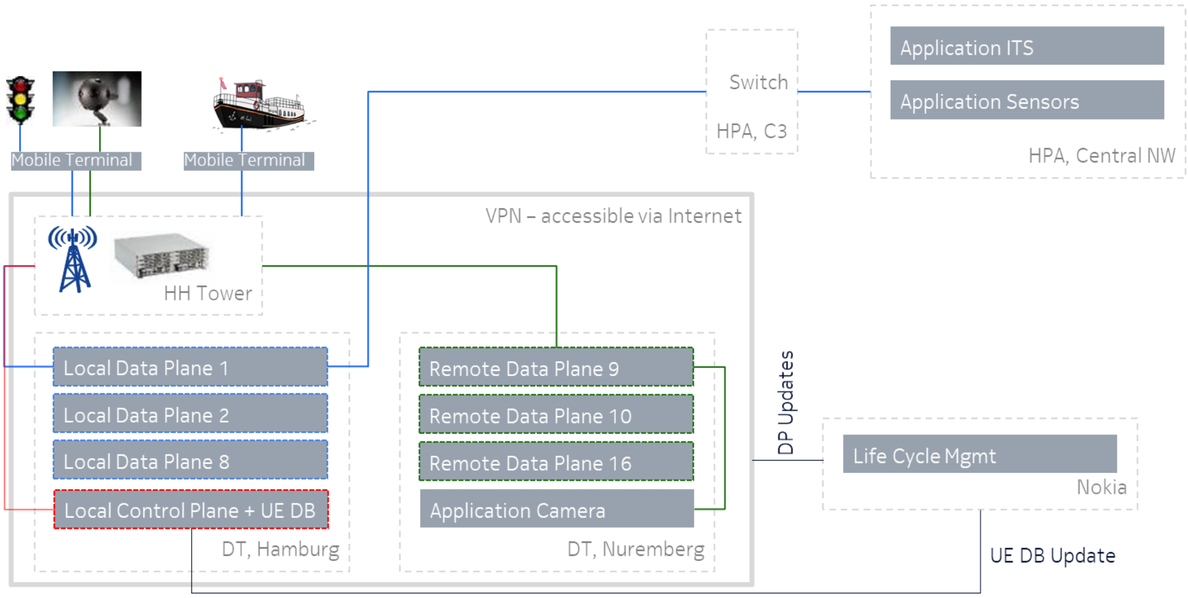


Figure 3-11: Logical deployment of the Smart Sea Port testbed

Figure 3-11 summarises the placement of logical functions of all three slices in the testbed, and Figure 3-12 shows a screenshot of the Graphical User Interface (GUI) used to design network slice blueprints.

Pre-provisioning

The pre-provisioning block checks if the necessary SW for the designed network slice is already available on the network infrastructure. If not, this SW must be on-boarded. In the testbed, this process is greatly simplified: all RAN functions are performed in the base station, and hence these functions are already on-boarded by default. Similarly, all CN functions are executed on a limited set of servers in the edge cloud data centre and in the central cloud data centre. The SW for these CN functions is on-boarded directly on these servers.

One of the objectives of the testbed is to demonstrate the feasibility of certain functions like multi-connectivity. This implies that SW must be updated when such functions are implemented and tested. In this way, on-boarding activities are triggered in the testbed mainly by changes of the SW and rarely by changes of the slice design.

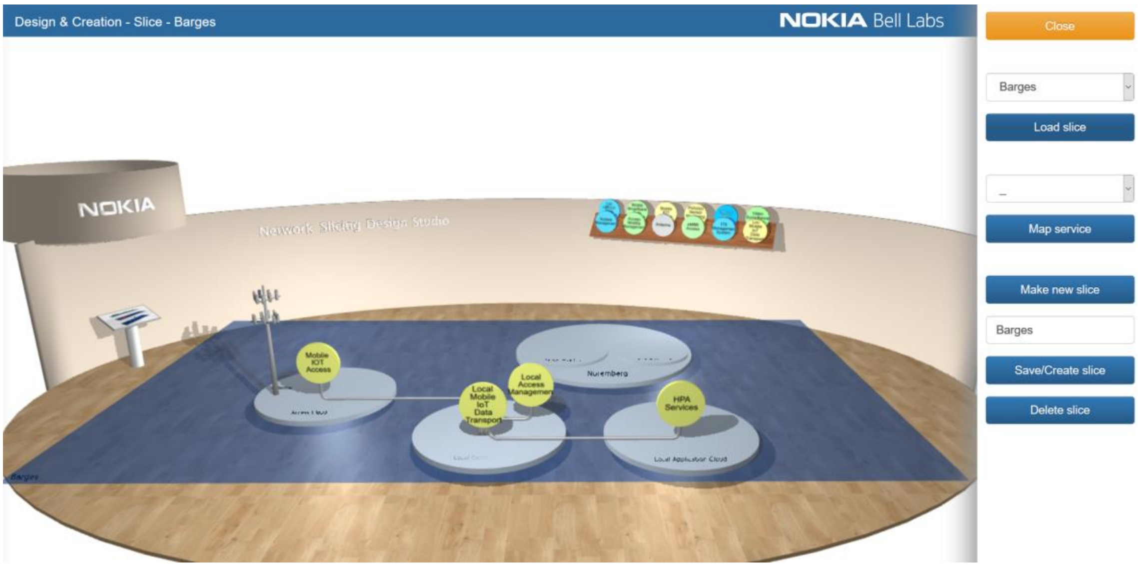


Figure 3-12: GUI to design network slice blueprints (example: slice for environmental measurements)

Network environment preparation

The available infrastructure is described in Section 3.4.1. The servers in the edge cloud data centre in Hamburg have been deployed exclusively for the testbed. On these servers, VMs are set up in a pool and made accessible to the LCM tool. The VIM implemented in the testbed is then selecting and preparing the VMs out of the available pool and upon request from the LCM. The selection of VMs (and check for availability) is performed as part of the "deployment" phase of the LCM, and afterwards the "Configuration" phase is performed. No further network environment preparations are needed in there. Similarly, multiple VMs in DT's data centre in Nuremberg are made accessible to the VIM in order to select and configure this VMs for the testbed operation, i.e., some servers in Nuremberg are assigned statically to the 5G-MoNArch testbed and used exclusively for this testbed. Again, VMs are made available in a pool of VMs to the VIM, which is then selecting VMs from this pool if needed.

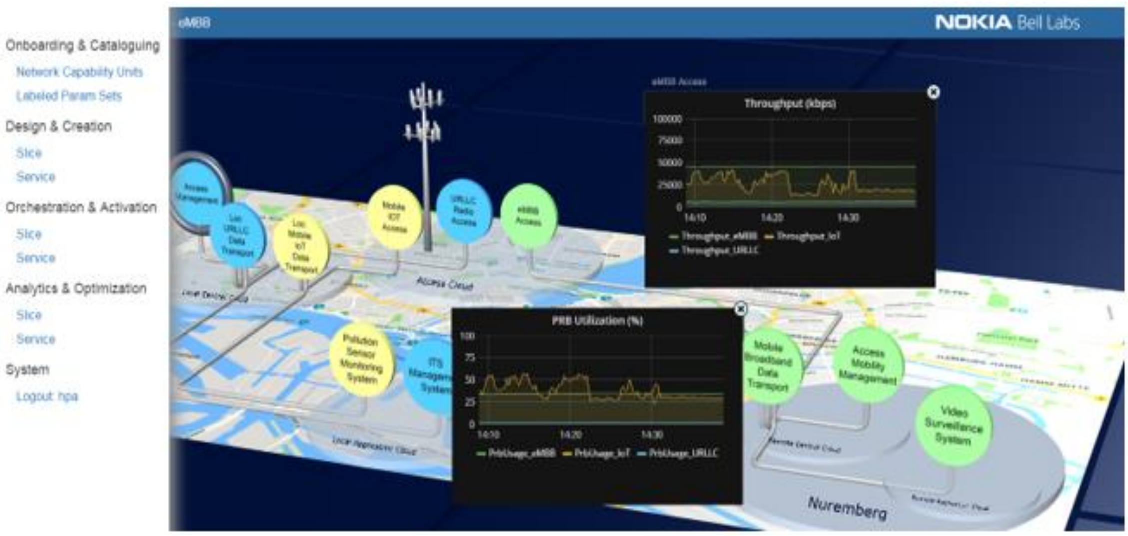


Figure 3-13: Screenshot of GUI used to control the network slice LCM

3.3.3.2 Slice deployment

At the M&O layer, a Life Cycle Management (LCM) interface (see Figure 3-13) is provided to define, commission, and monitor the individual network slices including all of the above-mentioned components. The interface allows setting up network slices, assigning them to individual network slice categories, associating devices with these network slices, and monitoring the operation of the network slices. The underlying LCM tool is specifically developed by 5G-MoNArch for the purpose of the Smart Sea Port testbed, providing the basic required slice life cycle management functionality (c.f. Figure 3-8). The LCM tool performs:

- Slice Deployment: triggering the VIM to select VMs from a pool of VMs both at local and central data centre;
- Slice Configuration: configuration of VMs and CN functions based on the slice blueprint (each VM could run each slice blueprint); and
- Slice Activation: activate the UP functions of a network slice to E2E connectivity.

Figure 3-13 shows a screenshot of the GUI, which is used to control the network slice LCM.

3.4 Testbed setup and deployment

The Smart Sea Port testbed has been setup by deploying the network infrastructure and terminals based on the implementations presented in the previous section. This section gives a brief overview on the infrastructure setup of the testbed. Further details on constraints, requirements and design of the testbed can be found in [5GM18-D51].

The Smart Sea Port testbed relies on a specific infrastructure setup that has been designed to address the use cases described in the previous section. Based on this infrastructure, a logical architecture has been mapped over the various elements to instantiate the appropriate slices. Slice design and deployment is then performed according to the requirements of the specific use cases.

In the following we first describe the physical setup of the testbed (Section 3.4.1) and the logical one (Section 3.4.2). Furthermore, we describe the deployment of the application supporting one of the use cases (Section 3.4.3).

3.4.1 Physical deployment

Figure 3-14 shows the main physical components of the testbed and their connectivity. It includes the different types of physical devices and shows how they are logically connected.

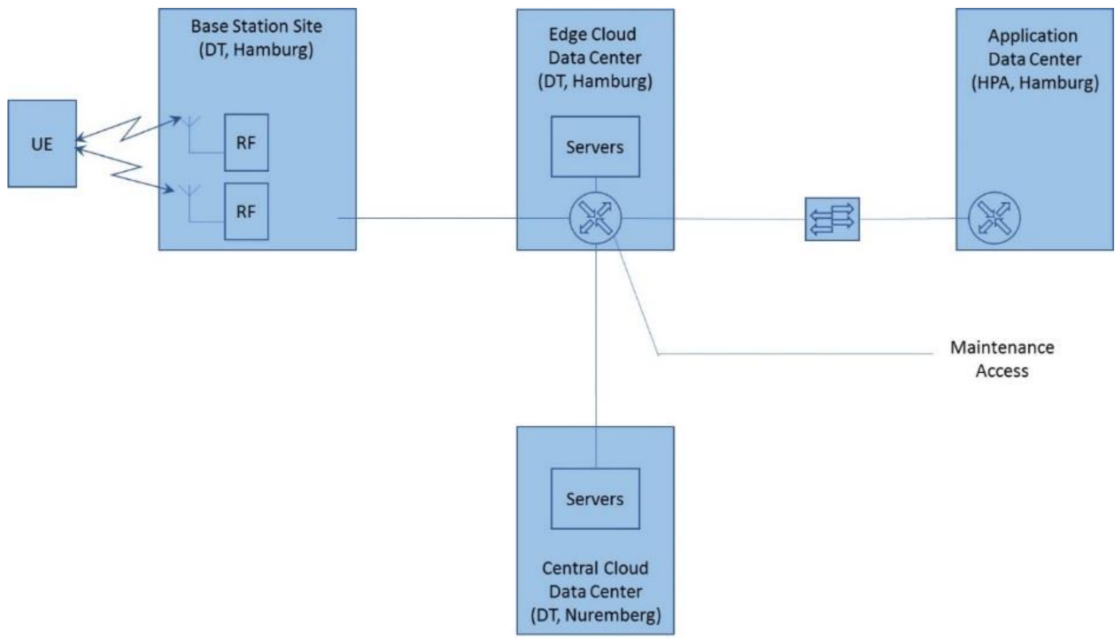


Figure 3-14: Physical components of the Smart Sea Port testbed

Starting from the different types of physical components, Figure 3-15 illustrates the physical deployment of these components in the Hamburg testbed. This includes the number and location of each of these types of devices. Specifically, the following components have been deployed:

- Two Nokia AirScale base stations at the Heinrich Hertz TV tower in Hamburg, with one antenna connected to each base station, and with one cell provided by each base station. For each base station, the radio components are mounted aside the antenna and connected via optical fibre to the base band unit. The antennas are installed at an elevation of 182 metres above ground (see Figure 7-29).
- Two Nokia Airframe servers at the Deutsche Telekom (DT) Data Centre in Hamburg. These servers run the local data and control plane of ***Edge Cloud*** network slices.
- Four servers in a DT Data Centre in Nuremberg running data planes of ***Central Cloud*** network slices. The geographical distance to Hamburg is about 460 km, which is sufficiently long for a realistic assessment of the benefits of Edge Cloud deployments in particular with respect to latency.

Figure 3-15 shows the transport network (TN) connections between the physical components as solid lines between the data centres, the base station and the local applications and control centre (hosted at HPA data centre). All TN connections consist of optical fibres with a capacity of 1 Gbit/s. The main task of the TN is to exchange user data and control data between the network sites respectively between the deployed NFs. In addition to the typical requirements, network slicing requires that the data from different slices are handled strictly isolated on all protocol layers (i.e., TN and above).

- One traffic light (representing Use Case 1, see Section 3.2) located at a large street intersection at the entrance of the port area (connecting the A7 motorway to the Köhlbrand bridge, which is one of the main inlets for truck traffic into the central area of the port) has been equipped with a multi-slice terminal[5].
- An AR test environment (representing Use Case 2) has been implemented at the HPA headquarter.
- Three barges (representing Use Case 3) have been equipped, each with a multi-slice terminal, a multi-connectivity terminal, a sensor box (with four sensors), and one additional multi-sensor (three sensors), see description of the sensor box in Section 3.3.1.
- Applications supporting the individual use cases (traffic light management, pollution control) running on the local HPA data centre in Hamburg, which has dedicated connectivity to the testbed.
- The DT testbed network provides a remote access enabling full access to the Slice Lifecycle Management tool as well as the monitoring SW using standard web browsers.
- Two additional multi-slice terminals have been deployed at the Nokia offices in Hamburg, which are used to demonstrate network slicing on trade shows and conferences (to avoid interference with the operational devices on barges and traffic light).

Furthermore, temporary installations have been used during press conferences and trade shows, e.g., a temporary traffic light has been connected to the network during a press conference in November 2018 as well as for the 5G-MoNArch event on June 5th, 2019, in particular to show the functionality of slice isolation and the corresponding impact of connecting the traffic light via a best effort slice vs. an isolated dedicated slice on the use case performance. With the installation of this temporary traffic light (which is normally used for construction sites) it could furthermore be shown that the overall lead time for integrating a new managed asset into the network only takes short time (approx. 30 minutes for modem installation, slice creation and deployment, and connection to the traffic management).

[5] Just one traffic light is controlled here because the intention is not to set up a completely new traffic management system, but to showcase that the requirements for such service can be achieved by our network architecture.

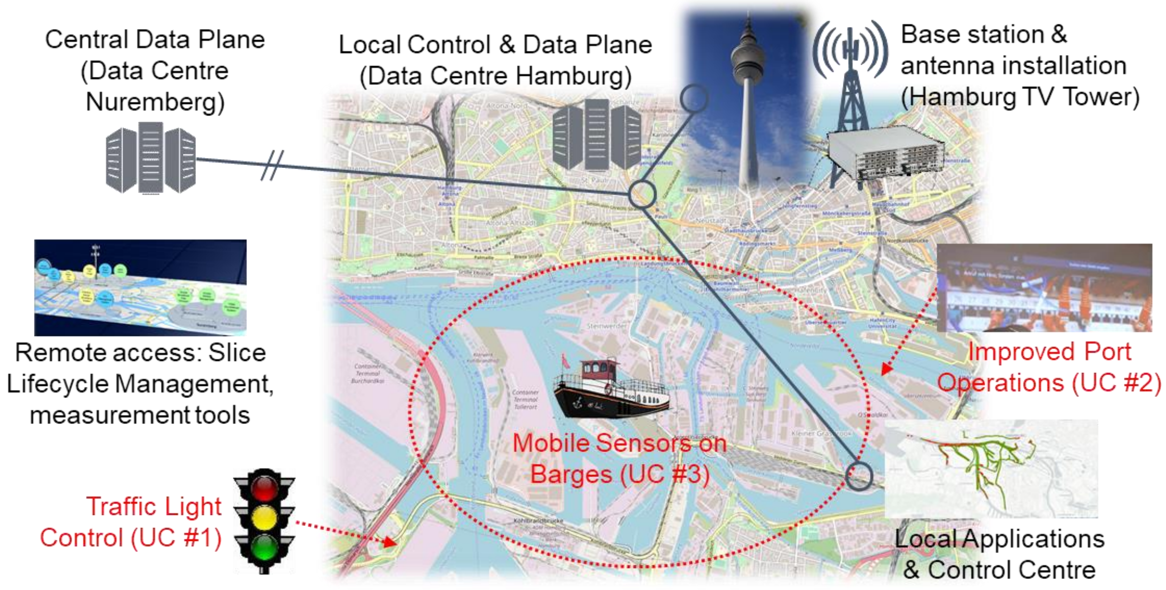


Figure 3-15 : Physical deployment of the Smart Sea Port testbed

3.4.2 Logical network architecture

In the following we explain how the physical deployment described above has been instantiated into a logical architecture that deploys the needed network slices across the various physical components in order to satisfy the needs of the different use cases.

The testbed demonstrates network slicing in an actual industrial deployment (see [5GM18-D51], Section 2.1, for more details), and evaluates the increased mobility robustness due to multi-connectivity using field-tests. In particular the testbed deploys multi-slice capable devices which connect to more than one network slice (logical network) providing different and isolated (network) services. Those services may be performed either in the edge cloud or in the central cloud, depending on the requirements of the service.

The network slicing concept applied in the Smart Sea Port testbed follows Option-2 RAN slicing with slice-specific Packet Data Convergence Protocol/Radio Link Control (PDCP/RLC) and RRC per slice as described in [5GNOR16-D41] and [5GM17-D21]. While all slices in the testbed apply the same radio access PHY and MAC implementation, higher RAN layers as well as CN functions provide specific features for each slice-type to allow for differentiating between services. In general, we have three slice-types as illustrated in Figure 3-16. The first type is enhanced Mobile Broadband (eMBB) and refers to an unmodified best-effort UP connection; the second type is a slice-type for high-priority isolated traffic (referred to as URLLC) using as LTE-A Pro compliant RAN; and the third slice-type is a modified RAN providing intra-frequency multi-connectivity with signal and data bearer duplication as discussed for enabling “0 ms handover” in 3GPP Release 16 (referred to as MTC) [3GPP19-NR_mob]. The functionality of the last slice-type introduces new features that make it non-supported by LTE-A Pro RAN and that will become part of the future 5G standard (the features implemented are not yet part of 3GPP standards). Therefore, commercially available user equipment (UE) is not applicable, but 5G-MoNArch developed an own UE using a USRP (cf. Section 3.3.1). The main application of the third slice-type is to provide high reliability even under mobility, which is a key requirement for any remote monitoring and remote-control use case. All software in the core network, i.e., on the right-hand side of the “Intra-Cloud router” is implemented on VMs, while the radio access network uses PNFs.

Furthermore, CN, CP and UP are separated and only a shared CP instance is deployed. This is a key feature to guarantee data privacy and integrity to verticals, as well as to provide the implementation flexibility needed in 5G networks. The management and control of UP and CP are performed by a dedicated LCM, which allows for defining, commissioning, and managing the installed network slices. The testbed is operated in an isolated network to avoid any kind of cross-talk with services using actual customer or provider information.

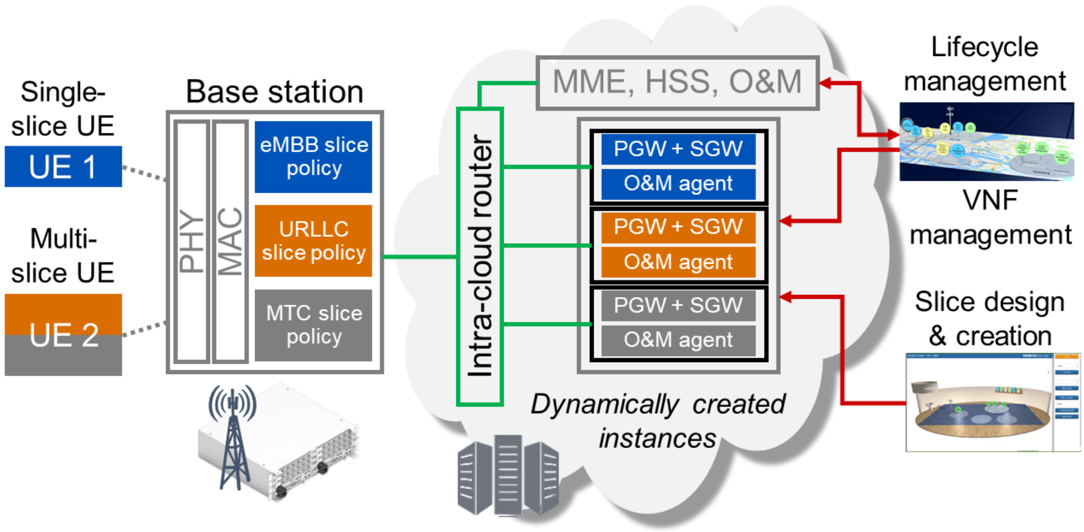


Figure 3-16: Slicing of network layer functionality in the Smart Sea Port testbed

3.4.3 AR application

Beyond the deployment of the physical network, and the instantiation of the logical architecture on top of the physical network, the 'Enhance maintenance experience' use case also requires the deployment of an AR application. In the following, we explain the deployment of this application.

The augmented reality use case is based on Microsoft HoloLens Hardware. For the two scenario two applications are used: Microsoft Dynamics 365 (Layouts) Layout and Microsoft 365 Remote Assist (Remote Assist). Remote Assist lays the basic communication Layer for both scenarios.

In the first scenario "remote assistance" the native features of Remote Assist are incorporated, to allow an expert to perform maintenance on an asset remotely. Therefore, the expert is using a desktop application to call a colleague, who is at the location of the asset to be maintained. The remote assistant is answering the call with the HoloLens and the HoloLens camera is transferring the current field of view to the expert. The expert is then able to guide the remote assistant by placing 3-Dimensions (3D) icons and drawing into the field of view. These holographic objects appear to be real for the remote assistant and are used for guidance. The asset to be maintained is displayed via camera-stream at the expert's desktop. The expert is then able to use the holographic objects to guide the remote assistant through the maintenance process.

The second scenario also uses the App Remote Assist. After a connection is established, the user is launching the App Layouts. Layouts is able to visualise any 3D-Object in the field of view of the user. Future buildings in Port of Hamburg are available as 3D-models. Therefore, HPA's engineers can visualise Building in their field of view and share the experience with colleagues using HoloLens.

3.5 Experimental evaluation

3.5.1 Introduction

By the successful setup of the Smart Sea Port testbed described above, several functional requirements on the 5G-MoNArch network architecture have already been verified, showing that the network architecture is technically feasible. Furthermore, its capability to support heterogeneous applications by specifically designed network slices has proved to be beneficial for the three exemplary use cases.

Aside of these functional aspects, performance aspects have been evaluated experimentally at the Smart Sea Port testbed. Network slicing allows to use slice-specific implementations of NFs and to place them in slice-specific locations. The benefits of such measures have been assessed and demonstrated by latency measurements:

- Application Layer latency between mobile terminals and application servers in the edge cloud located in Hamburg and the central cloud located in Nuremberg, respectively;

- Cross-correlation of latency experienced in different network slices (terminated at the same terminal) in order to quantify their independence;
- Average latency during handovers when the self-developed SDR terminals are connected to two radio access points at the same time;
- Service Creation Time of network slices (not measured automatically but manually).

In this section, we please describe the setup that we have employed to gather measurements (Section 3.5.2) and then report on the various experimental results obtained from the Hamburg Smart Sea Port testbed (Sections 3.5.3 - 3.5.7).

3.5.2 Performance monitoring setup

Before reporting on the experimental results, we describe the performance measurement platform that has been employed in order to gather these results.

To perform measurements in the Hamburg testbed, a tool for continuous performance monitoring has been implemented. This tool allows for tracking time-variant performance indicators such end-to-end latency, channel conditions, as well as sensor measurements such as position, velocity, or temperature. Figure 3-17 illustrates the setup how the performance monitoring in the testbed is implemented. However, this tool only tracks time-variant variables, i.e., it does not monitor the performance of rare events such as the instantiation of slices.

The tool works as follows. Each barge and traffic light are equipped with a modified commercial LTE-A pro terminal (supporting network slicing) and a self-developed SDR terminal (supporting multi-connectivity). Both terminals provide continuous measurement reports of different parameters, including (i) Reference Signals Received Power (RSRP), (ii) Reference Signal Received Quality (RSRQ), and (iii) connected cell, among other parameters. These measurements are collected and sent through the User Datagram Protocol (UDP) over an Internet Protocol (IP) connection on the LTE terminal to a data collector running in the local data centre.

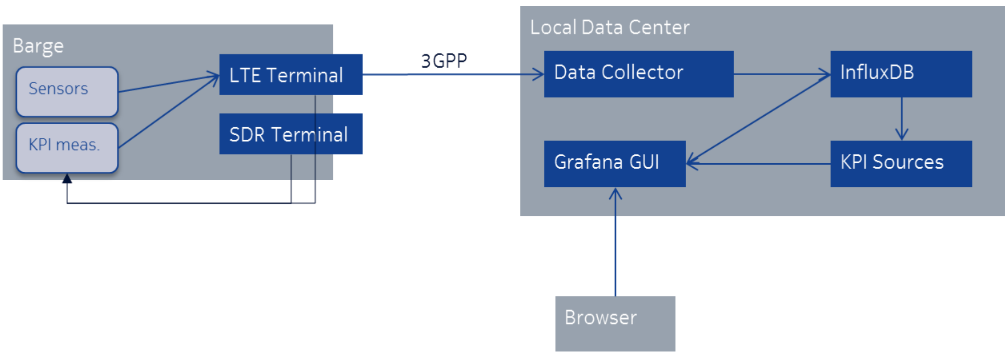


Figure 3-17: Setup of performance monitoring in Smart Sea Port testbed

In addition to the above measurements, we also collect different sensor measurements such as GPS position, speed, acceleration, temperature, humidity, and others. The data collector writes this data to a local InfluxDB [Inf19] real-time measurement database in order to have permanent monitoring of the network performance and to be able to "replay" different situations in the network. Using the InfluxDB measurements as well as own KPI sources, which evaluate and transform the InfluxDB measurements, a set of Grafana [Gra19] dashboards has been set up in order to provide an easy access to the testbed measurements. Using a browser and Virtual Private Network (VPN) connectivity to the local data centre, the testbed measurements can be shown at any location and computer, which is of significant value for presentations at exhibitions and conferences.

3.5.3 Latency KPI of different terminals

Latency is one of the most important KPIs for industrial use cases as considered in the Hamburg testbed. In particular, the Hamburg testbed considers a very large coverage area with difficult channel conditions and terminals, which are mobile. Therefore, our measurements put particular emphasis on the latency performance under different conditions such as under mobility, handover, or when operating the system at maximum throughput.

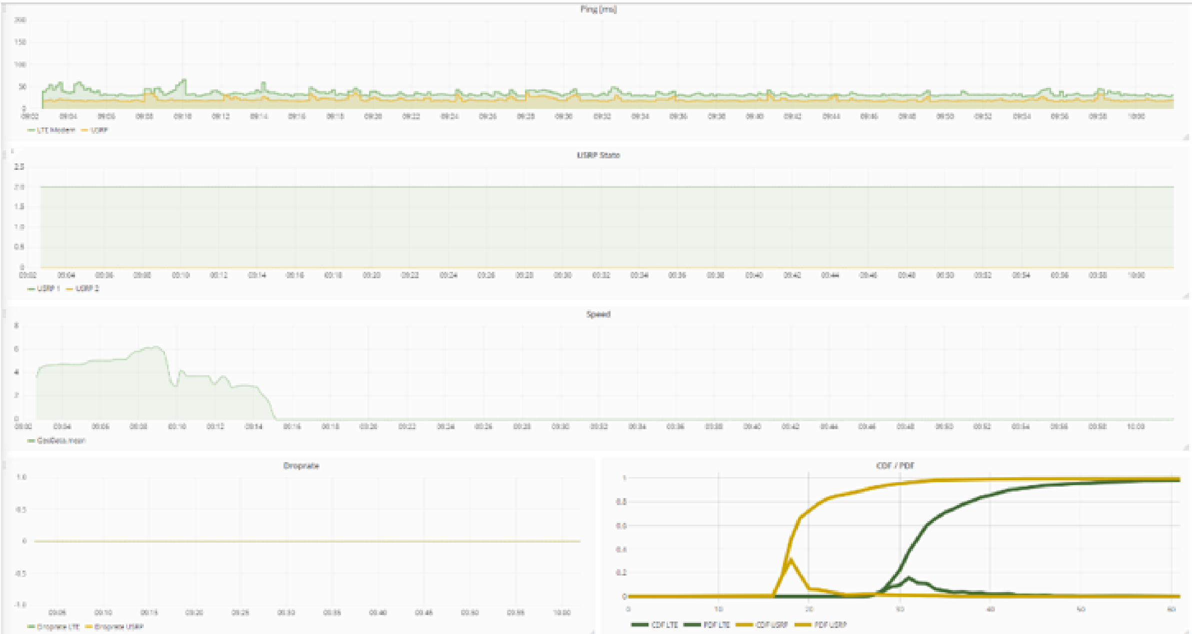


Figure 3-18: Screenshot of performance monitor implemented in Smart Sea Port testbed

In our first measurement, we compare the latency performance of the modified LTE terminal and the SDR terminal, which uses a dedicated network slice implementation. Figure 3-18 shows a screenshot of an exemplary Grafana dashboard visualising the latency performance of both terminals. The dashboard shows the instantaneous RSRP, RSRQ, Ping latency, packet drop rate, velocity, and the current Cumulative Distribution function (CDF)/PDF. This dashboard is available for each barge and it can be properly modified for presentations at exhibitions and conferences in order to focus on certain aspects. This allows for permanent monitoring of the performance of all terminals in the network.

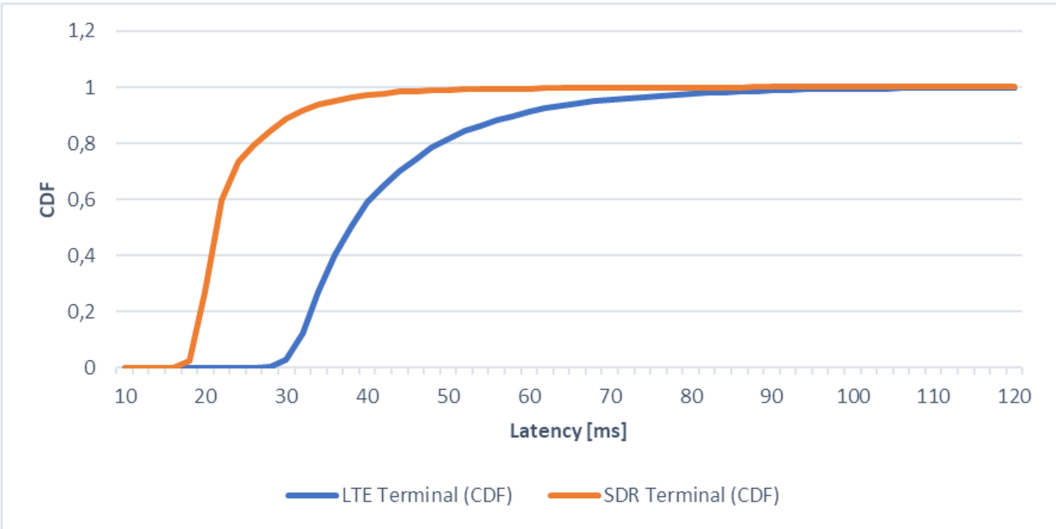


Figure 3-19: CDF of round-trip latency of LTE terminals compared to self-developed SDR device

Figure 3-19 shows the CDF for the Ping round-trip latency between user terminal and local data centre of both, the modified LTE and the SDR terminals. It can be clearly seen that the SDR terminal has a significantly lower absolute latency compared to the LTE terminal. This is due to (i) improvements at the SDR terminals themselves, and (ii) improvements at the base station for the particular slice where the SDR terminals are connected to. As a matter of fact, we applied two different base station implementations for the network slices connecting the modified LTE terminals and those connecting the SDR terminals. The performance difference mainly results from a more conservative choice of the parameters for the network slice connecting the SDR terminals (e.g., different outer loop link adaptations are used at each slice). This illustrates the strength of network slicing: it shows that dedicated use-case-specific implementations can be applied to each network slice in order to adapt to the requirements of different use cases.

3.5.4 Independent and flexible data plane deployments

An essential feature of network slicing is the possibility to place user plane instances at different and independent locations (this is referred to as orchestration). This allows to separate and/or localise traffic, and also to make sure that failures in one user plane instance do not affect other user planes, thereby increasing the resilience of the overall system, cf. [3GM19-D32]. One such deployment has been implemented in Hamburg where a local (edge) user plane instance serves traffic relevant to the port operations, and a remote (central) user plane instance serves general MBB traffic. This type of deployment allows for decoupling the operation of different use cases, and to avoid side-effects between different user plane instances, e.g., if there was a performance impairment of the central cloud, this would have no impact on the local services provided by the local user plane.

Furthermore, the placement of the user plane instances has an impact on the slice performance, e.g., on the latency. To evaluate this impact, we have performed latency measurements. In Figure 3-20 we show measured Ping round-trip latency of the same terminal towards a network slice with centrally deployed UPF which is located at the data centre in Nuremberg, i.e., about 500 Km distance from Hamburg, and a network slice with a locally deployed UPF at the data centre in Hamburg.

As expected, the latency for the locally deployed network slice is significantly lower than for the centrally deployed network slice. This shows the potential of orchestrating user planes at different locations depending on the latency requirements. Indeed, this performance gain is not only of conceptual nature, but that network slicing can indeed take advantage of different deployment setups. We also note that the absolute latency of the individual devices depends on the devices' location, channel conditions, and system load. Since the terminals' distances varies between 3-10 Km, the latency performance may vary significantly. Furthermore, the minimum latency in Figure 3-20 is about 22ms, which includes the time for uplink grant reservation, buffer status reporting, scheduling, and network latencies within the data centre (about 1ms one-way between base station and local data centre).

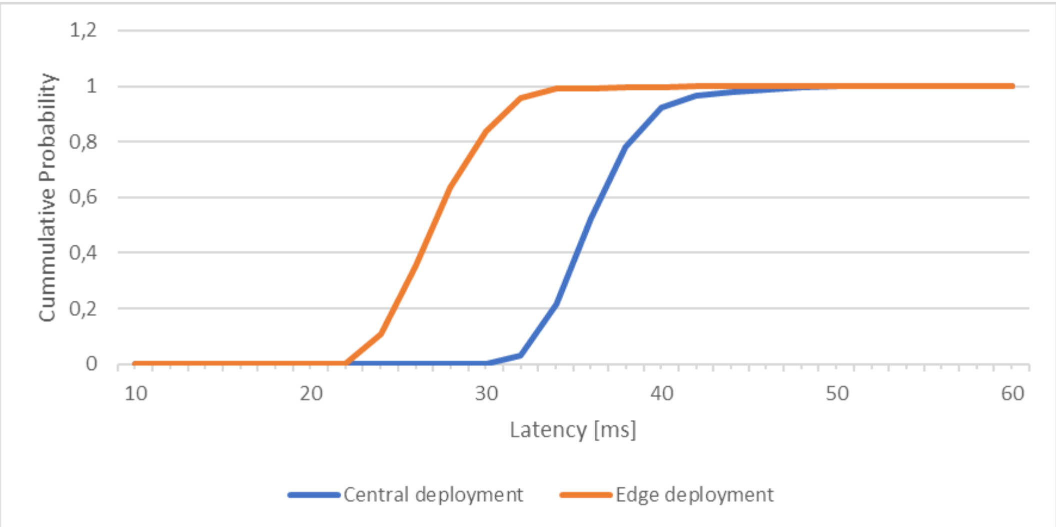


Figure 3-20: Round-trip latency towards centrally and locally deployed UPF

3.5.5 Isolation KPI: cross-correlation of network slice performance

One of the requirements on network slices is the ability to separate traffic, both in terms of security/privacy but also in terms of performance, in order to avoid side-effects from one slice to the other. For instance, a traffic light may be connected through a mobile network with two different services: one service to control and monitor the state of the traffic light, and another service to manage the traffic light, e.g., to apply SW updates. If a traffic light is connected through a mobile network, it must not happen that its operation is impaired by either the management traffic or traffic in the mobile network in general, such as during large events when many users are uploading pictures and videos. This isolation ability is essential to guarantee mission-critical services over mobile networks.

In our testbed, we apply a setup where one single terminal is connected through a best-effort MBB slice as well as a critical-IoT slice. During the experiment, we generate traffic on the terminal using iperf3 [Gue19] with a pre-defined throughput goal in order to generate a situation of high load on the best-effort slice while having a low load on the critical-IoT slice. We measure the round-trip latency on both network slices in order to determine whether the traffic on the best-effort has any impact on the critical-IoT slice. The result of this experiment is shown in Figure 3-21, where the solid red and blue line show the moving average round-trip latency of best-effort (red) and critical-IoT (blue) slice. The dashed red and blue lines show the moving standard deviation of both slices. At time-instance of 23 min, we generate the mentioned iperf3 traffic on the best-effort slice and we can observe that the average latency and standard-deviation are significantly increasing (note the logarithmic y-axis) while the performance of the critical-IoT slice is almost unaffected. This is confirmed by the correlation coefficient of the moving average latency of both slices, which is shown as black line in Figure 3-21. Until the time-instance when the traffic has been generated on the best-effort slice, the correlation coefficient is almost 1 due to the fact that we measure the latency towards the same terminal and therefore the latency without load in the network is comparable. As soon as the traffic has been generated, the correlation coefficient drops significantly because there is almost no correlation between both slices anymore. In between, there are peaks with slightly higher correlation due to the fact that iperf3 is not generating a constant bit-rate stream but rather averages the throughput over larger blocks and therefore, for shorter periods of time, no further traffic is generated. As soon as the generated traffic has been stopped at about 38 min, the correlation coefficient is again 1 as before. These results show that we are able to achieve the requirements on network isolation as stated in [5GM18-D62].

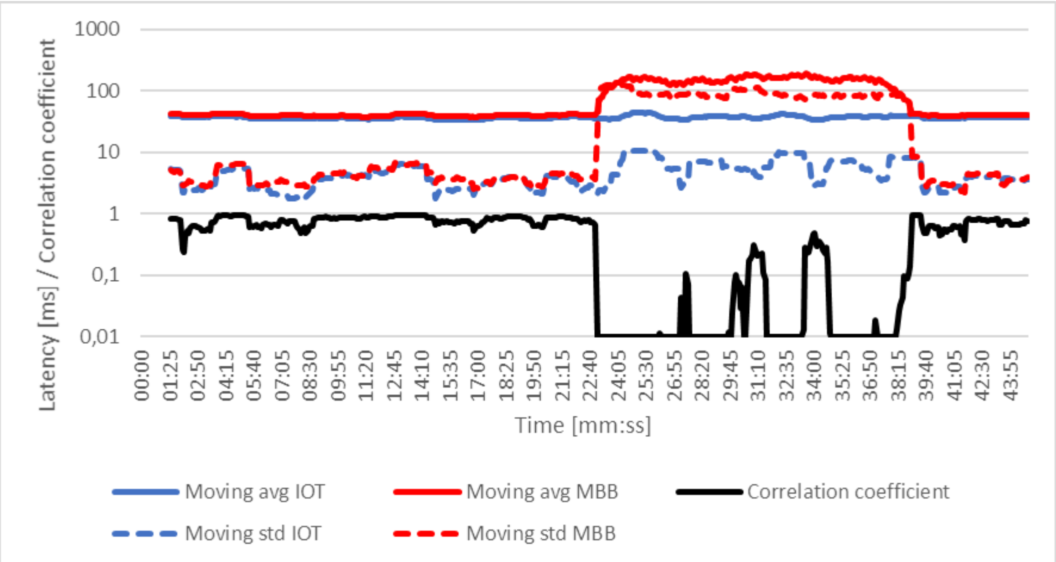


Figure 3-21: Cross-correlation of round-trip latency of two network slices

3.5.6 Reliability: latency KPI during handover for Multi-Connectivity Terminal

RAN Reliability has been intensely investigated in 5G-MoNArch Work Package 3 [5GM19-D32, Chapter 2]. One of the approaches analysed is Data Duplication, where packets are transmitted over more than one radio interface in order to provide additional redundancy. Data Duplication may be done over multiple radio access technologies, frequency carriers, or, in the most challenging case, over the same radio access technology and the same frequency carrier, which is called intra-frequency multi-connectivity with data duplication. In the Smart Sea Port testbed, this approach has been implemented using the SDR terminals presented above and using the method already described in [5GM18-D51]. In general, this approach allows for increasing the reliability of a data connection by providing a fully redundant path. In addition, it is applied during handover in order to ensure an uninterrupted handover (“0ms handover”), i.e., there is always at least one radio link active while moving from one cell to another. This is of particular importance in order to make sure that reliable low-latency services are also provided during mobility between cells, as handovers are the common case in outdoor deployments.

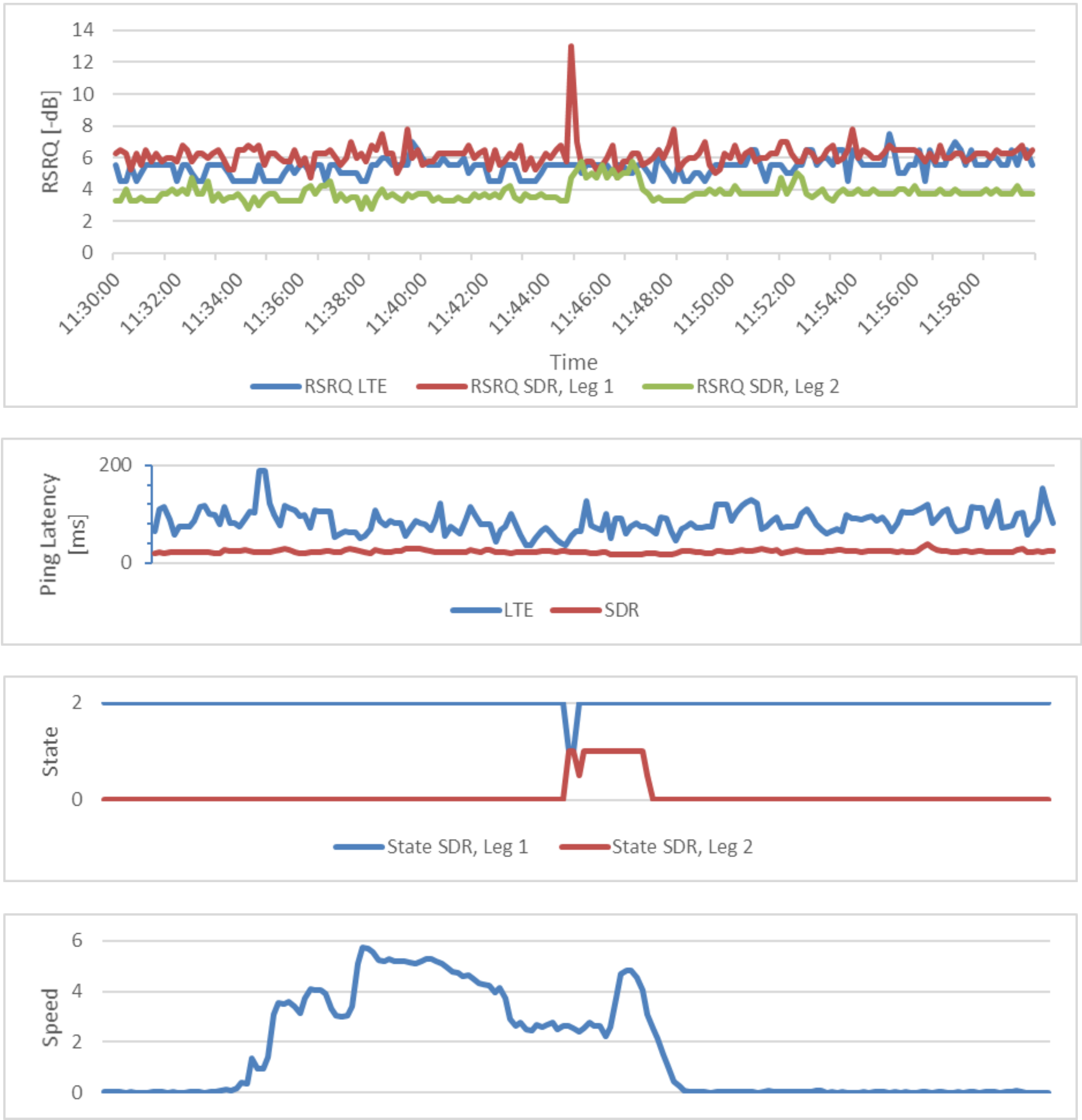


Figure 3-22: Grafana dashboard output for the tracking the performance of SDR terminals

The performance of the SDR terminals is tracked separately and various performance KPIs are recorded including the RSRP, RSRQ, Latency, and whether the multi-connectivity feature is utilised or not. Figure 3-22 shows the output extracted from the monitoring tool which continuously displays these KPIs.

- The first graph shows the RSRQ of the modified LTE terminal and both legs of the SDR terminal. We can see from this plot that the modified (commercial) LTE terminal outperforms the SDR terminals in terms of RSRQ, which was expected due to the fact that the SDR terminals are based on lab equipment.
- The second graph shows the round-trip latency performance of modified LTE terminal and SDR terminal confirming the better performance of the SDR terminal as already shown in Figure 3-19.
- The third graph shows the status of both SDR terminal legs where the value 2 corresponds to being the Master, 1 corresponds to being the Slave, and 0 corresponds to being not connected. We can see that for most of the time shown, the first leg is connected as Master, while during a period of about 3 min, also the second terminal leg is connected as Slave. The transition to multi-connectivity and back to single-connectivity happens without any interruption.
- The fourth graph shows the speed of the barge confirming that the changes of the terminal's state are caused by the mobility of the barge.

Figure 3-23 shows the CDF of the average round-trip latency of LTE and SDR terminal during handover. These results have been obtained by using those periods of times when the SDR terminal has been in multi-connectivity and determining the average latency during each of the periods. This KPI allows us to draw conclusions on the increased robustness and reliability using multi-connectivity during handover, i.e., if the number of packet drops is lower, also the number of HARQ and ARQ retransmissions is lower and therefore, the latency is reduced, and the CDF becomes steeper (less variance).

The results of Figure 3-23 confirm that the improved expected latency of the SDR terminals can be maintained during handover. In addition, the CDF of the SDR terminal is steeper than the CDF of the modified LTE terminal, which implies that the variance is lower and the modified LTE terminal experiences more often latency peaks. However, we can also see a rather long tail distribution which is due to the fact that in some cases the connection of the SDR terminals drops and therefore very high latency values are experienced. This illustrates that we were applying an experimental prototype proving the feasibility and benefits of multi-connectivity, but also not providing the same implementation robustness as a commercial terminal.

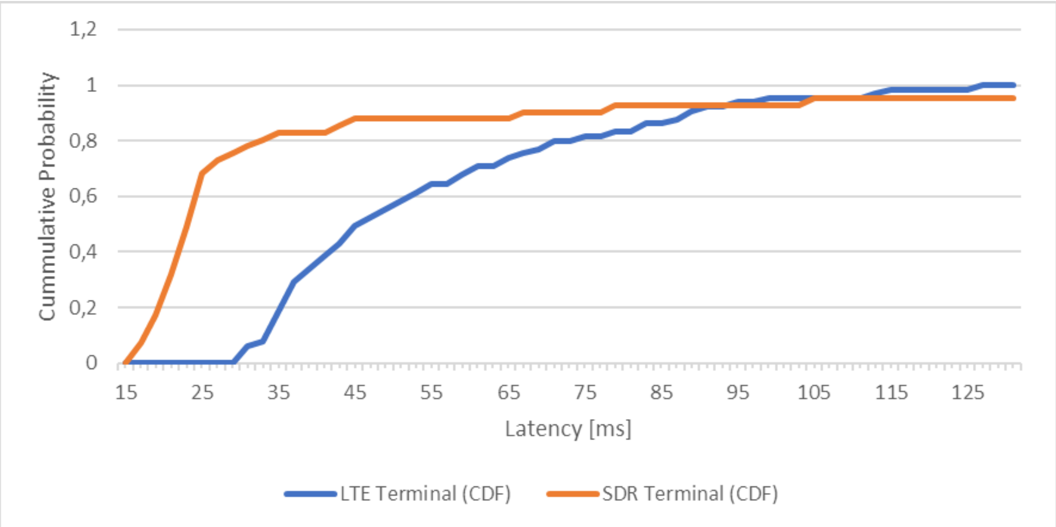


Figure 3-23: CDF of average round-trip latency of LTE and SDR terminal during handover

Beyond the experimental results obtained here, further evaluations on gains w.r.t. RAN reliability achievable with data duplication in combination with multi-connectivity have been performed in WP6 based on network level simulations for the Hamburg study area (see Chapter 3 of [5GM19-D63]). Even though the underlying application is different, a similar positive impact as measured in the testbed could be observed by the simulations for those UR slice UEs that are placed in an area covered by 2 or more cells. The size of these overlap areas determines the range where gains for UR(LLC) slices are achievable, but on the other hand it may result in increased interference zones for other slices like eMBB where the presented enablers are not applied. The resulting trade-off has to be considered by the MNO in the radio network planning and optimisation process.

3.5.7 Service creation time KPI

In the Hamburg testbed, measurements have been performed to quantify the service creation time. As mentioned above, the life cycle of a network slice is divided into multiple phases. The first phase is the slice preparation, which mainly represents the design of the network slice and pre-provisioning of network slices. In the case of our testbed, the network slice design is done manually in a GUI in order to allow for interactivity during the design phase. Therefore, it is difficult to provide a reliable measurement on the time needed for this phase.

The second phase is the slice deployment, which is divided in the case of our testbed into slice deployment, configuration, and activation. After these three steps, the network slice is operational and may be used. In the case of our testbed, these three steps take less than 10s. In order to achieve this very short deployment time, a set of pre-provisioned VMs is maintained in a pool from which the VIM can take one VM to deploy the slice-components (note that, whenever a VM is taken from the pool, a new one is pre-provisioned in order to always have a number of running VMs available). Hence, the slice deployment time does not include the boot time of a VM, which is very specific to the hypervisor, the operating system, the size of the VM, and other factors.

After taking the network slice operational, the individual user terminals need to be informed and reconfigured. In the case of our testbed, this is implemented using a polling-method, i.e., user terminals poll the network whether updated information is available. This polling interval is set in the testbed to 30 seconds and may also be triggered manually, or it could be implemented as a push-method where the network pushed the new configuration to each user terminal. The reconfiguration of the user terminal currently takes another 30 seconds mainly due to waiting timers that have been implemented and may be optimised to further reduce the user terminal configuration time[6].

Hence, the overall time from triggering the slice deployment to the time when user terminals are utilising the service offered by the network slice is about 1 minute.

3.6 Simulated Smart Sea Port to demonstrate security protection capabilities

Although the security system has not been actually deployed on site in the Port of Hamburg, namely due to security restrictions at the HPA premises, work based on simulations has been done to evaluate the way in which this deployment should be carried out. In the following, we describe the basic concepts and how this deployment should be addressed. Also, the main results of the simulations are provided.

3.6.1 Security Trust Zone concepts

According to the work developed in WP3 and described in [5GM18-D31], the security protection capabilities are managed at Network Slice level, grouping their assets according to similar security requirements into so-called Security Trust Zones. Figure 3-24 represents the STZ approach and the possible components to manage different security aspects. Each Network Slice can contain one or more STZs which might incorporate either detection, prevention and reaction capabilities, which are

[6] Although this is not strictly part of the service creation time (i.e., the time needed to create the service in the network side) we mention it here for completeness because, in the end, it matters how long it takes until a service is operational, and this includes also the UE – from a practical perspective it is not sufficient to know that we "could" connect a device, but it only matters that we "have connected" a device.

provided by SthD (Security Threat Detector), SthP (Security Threat Prevention) and SthR (Security Threat Reaction) respectively. Additionally, each Network Slice incorporates the SMm (Security Monitoring Manager), a component in charge of triggering security alerts when any security incident happens in any STZ, and optionally, a ThIntEx (Threat Intelligent Exchange) component, capable of exporting information about incidents happening in that Network Slice and importing information about incidents happening in others[7].

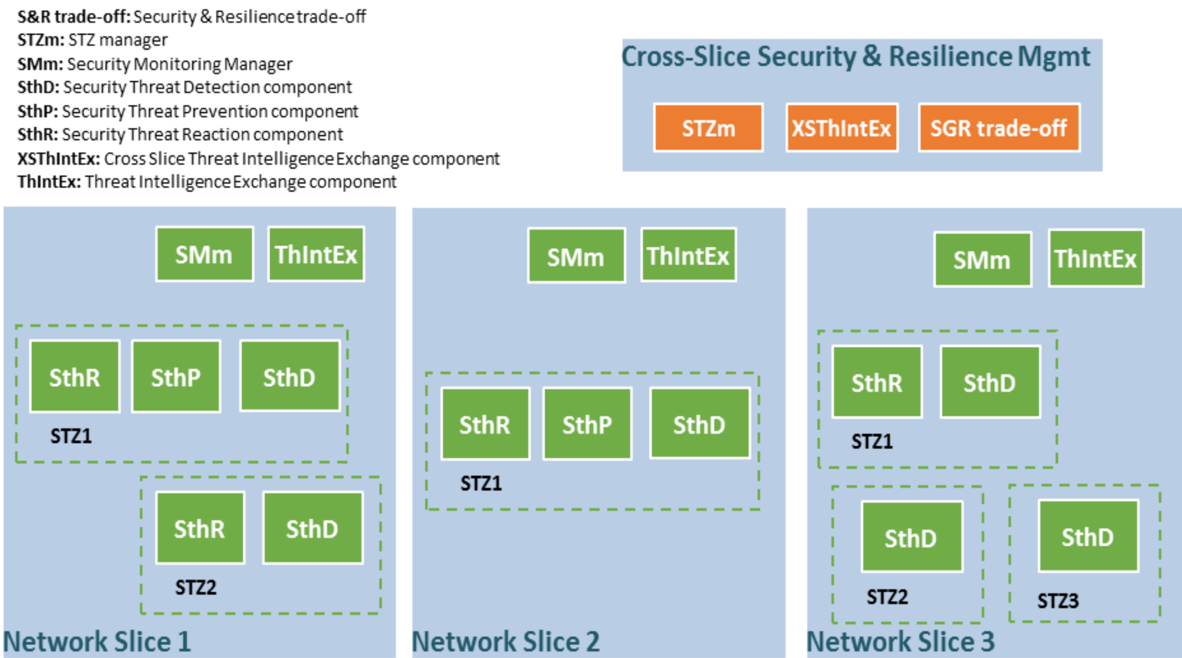


Figure 3-24: STZ approach for managing security within 5G Network Slices

Zooming in over an STZ results in the blocks is represented in Figure 3-25. The bottom of the figure represents the assets to protect within the STZ. Depending on the type of STZ the assets would be different and would have different security requirements. On top of them different security probes are gathering information about the assets deployed in the STZ. This includes Network Intrusion Detection Systems which analyse the network traffic, and in general any type of event logged by any tool capable of monitoring assets and networks, such as authentication loggers, detectors of anomalous behaviour of devices or jammer detectors.

The events reported by these security probes are usually in different formats and containing many different types of information. The SthD collects this heterogenous set of events, extracting the relevant information contained and reporting them in a unified message format to the SMm, which analyse them, looking for patterns and generating alerts in case of anomalies are detected. The verdicts about incidents generated by the SMm are reported to the ThIntEx, which distributes it across the different network slices through the Cross-Slice Threat Intelligence Exchange component (XSThIntEx), which is part of the Cross-Slice Security and Resilience Management layer. In that layer there is also a Security Trust Zone manager (STZm), which manages the creation, configuration of STZs and the deployment of the corresponding security components.

[7] Since the ThIntEx component is optional, without it there won't be propagation prevention capabilities, but that doesn't prevent the rest of the components of the STZs to work and provide security assurance to the STZ (this has been addressed in more detail in the context of WP3).

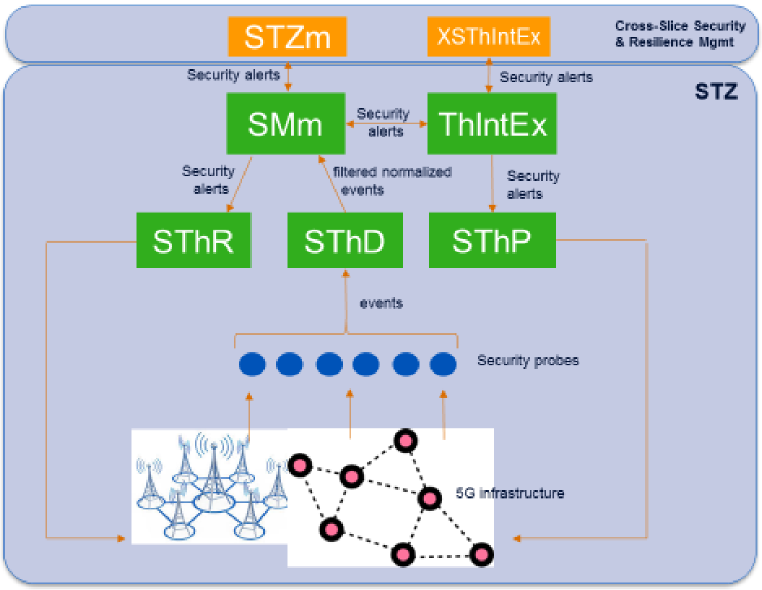


Figure 3-25: Security components within a STZ

3.6.2 Network slices and STZs for the Smart Sea Port

Section 4.2.2 in Deliverable D3.2 [5GM19-D32] describes the process for defining STZs within a Network Slice. Five main steps (see Figure 3-26) are required: analysis of the infrastructure, identify security threats, find ways to detect threats, deploy required SthD/P/R and adaptation of the SMm.

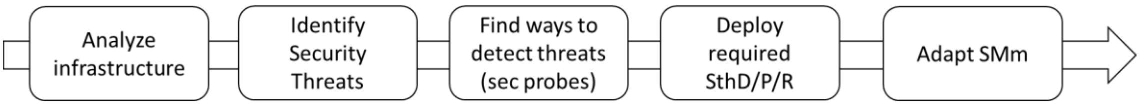


Figure 3-26: Process for defining STZs within a Network Slice

In the following paragraphs each step of the STZ definition process is described when considering the Smart Sea Port testbed infrastructure. It is worth noticing that the analysis here presented is done considering a simulated security infrastructure. This means, threats and probes identified are just an approximation to what a real deployment would need.

Analysis of the Smart Sea Port infrastructure

In this step the Smart Sea Port infrastructure is analysed in order to identify elements with similar requirements, mainly in terms of security but also using other criteria, such as, for instance, physical or logical proximity. Since three network slices have been used (one for each use case) several assets are deployed in each Network Slice, in charge of carrying out the specific operations required in each use case. For example, the following assets (Table 3-1) are deployed in each network slice.

Table 3-1: Assets and network slices for the Smart Sea Port

Network Slice	Security Trust Zone	Assets
Better traffic flow (UC1)	Port entrance	Network
		Central Traffic Control Data centre
		Traffic lights
		VPN server
Enhanced Maintenance Experience (UC2)	Virtual Operation Control Centre	Streaming server
		Network

		AR-Headsets
		Database
Improved Pollution Control (UC3)	Barge 1	Pollution sensors
		Wireless spectrum
	Barge 2	Pollution sensors
		Wireless spectrum
	Barge 3	Pollution sensors
		Wireless spectrum

For the better traffic flow use case (UC1) there can be several STZs. For instance, there could be different STZs for the traffic lights of different areas of the port. In the case of the simulated testbed built to demonstrate the security capabilities we have considered a single STZ which would be located at the entrance of the port, assuming the following assets:

- The network that interconnects the different components which are part of the network slice.
- Central traffic control data centre, which includes the servers hosting the logic for the traffic lights operations and the management of the used data.
- The traffic lights devices, which are orchestrated through the 5G network and managed by the central traffic control data centre.
- A VPN server, which manages the connections within the assets of the network slice to the 5G network.

For the enhanced maintenance use case (UC2) we have considered a single STZ focused on the devices operating in a virtual operation control centre, which would include the following assets:

- A streaming server, which would feed the AR headsets with the required content.
- The network interconnecting the different components that are part of the network slice.
- The AR headsets, which provide the visualisation capabilities to end users.
- The HPA database, which contains relevant data for the operation of the use case.

For the improved pollution control use case (UC3), we have considered that each barge with pollution sensors deployed would be considered an individual network slice. Therefore, there would be three STZs within this network slice, each containing the following assets[8]:

- Pollution sensors, which monitor different aspects of air quality, temperature, humidity, NO_2 levels, etc.
- Wireless spectrum. We have considered the wireless spectrum as an asset in this use case due to its special characteristics: barges moving across the port.

Identification of security threats within the simulated Smart Sea Port testbed infrastructure

Next step in the process for designing STZs consist in the identification of the main security threats that might affect the assets elicited in the previous steps. Same as before, in this deliverable this threat elicitation does not pretend to be a comprehensive analysis, nor a complete threat model, though it is a good approximation to what it is required for the simulation purposes of this section. Having pointed it out and considering the assets in Table 3-1 we can find the threats represented in Table 7-2 (Annex 7.5).

Mechanism to detect the identified threats

Next step in the process of defining STZs consists in the identification of mechanisms to detect the threats elicited in Table 7-2. For this, Table 7-3 in Annex 7.5 includes the possible security probes

[8] Network and central assets (e.g., the database) are not considered here because considering all the possible assets for all the network slices would have become unmanageable (too many resources would have been required).

capable of detecting the identified threats. It also includes its function, its type (hardware or software) and potential available products (free and commercial) providing such functionality.

3.6.3 Deploy security components

The next step consists in the deployment of the assets and probes in the defined network slice and STZ. For the testing purposes it has been deployed just the detection capability, deploying one Security Threat Detector for each STZ[9]. According to the deployment architecture shown in Figure 3-24 each Network Slice should have a SMm. However, in the testbed deployed we have preferred a shared SMm, which allow to optimise the resources required. However, multitenant capabilities are guaranteed by the SMm deployed, which allows to isolate the activities carried out by each Network Slice. Figure 3-27 represents the components deployed for each asset and the corresponding security probes. The diagram also shows the subnets used for the simulated testbed and the IPs assigned to each asset and for the SthD of every STZ.

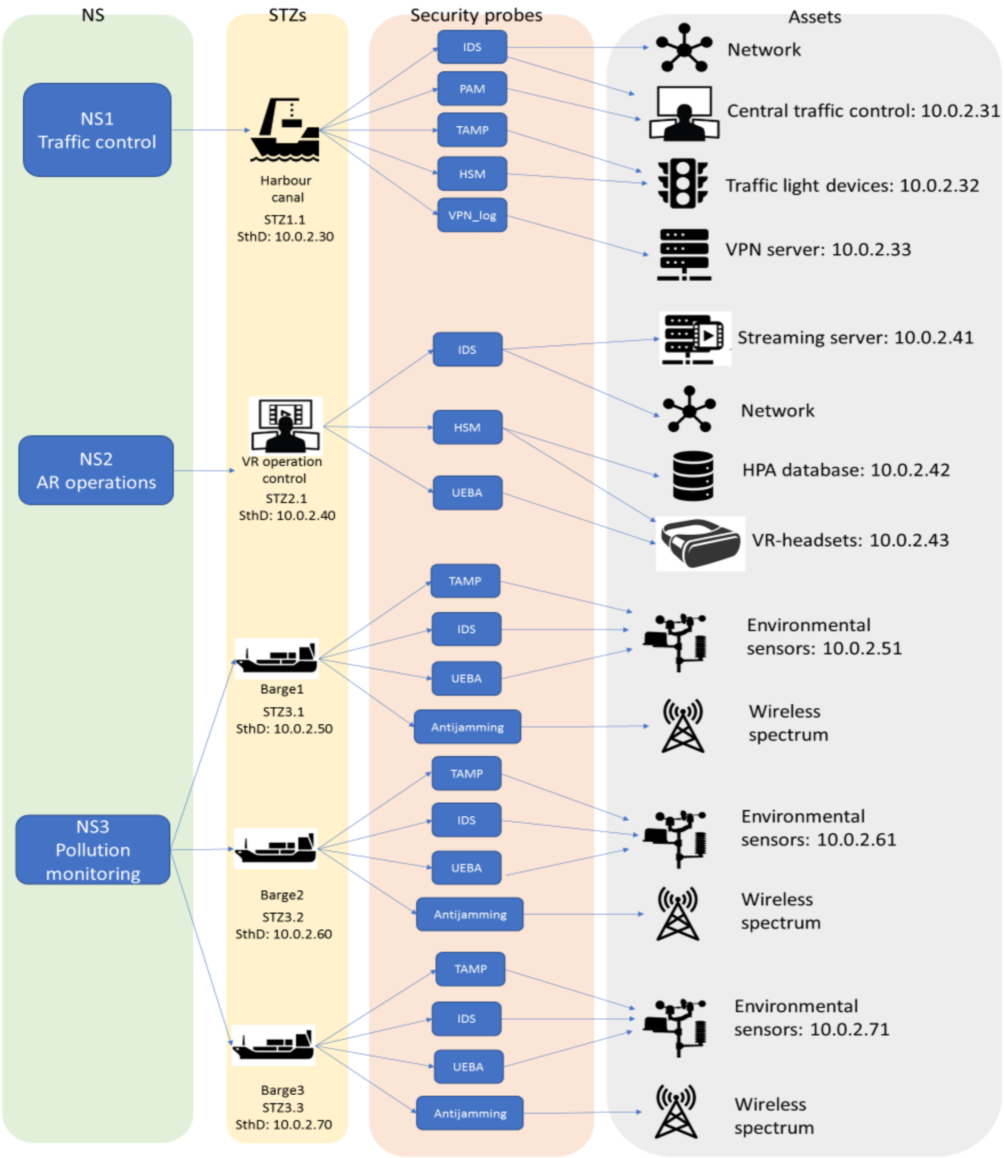


Figure 3-27: Deployment of SthD simulated assets and probes in the testbed

[9] Obviously, omitting SthP (prevention) and R (reaction) prevents from testing prevention and reaction capabilities; however, the target of the test was to test the detection capabilities in a STZ based deployment.

SMm adaptation

The final step requires to adapt the SMm to the setup of the testbed. Among other activities, the SMm needs to know about the number of STZs for each NS, the IP address of the security components deployed (i.e., the SThD), the number and type of security probes available in all the STZs and information about the assets deployed in the NS. Additionally, and depending on the type of security probes available, the SMm would require several customisations to define new correlation rules to process different type of events from different security probes, and to interpret the information received from them. The SMm also interprets the criticality of the assets deployed in order to report about security alerts with a quantitative risk value representing the impact of the security incident detected. For the simulated testbed deployed the ATOS XL-SIEM tool was used[10], a Security Incident and Event Management tool that provides with the SMm functionalities. The ATOS XL-SIEM was adapted to support the STZ approach for 5G networks, incorporating the required configuration to support the testbed deployed as shown in Figure 3-28.

```
user@kali:~$ java -jar attacker5G_Seaport.jar
Please, specify of attacks as arguments from the following list:

          ATTACKS AGAINST TRAFFIC LIGHT CONTROL FACILITY

traffic_tampering         - Trigger tampering attack against a traffic light device
traffic_hsm_bruteforce    - Trigger bruteforce attack against HSM at traffic light device
traffic_hsm_mim           - Trigger man in the middle attack at traffic light by modifying message integrity with HSM device
traffic_vpn_encryption    - Trigger a VPN connection with a weak encryption
traffic_vpn_dos           - Trigger a DOS attack against the VPN server by establishing too many connections
traffic_vpn_bruteforce    - Trigger a Bruteforce attack against the VPN server
traffic_vpn_manipulation - Trigger a settings manipulation attack with an incorrect configuration for the VPN connection

          ATTACKS AGAINST VIRTUAL REALITY OPERATIONS

vr_ueba             - Trigger anomalous behavior of VR headsets
vr_hsm_bruteforce   - Trigger bruteforce attack against HSM at VR headset
vr_hsm_mim          - Trigger man in the middle attack at VR headset by modifying message integrity by using the HSM

          ATTACK AGAINST ENVIRONMENT MONITORING AT BARGES

barge_antijamming - Trigger Wide Band jamming attack at a pre-defined frequency against barges

  ATTACK AGAINST BARGE 1
barge1_ueba         - Trigger anomalous behavior of environment monitoring sensors in barge 1
barge1_tampering    - Trigger tampering attack against environment sensors in barge 1

  ATTACK AGAINST BARGE 2

barge2_ueba         - Trigger anomalous behavior of environment monitoring sensors in barge 2
barge2_tampering    - Trigger tampering attack against environment sensors in barge 2

  ATTACK AGAINST BARGE 3

barge3_ueba         - Trigger anomalous behavior of environment monitoring sensors in barge 3
barge3_tampering    - Trigger tampering attack against environment sensors in barge 3
```

Figure 3-28: Attacks simulation available for the simulated Smart Sea Port testbed

3.6.4 Security system simulation tests

This section summarises some of the tests performed against the simulated Smart Sea Port testbed with the STZ and the network slice setup described in the previous sections. The tests consist in a set of attacks, simulated or real, carried out against the testbed. Table 3-2 lists the attacks considered, simulated or real.

Table 3-2: Set of attacks for the simulated Smart Sea Port testbed

Attack	Simulated/Real	Tool used
Denial of service	Real	nping3, Suricata
Service discovery	Real	nmap, Suricata
Brute force attack	Real	ncrack, Suricata
Device tampering	Simulated	script
Man in the middle	Simulated	script
Unauthorised VPN connection	Simulated	script
Anomalous UE behaviour	Real	CERTH software
Jammer attacks	Simulated	script

[10] The ATOS XL-SIEM tool is an internal asset from the Atos Research and Innovation department.

Among the real attacks, three of them (Denial of service, Service Discovery and Brute force attack) were performed using different tools (hping3 [Lin19], nmap [Nma19] and ncrack [Nma19a] respectively) available at the Kali Linux distribution, which is a specialised Linux distribution with tools preinstalled to carry out penetration testing [Kal19]. The detection of these attacks was carried out using the Suricata NIDS [Sur19], which was installed in the testbed. Details of this type of incident and the corresponding detection are included in the 5G-MoNArch Deliverable D3.1 [5GM18-D31].

For the rest of the attacks it has been created a specific script which simulates the probes to detect those attacks, issuing the events that these probes would send for each incident. Figure 3-28 represents a screenshot of the different options available for that script. The script also includes the possibility to simulate anomalous UE behaviours incidents, which speeded up the integration activities.

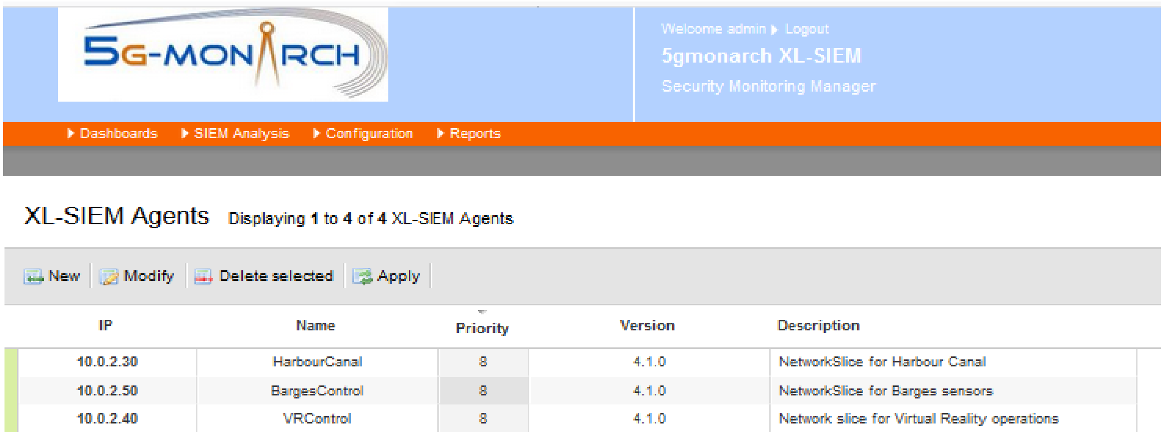


Figure 3-29: SthDs configured at the SMm for the simulated Smart Sea Port testbed

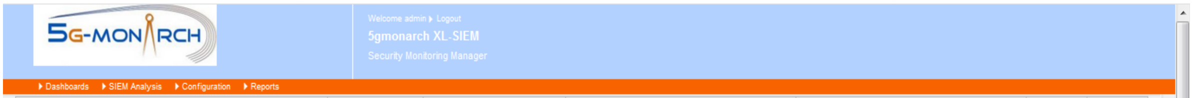
Also, Figure 3-29 represents a screenshot of the ATOS XL-SIEM tool that acts as SMm for the testbed. More specifically the screenshots show the three SthD configured for the three network slices used in the testbed (the column IP represents the IP of the SthD; the column Name represents the name given to the SthD); in this screen we have:

- HarbourCanal: the SthD used for the network slice that manages the "better traffic flow" use case.
- VRControl: the SthD used for the network slice that manages the "enhanced maintenance experience" use case.
- BargesControl: the SthD used for the network slice that manages the "improved pollution control" use case. In this network there are three STZs (one per barge) but only one SthD is included to optimise the testbed resources.

Figure 3-30 depicts the events received by the SMm after simulating several attacks against several assets of the STZs available in the testbed. The screen shows one event per row with different columns; they are:

- Signature: textual description of the type of event received.
- Date: the timestamp when the event was received.
- Sensor: the name of the SthD that received the event. In the context of this simulation, this field represents the STZs where the incident is happening. It is worth reminding that in this simulation we have reused the SMm for the three network slices available. However, the features of the ATOS XL-SIEM tool allows to isolate the data received from different SthD, which allows to generate alerts separately, and not mixing events from different STZs.
- Source: the source IP included in the event. In general terms this IP represents the attacker but depending on the event this field would simply represent an estimation of the origin of the attack with coarse grained granularity. For example, in the case of a denial of service attack the event contains the source IP sending the flooding packets. In the case of events representing jammer attacks it is impossible to know such a detailed information as an IP of

the attacker, because in that type of attacks the attacker does not need to be connected to the wireless network and therefore, the attacker can interfere frequencies without being revealed. In this case the jammer attack event would be received by the SMm but the field source IP would show just the IP of the SthD that received the event, the IP of the subnet affected or simply a "N/A" (the specific value would depend on the system admin preferences).



Signature	Date GMT+2:00	Sensor	Source	Destination	Asset S → D	Risk
Detected tampered device	2019-05-03 12:44:06	BargesControl	10.0.2.50	Barge3Environmental	5->3	10
Detected tampered device	2019-05-03 12:44:03	BargesControl	10.0.2.50	Barge2Environmental	5->3	10
5G behavioral analysis: service anomaly	2019-05-03 12:44:03	BargesControl	10.0.2.50	Barge3Environmental	5->3	6
Detected tampered device	2019-05-03 12:44:00	BargesControl	10.0.2.50	Barge1Environmental	5->3	10
5G behavioral analysis: service anomaly	2019-05-03 12:44:00	BargesControl	10.0.2.50	Barge2Environmental	5->3	6
5G behavioral analysis: service anomaly	2019-05-03 12:43:57	BargesControl	10.0.2.50	Barge1Environmental	5->3	6
Anomaly: Wide Band Jammer detected	2019-05-03 12:43:55	BargesControl	10.0.2.50	Barge3Environmental	5->3	6
Message integrity failure at HSM	2019-05-03 12:43:47	VRControl	10.0.2.40	VRHeadset	5->3	5
Password authentication failure at HSM	2019-05-03 12:43:45	VRControl	10.0.2.40	VRHeadset	5->3	4
Password authentication failure at HSM	2019-05-03 12:43:43	VRControl	10.0.2.40	VRHeadset	5->3	4
Incorrect VPN configuration detected	2019-05-03 12:43:36	HarbourCanal	TrafficLights	VPNServer	3->2	2
Failed VPN connection attempt	2019-05-03 12:43:33	HarbourCanal	TrafficLights	VPNServer	3->2	4
Failed VPN connection attempt	2019-05-03 12:43:31	HarbourCanal	TrafficLights	VPNServer	3->2	4
Failed VPN connection attempt	2019-05-03 12:43:29	HarbourCanal	TrafficLights	VPNServer	3->2	4
High number of VPN connections	2019-05-03 12:43:27	HarbourCanal	TrafficLights	VPNServer	3->2	2
Weak Encryption at VPN connection	2019-05-03 12:43:24	HarbourCanal	TrafficLights	VPNServer	3->2	3
Message integrity failure at HSM	2019-05-03 12:43:22	HarbourCanal	10.0.2.30	TrafficLights	5->3	5
Password authentication failure at HSM	2019-05-03 12:43:19	HarbourCanal	10.0.2.30	TrafficLights	5->3	4
Password authentication failure at HSM	2019-05-03 12:43:17	HarbourCanal	10.0.2.30	TrafficLights	5->3	4
Detected tampered device	2019-05-03 12:43:15	HarbourCanal	10.0.2.30	TrafficLights	5->3	10
Password authentication failure at HSM	2019-05-03 12:43:15	HarbourCanal	10.0.2.30	TrafficLights	5->3	4
High number of VPN connections	2019-05-03 12:42:37	HarbourCanal	TrafficLights	VPNServer	3->2	2
Weak Encryption at VPN connection	2019-05-03 12:42:35	HarbourCanal	TrafficLights	VPNServer	3->2	3
Message integrity failure at HSM	2019-05-03 12:42:32	HarbourCanal	10.0.2.30	TrafficLights	5->3	5
Password authentication failure at HSM	2019-05-03 12:42:30	HarbourCanal	10.0.2.30	TrafficLights	5->3	4

Figure 3-30: Events generated during tests in the security simulation of the Smart Sea Port testbed

- Destination: same as for the source IP, this field contains the IP of the targeted asset. This information is also extracted from the event received. Additionally, and same as for the source IP, in some cases the granularity of the Destination field would be more or less fine depending on the type of event. For example, the destination of a denial of service attack is a specific IP, while the destination of a jammer attack would be a subnet in the best of the cases.
- Asset S→D: represents the importance of the source and destination assets represented by their IPs. The importance level is configured by the system admin and is used by the SMm to compute the risk associated to the event received (this parameter considers both: the importance of the combination and the importance of the targeted asset).
- The risk column represents a quantitative value representing the importance of the event. The algorithm used by the SMm uses the assets importance mentioned above and additional data obtained from the events. A higher value represents a higher risk.

Not all events derive into actual incidents. The SMm contains a set of rules to analyse the received events and trigger alerts in case of any malicious incident detected. A set of security alerts were generated for the set of events executed during the simulation carried out. Figure 3-31 represents a screenshot of the alerts produced. For triggering each alert, it might have been required to correlate one or more events received from the same or different security probes. In this case the meaning of the different columns is as follows:

- "Signature", represents a textual description of the alert generated.
- "Events", indicates the number of events correlated to generate such alert.
- "Risk", indicates the importance of the alert, computed in a scale from 0 to 10 (being 0 the less important and 10 the higher). To get this value it used data such as the risk associated to the events used to correlate the alert, and also the importance of the assets affected.
- "Duration", indicates the duration of the attack that derived in the alert.
- "Source" and "Destination", indicate (if available) information about attacker and target.

- “Status”, indicates whether the alert has been solved (closed) or if the incident is yet to be solved (open).

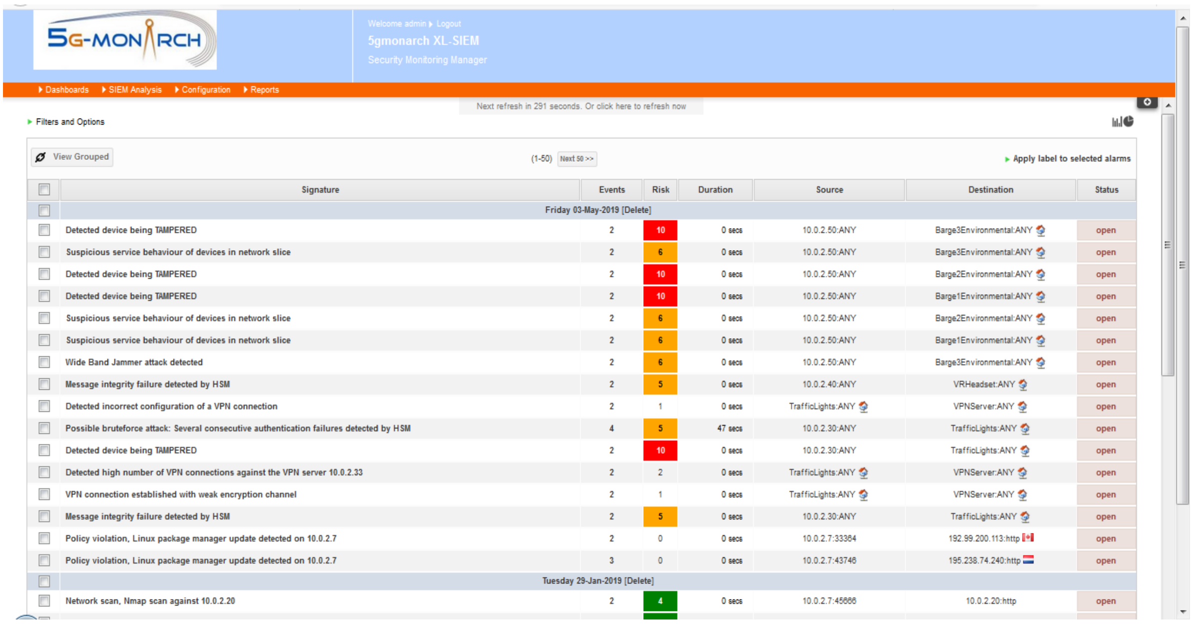


Figure 3-31: Alerts generated during tests in the security simulation of the Smart Sea Port testbed

The tests carried out simulating the sea port infrastructure has proven the possibility to apply such solution to network slice-based environments. The general approach was demonstrated and validated in WP3. In this task it was applied to a specific testbed like the Smart Sea Port, with specific threats and assets, which also demonstrated its validity and showed the flexibility of the STZ-based approach for securing network slices.

4 Touristic City testbed

4.1 Introduction

The Touristic City testbed represents a use case of future advanced multimedia and entertainment services related to touristic places such as museums, historical buildings, art galleries and others. In the context of the 5G, this kind of services are referenced as digital tourism services. These type of services (together with IoT, automotive and Industry 4.0 services) provides new business opportunities for MNOs that go beyond the traditional mobile broadband applications and target the so-called adjacent market of verticals. To meet the requirements of such vertical markets, 5G solutions do not only need to improve capacity and throughput performances but, in addition to that, they also need to provide very low latencies and high reliability. Furthermore, to provide the required services efficiently, it is essential to implement the necessary elasticity at the infrastructure side, to adjust the consumption of computing and network resources to the available resources without harming the resulting quality. In this way, it is possible to avoid overprovisioning resources and thus minimise the cost involved in providing this service.

In the context of 5G vertical services, the objective of the Touristic City testbed is to demonstrate the benefits of the 5G solutions developed by 5G-MoNArch to provide an immersive and interactive experience to the users in the context of digital tourism service. Such solutions include: (i) network slicing, to provide the custom requirements needed for VR, (ii) Mobile Edge Computing (MEC), to satisfy the latency requirements of interactions between the users, and (iii) resource elasticity, to provide an efficient management and orchestration of resources.

The Touristic City testbed is deployed in the city of Turin, where a large number of visitors are present throughout the year. The specific location chosen for the testbed deployment is Palazzo Madama [Pal19], one of the most representative monumental buildings of Piedmont, located in downtown Turin. The usage of the spaces of Palazzo Madama has been granted by the Turin Municipality [Tur19] in collaboration with the Fondazione Torino Musei consortium [Fon19] which take part to the Touristic City testbed activities as vertical stakeholders on voluntary basis, showing how the 5G-MoNArch solutions have a big potential in providing new innovative approaches and opportunities for the touristic sector, contributing to its growth.

Fondazione Torino Musei plays a key role in the testbed as the vertical customer of the 5G technology. As a matter of fact, Fondazione Torino Musei is interested in exploring of the possibilities that a technology that allows multi-user access to the contents can offer to the visitors of Palazzo Madama. In addition to that, there is also the interest to understand and experience VR applications for visitors who cannot physically reach the museum such as tourists or students who live far away, and also people with physical disabilities. For example, the VR technology could be used for people having a cognitive disability to get them used to the visit experience before going physically to the museum. Fondazione Torino Musei is also providing their expertise in the analysis and the development of the contents, exchanging opinions about the developments of the VR application in the context of a museum visit and the related challenges.

This Chapter details the implementation and the integration steps, along the KPI evaluation and measurement results of the Touristic City testbed. The rest of the chapter is structured as follows: Section 4.2 provides a description of the use case. The implementation of the three major components of the testbed (5G RAN 5G Core, and VR application) is described in detail in Section 4.3: (i) the implementation of RAN part including PHY/MAC and high layers is provided in subsection 4.3.1, (ii) the implementation of the 5GC part including the orchestrator, network function virtualisation (NFV) and AI algorithms is described in subsection 4.3.2, and (iii) the VR application is presented in subsection 4.3.3 including the explanation of how the VR environment is created, as well as the avatars development and the user interaction. In section 4.4 the overall setup and integration of the different testbed parts is described. Finally, the KPIs and the performance results of the integrated testbed is summarised is section 4.5.

4.2 Use case

From the specific use case point of view, the Touristic City testbed provides an interactive VR visit of the Madama Reale chamber, one of the most relevant rooms of Palazzo Madama. The end-user has the possibility to interact with a real touristic guide (i.e., a physical person) through their avatars in a VR environment where the touristic guide accompanies the tourist in the virtual visit of the Madama Reale chamber, as well as to involve him in specific activities such as thematic tutorials or instructional games. On this aspect, the VR application has been developed based on the input provided by the Palazzo Madama's restorer in charge of the wooden parts and artefacts, in order to implement a restoring tutorial in which the tourist can be involved (see Section 4.3.3.3 for further details). We consider this is a relevant use case to be implemented using 5G technologies because it requires both: real time video streaming transmission (by means of a specific eMBB network slice) and low latency responses (also using a dedicated URLLC slice to react in real time to the user interactions).

The Touristic City testbed use case consists in the following three phases:

Madama Reale chamber tour

The users have a unique realistic experience visiting a VR museum with its detailed artefacts and the actual interior of the museum. At first, a tutorial room is provided in order to describe button mapping and the environment. After selection of the language, the user is transported in a tutorial room. The first steps regarding movement are introduced through informative gifs. In the centre of the room is a ticket booth in which user has to choose role and gender. In this specific phase a second user has always the role of virtual guide in order to help the new user with the procedure. Therefore, when a new user enters the app, there are only two options on ticket booth's screen: male or female tourist. A VoIP server is enabled so both users can talk each other. Then, a ticket is printed with a photo of selected role on it. The user must use this ticket in ticket validator machine in order to proceed to Madama Real Chamber. Both users are transported to their start points inside Chamber room. The video stream from camera is now projected and the user has the feeling that is inside the real room. At first only one artefact is shown and a table with some tools over it.

Restoration scenario

After the transportation to the chamber room the guide asks the user to perform a restoration procedure in a damaged artefact. A tool table provides all the equipment the user needs in order to complete the procedure. Flashing indicators accompanied with informative gif over each object are enabled on its turn to use. The guide is still present in order to explain the procedure and to help user in case cannot complete some task. He is able to pass the objects to the tourist if tourist asks for it. Regarding the scenario, first the user has to remove the damaged part from the artefact with a scrapper. Then using a new bigger wooden piece and a chainsaw, the user cuts a similar sized wood. The new piece needs to be varnished and glued with a brush before placed in the artefact. When the user completes the procedure the rest of the artefacts appear in their position as 3D objects.

Migration to the edge cloud

As mentioned before, the VR application considers a situation in which the tourist user has to cut a wood in a saw. For this task, since the introduction of a long delay makes this operation impossible, the orchestrator performs the migration of the UPF from the central cloud to the edge cloud, thus solving the delay issue (the two users will then be able to perform the action). Therefore, the user will notice the difference between "without" and "with" 5G-MoNArch innovations in a single try and the relation with the KPIs (i.e. latency measurement) will be still possible.

During the third week of May, the Touristic City testbed has been opened to the visitors of the Palazzo Madama to complement their visit of the museum. In this way, it has been possible to gather their feedback and evaluate with real users how the 5G solutions developed by 5G-MoNArch can contribute to improve the user experience in a relevant use case for 5G such as digital tourism.

4.3 Testbed implementation

An illustration of the high-level scheme of the Touristic City testbed is depicted in Figure 4-1. The system works as a client-server system based on two slices: the first slice carries the 360° video stream, while the second slice handles all the other client-server communications (e.g., multi-user interaction and 3D model registration and control, Voice over IP -VoIP-). For the 360° video stream, an eMBB slice is set up to provide the required throughput for a good video quality, while for the bidirectional communications an URLLC slice is used to guarantee the required delays for the best user experience.

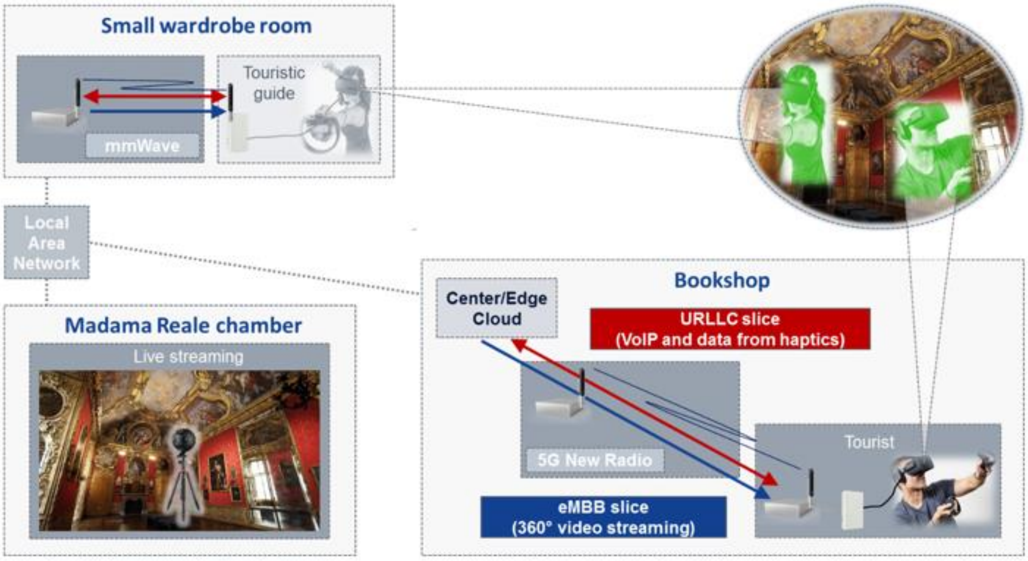


Figure 4-1: High level implementation of the Touristic City testbed

From the implementation point of view, the Touristic City testbed consists of several HW and SW modules that have been designed to provide a flexible and scalable solution for the use case described in Section 4.2. More in detail, the testbed implements a standard compliant 5G radio interface based on an SDR PHY/MAC and higher layers implementation providing the connectivity to the user (see Section 4.3.1). The flexibility of the solution in terms of resource elasticity is achieved through an orchestration module that takes advantage of the NFV technology to place VNFs over a central and edge clouds (see Section 4.3.2). Visualisation of the relevant metrics describing the performances of the testbed is provided through a GUI-based Measurement tool which is introduced in Section 4.5.1 and described in more detail in Appendix 7.6. Figure 4-2 shows the final technical scheme of the testbed as it has been implemented in the opening to the public at the end of May. In the next subsections a description of the implementation aspects of the different modules of the testbed: (i) the PHY/MAC and higher layers, (ii) the AI algorithms, orchestration and elasticity, and (iii) the VR application itself.

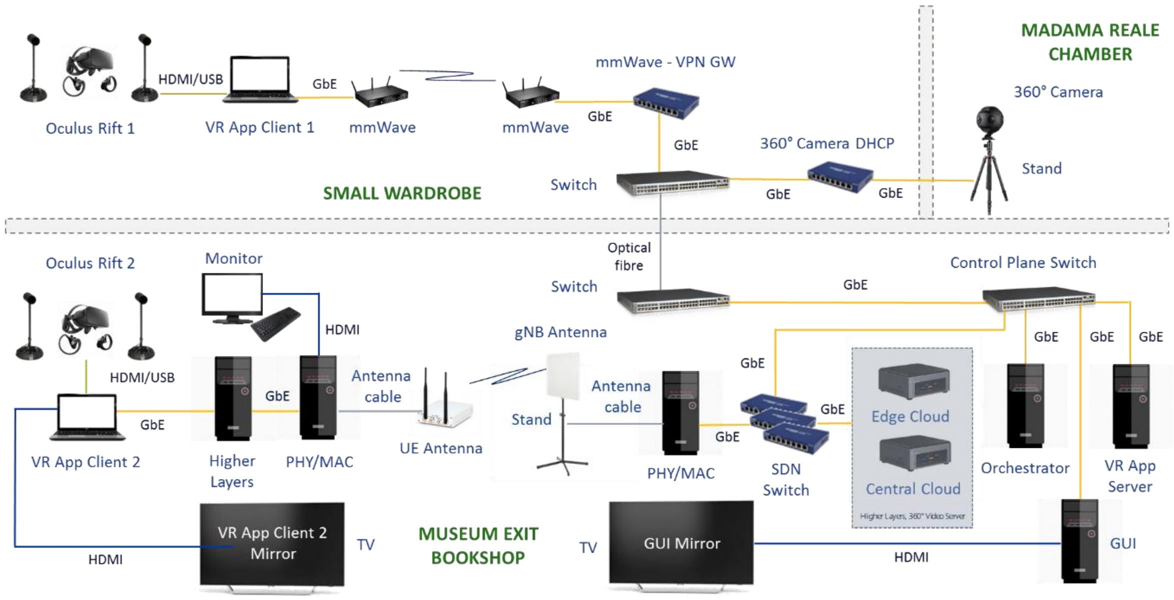


Figure 4-2: Final Touristic City testbed setup scheme

4.3.1 PHY/MAC and higher layers

The implementation of the PHY/MAC and higher layers (or Protocol Stack - PS) for both 5G base station (gNB) and UE was completed during the first phase of the project and the related details are reported in [5GM18-D51]. Regarding the higher layers, the user plane parts of the PDCP and RLC functions were implemented and connected to the machine that implements MAC and PHY functions. The PDCP and RLC functions were implemented in python, allowing their easy installation in VMs and thus facilitating their migration to different hosts when needed. Regarding virtualisation, OpenStack was used to host the VMs with those functions. In this section, more information on the PHY layer implementation is reported to highlight its compliance with the 3GPP standard on NR [3GPP18-38211]. The integration aspects are reported in Section 4.4. Figure 4-3 represents the high-level overview of the PHY/MAC modules.

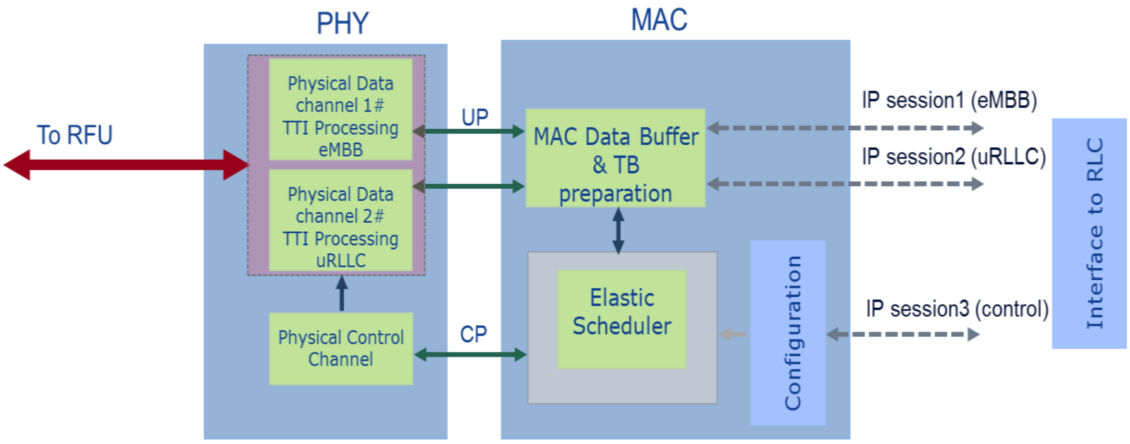


Figure 4-3: High level overview of PHY/MAC modules

The PHY of the testbed is an SDR based system with reconfigurable system parameters. The Baseband Unit (BBU) for baseband processing of PHY and MAC layers was implemented with embedded C/C++ on an Intel 0x86 platform[11]. For RF, the NI X310 USRP [Ett19b] was used. The radio interface of the testbed uses various 5G NR functionalities aligned with NR Rel.15 [3GPP18-38211]. For example, unlike LTE, both DL and UL channels of the testbed use Cyclic Prefix Orthogonal Frequency-Division Multiplexing (CP-OFDM) based waveform as described in [3GPP18-38211].

The system also supports other waveforms such as W-OFDM (windowed -OFDM, P-OFDM (Pulse shaped OFDM) that enables high reliability & low latency The PHY adopts the NR concept of multiple numerologies and multiple slot configurations. It supports multiple subcarrier spacing (15, 30, 60, 120 KHz), multiple Transmission Time Interval (TTI) lengths (1, 0.5, 0.25 ms), and flexible slot format of different UL/DL configurations. The eMBB and URLLC slices use different Bandwidth Parts (BWPs) with different numerology configurations aiming at satisfying the different requirements of both slices in terms of latency and reliability. On this basis, the testbed demonstrates multi-slice UE capability, which is also an NR feature [3GPP18-38300]. Furthermore, the system supports TDD/FDD duplexing schemes and dynamic reconfiguration of reference signal (scattered or preamble) based on channel conditions. For low PAPR, the system supports spreading. Figure 4-4 illustrates the block diagram of the physical layer of the system.

[11] This software framework is Huawei specific. The PHY modules for Tx and Rx were self-implemented, while some modules such as encoder and QAM are based on SRSLTE.

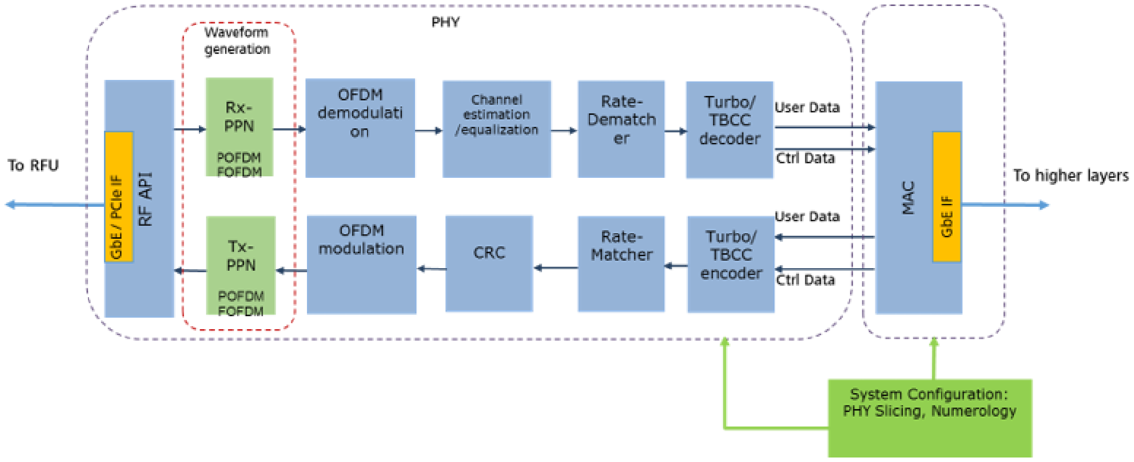


Figure 4-4: Block diagram of the baseband unit (PHY layer)

The physical data channels are configured with different numerologies (TTI length, sub-carrier spacing, BW, CP, etc.), and can be mapped to one component carrier, or to multiple component carriers to extend the bandwidth. Each slice is assigned a unique IP address as depicted in. The control channel can be assigned a different IP address or mapped to one of the slice IPs. Table 4-1 shows the chosen numerologies and PHY configurations.

Table 4-1: Numerologies and PHY configurations

Numerology/PHY configs	**eMBB**		**URRLC**	
	gNB	UE	gNB	UE
Tx power	10dBm	6dBm	10dBm	6dBm
Antenna Gain	18dBi	3dBi	18dBi	3dBi
SCS	60KHz		120KHz	
TTI length	1ms		0.5ms	
UL/DL configuration	D,U,D,U,D,U,D		D,U,D,U,D,U,D	
CP length	10%		20%	
Antenna scheme	SISO		SISO	
Carrier Frequency	3.730GHz		3.760GHz	
Duplexing mode	TDD		TDD	
MCS	16QAM/6/7		QPSK 4/5	

mmWave radio link

In order to enhance the wireless setup of the system and have a heterogeneous access network within the orchestration framework, we deployed a mmWave system to provide the guide with high bandwidth. This link consists of two Mikrotik [Mik19] RBwAPG-60ad-A wAP 60GHz devices. From extensive tests we achieved between 500 and 900 Mbps over distances around 100-150 meters, which are more than sufficient to meet the very high bandwidth requirements of our setup. In order to integrate this element into our architecture, we implemented a non-3GPP access gateway that provide seamless connectivity to the rest of the network (i.e., the VR server or the 360-video server).

4.3.2 AI algorithms, orchestration and elasticity

The touristic city testbed is the showcase of the elasticity algorithms developed in [5GM19-D42] in a real scenario comprising real users (namely, the visitors of the museum). Therefore, in addition to the specific VNFs that have been developed for the testbed, a very major role is played by the management and orchestration software that has to manage the different components of the testbed and provide the required orchestration functionality on top of the underlying virtualisation infrastructure. In particular, as discussed in [5GM19-D42], we employ AI techniques for the management and orchestration of the network.

Besides the showcase of specific 5G-MoNArch solutions, the Touristic City testbed also contributes to demonstrate the ETSI Experiential Network Intelligence (ENI) [Eni17] concepts as a viable technology for the improvement of telecommunication networks. Specifically, this testbed serves to prove the applicability of the techniques described in ETSI ENI Use Case #2-6 [ETSI18-GRENI001], named as “Elastic Resource MANO”. Namely, the main goal of the Touristic City testbed consists in showing the feasibility and the benefits of an AI-assisted “elastic” management and orchestration of the network, which entails an improvement of the network efficiency and its capability to smoothly adapt the resource allocation and utilisation. The public demonstration of these ENI concepts, based on an architectural framework compatible with ENI Reference Architecture ETSI GS ENI 005 [Eni18], will contribute to build commercial awareness and confidence in the ENI approach.

In the above context, the Touristic City testbed demonstrates how ENI can be used to integrate and improve the resource and NF MANO, including slice admission control and deployment, horizontal and vertical scaling of VNFs, and their migration between different nodes of the network. Due its very high innovation potential and for the availability of different AI algorithms in the proposed MANO (see [5GM19-D42] for more details about them), the Touristic City testbed has been selected as Proof of Concept (PoC) for the ETSI ENI ISG [Eni19]. The evaluation committee of the PoC saw very high potential in this testbed setup, not only for the enhanced KPIs that can be provided by the 5G-MoNArch technology, but also for the hands-on experience that is leveraged for extending the state-of-the-art MANO architecture with AI parts.

As discussed at the beginning of this chapter, the Touristic City testbed features two different network slices: (i) an eMBB slice that serves 360° videos to a VR device, and (ii) an URLLC slice (used also by the VR device) that provides voice connectivity and haptic interaction with certain avatars in the virtual scenario. From the physical infrastructure point of view, the testbed includes a set of PNFs to implement the radio lower layers and a cloud infrastructure (composed by a farther from the radio but highly capable central cloud and a closer but less capable edge cloud). A high-level scheme of the testbed HW and SW setup is depicted in Figure 4-5. More in detail, besides the VR application, the testbed consists of the following set of functionalities, which are detailed next:

- Network Slice Blueprinting and Onboarding;
- VNF Relocation due to latency;
- Horizontal and Vertical VNF Scaling (AI-assisted);
- Admission Control (AI-assisted).

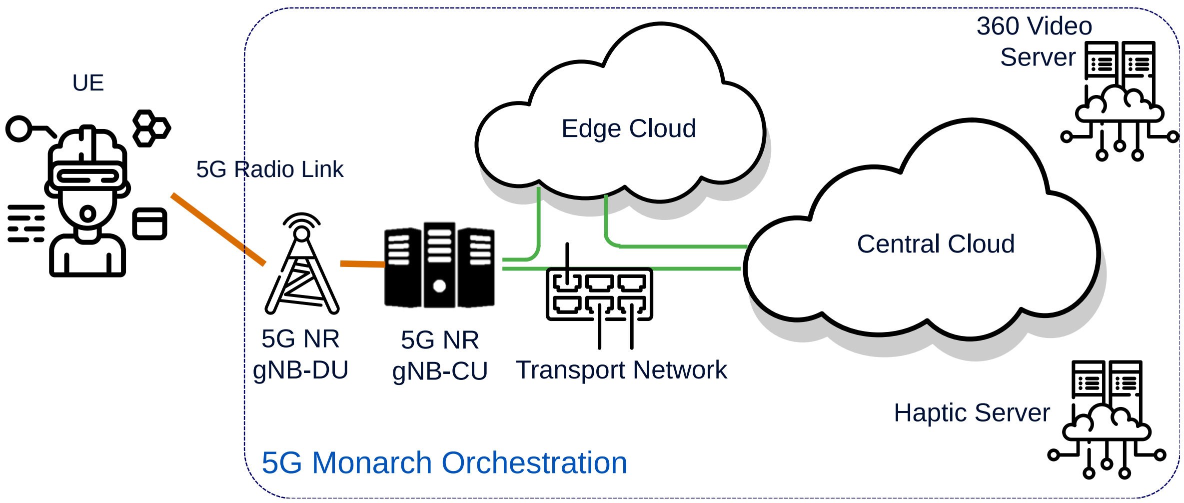


Figure 4-5: Testbed HW and SW setup

4.3.2.1 Network orchestration setup

In order to implement the features envisioned by the testbed architecture described above, we rely on an orchestration architecture based on the Open Source MANO (OSM) orchestrator Release Four [Osm19] and the OpenStack VIM [Ope19].

OSM is one of the leading solutions for the implementation of a fully-fledged mobile network orchestrator, including several components for the automatic LCM. Still, to provide the enhanced functionality needed, substantial changes to the OSM code and architecture are provided.

Figure 4-6 shows the OSM architecture, composed by several modules that carry out the different functionality needed by an orchestrator. However, it was also necessary to implement certain additions for supporting the 5G-MoNArch specific features (they have not been pushed towards the OSM codebase, but they will be useful to understand the implications of the usage of ETSI ENI in conjunction with ETSI NFV MANO); they are the following:

Resource Orchestrator: this module provides the hooks towards different VIMs supported by OSM. In the context of the testbed, we use OpenStack as the main VIM to manage the local Network Function Virtualisation Infrastructure (NFVI) deployment. However, we leverage also on other APIs: The Amazon AWS EC2 [Ama19] (to demonstrate the feasibility of a large-scale deployment over public clouds) and the Kubernetes one [Kub19] (to include container-based VNFs).

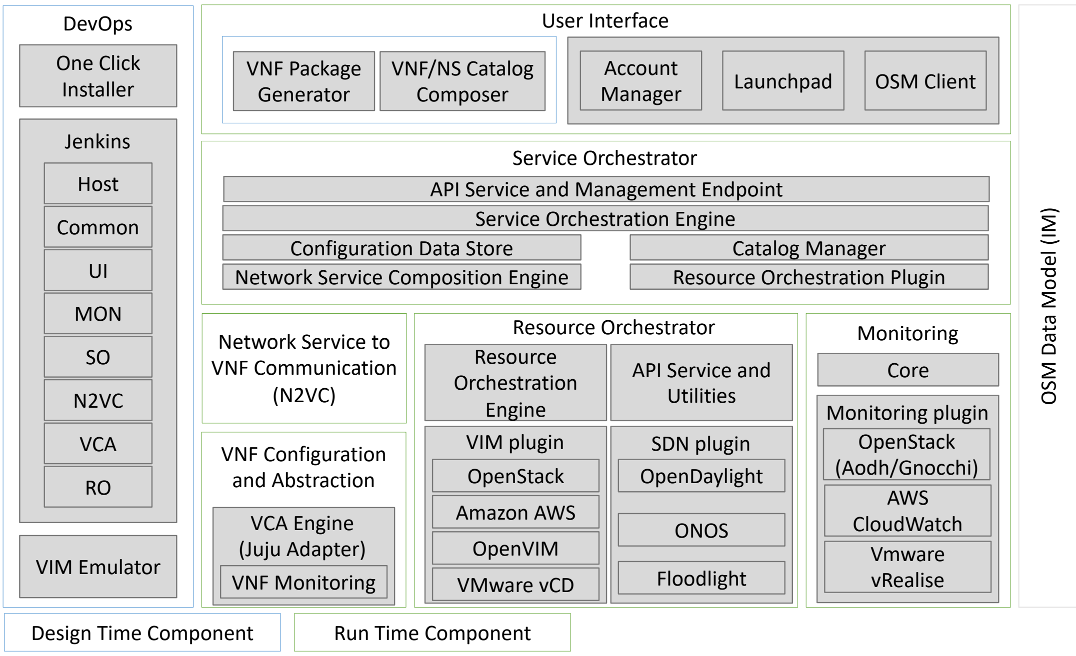


Figure 4-6: OSM architecture [Osm19]

Network Service to VNF communications: we extend this Application Programming Interface (API), that is used to configure the VNFs in a chain, to allow more specific configurations (such as the ones needed by the radio ones, which are PNFs) and support the VNF relocation.

Network slice blueprinting: the 5G-MoNArch Network slice blueprints and templates must be translated into descriptors that can be used by the OSM internal modules. For instance, certain parameters present in our blueprint were not explicitly managed by OSM.

5G-MoNArch orchestration module: the specific algorithms such as the Admission Control and the Horizontal Vertical Scaling (see Section 4.3.2.4) reside on top of the Service Orchestration module, to use the API specifically designed for that purpose. This module is a specific add on to OSM, tailored to the purposes of the testbed.

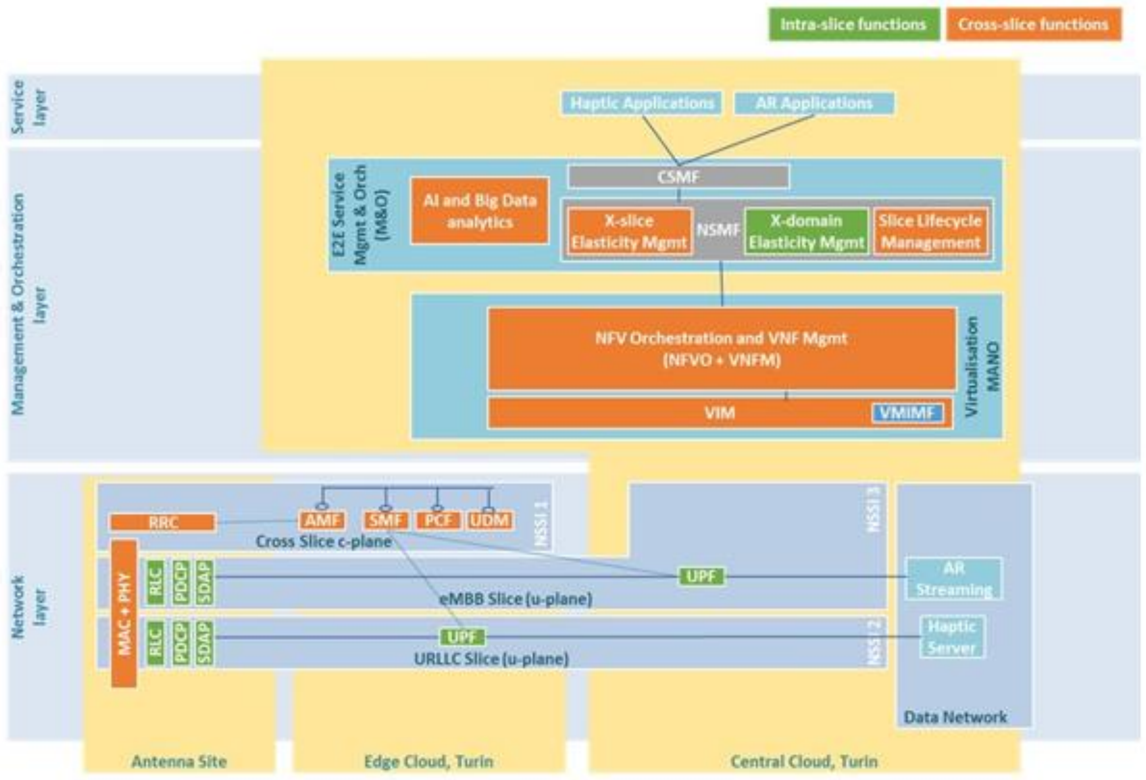


Figure 4-7: Targeted functional architecture for the Touristic City testbed [5GM18-D22]

4.3.2.2 Network functions virtualisation

The Touristic City testbed target architecture instantiation is depicted in Figure 4-7 that shows the NFs in each layer, i.e., Network layer, M&O layer, and Service layer. The NFs of this testbed are deployed in three main locations (depicted as yellow boxes in Figure 4-7): the antenna site (where the radio PNFs are executed), the edge cloud, and a central cloud (with a higher latency to the UE). In practice, the edge and the central cloud are two dedicated cloud infrastructures, connected via fibre to the antenna site. The central cloud emulates a farther processing site, with an increased latency (artificially emulated in the testbed) but a lower operational cost. Both sites are deployed in the premises of the demo but are owned by the operator.

In the Network layer, the testbed implements the two network slices mentioned before: for eMBB communications and for URLLC, which are used to provide the two different services: the high-resolution video streaming and the haptic server connecting the VR avatars for their interactions. They have the following deployment features:

- **eMBB network slice**: The eMBB network slice delivers the high resolution 360 video to the mobile user. In the RAN, the slice uses the common PHY and MAC layers of the testbed radio infrastructure, while the higher layers are slice-specific due to customisations reflecting specific service requirements. The RRC instead, is common to both slices. The CP functionality is shared across slices, while the UPF is dedicated to each slice. In terms of deployment, the CN NFs are deployed in the central cloud, as well as the UPF. Also, the application server runs in the central cloud.
- **URLLC network slice**: The URLLC network slice is used for delivering the low latency (see the application requirements in Section 4.5.2) haptic interactions among the VR avatars. The radio deployment is equivalent to the eMBB network slice. Also, the CN NF setup is similar in terms of sharing and deployment. However, the UPF may be moved from one cloud to the other according to the specific load of the network, according to the inputs coming from the elasticity modules deployed in the NFVO, as discussed in Section 4.3.2.1.

These NFs are managed by an ETSI compliant framework, as discussed above and detailed in the next section.

4.3.2.3 Orchestration and elasticity

The diagram represented in Figure 4-8 shows the framework of the testbed based on the well-known ETSI NFV [ETSI14-GSNFV001] and 3GPP MANO architectures [3GPP18-28530] that are also building the overall 5G-MoNArch architecture. We populated the specific modules of the architecture with our specific algorithms, as specified next.

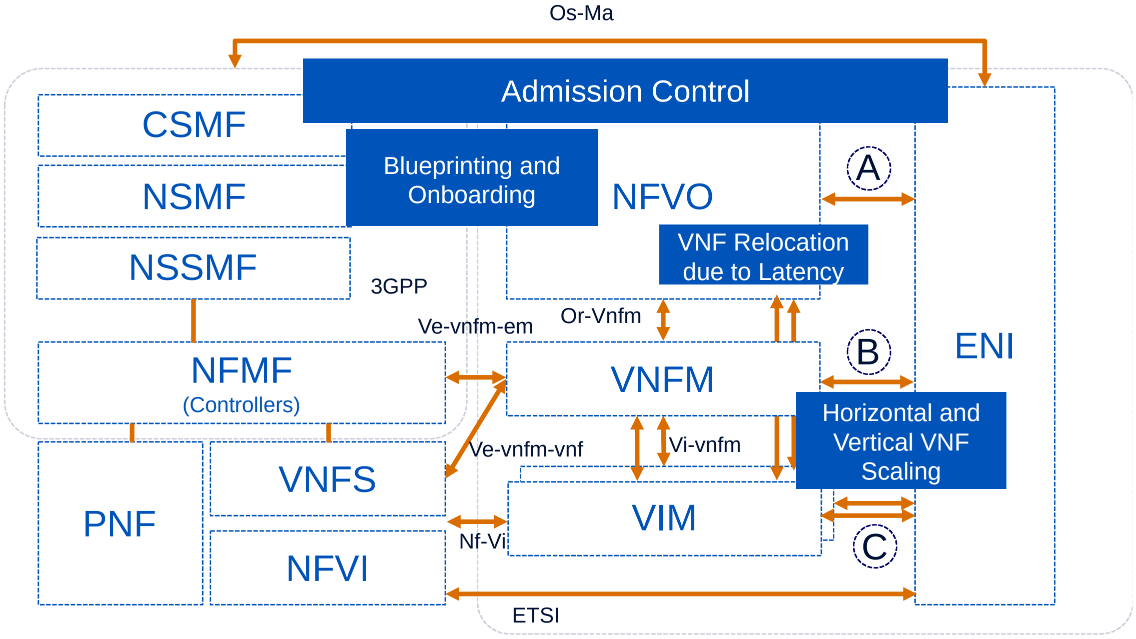


Figure 4-8: Touristic City testbed framework

Network slice blueprinting and onboarding: this functionality encompasses all the baseline operation available in a 5G MANO system. The automatic onboarding of the two network slices is a seamless operation that allows (i) verticals to define the requirements associated with each slice (in this case, high bandwidth for the eMBB and very low latency for the URLLC) and (ii) the deployment of the VNF in the cloud to fulfil the set of requirements. This part has specific elements of novelties for the blueprinting. Specifically, the network slice templates that are currently being investigated by 3GPP and Next Generation Mobile Networks (NGMN), and their practical implementations with technologies such as Topology and Orchestration Specification for Cloud Applications (TOSCA) [Oas19] or OpenStack Heat [Wik19] are extended and integrated with specific elasticity fields. While more details on this topic can be found in [5GM19-D23], we discuss here the implementation details of the extension related to the UPF network descriptor. The definition of VNFs in OSM is managed through descriptors written in YAML that define fundamental elements such as the ones defined below. More specifically, two kinds of descriptors have to be created: one for the virtual networks (i.e., the virtual links that span one or more VM and connect VNFs among them) called Network Slice Descriptor (NSD) and one for the virtual machines (our implemented option, but it can also work for containers) that run on top of them, Virtual Network Function Descriptor (VNFD). The former descriptor is the one that has been modified by 5G-MoNArch to include the new parameters defined in WP2, as detailed next in Figure 4-9 and Figure 4-10.

```
vld:
        # Networks for the VNFs
        - id: management
          name: management
          short-name: management
          type: ELAN
          mgmt-network: "true"
          vim-network-name: public
          # provider-network:
          #     overlay-type: VLAN
          #     segmentation_id: <update>
          vnfd-connection-point-ref:
            # Specify the constituent VNFs
            # member-vnf-index-ref - entry from constituent vnf
            # vnfd-id-ref - VNFD id


            # NGINX
            - vnfd-id-ref: cirros_vnfd_sl1
              member-vnf-index-ref: "1"
              vnfd-connection-point-ref: vnf-mgmt1
              ip-address: 192.168.200.185
              # 245_UPPER_LAYERS
            - vnfd-id-ref: cirros_vnfd_sl1
              member-vnf-index-ref: "1"
              vnfd-connection-point-ref: vnf-mgmt2
              ip-address: 192.168.200.186
              # 245_UPF
            - vnfd-id-ref: cirros_vnfd_sl1
              member-vnf-index-ref: "1"
              vnfd-connection-point-ref: vnf-mgmt3
              ip-address: 192.168.200.187
```

Figure 4-9: Snippet of an NSD file descriptor

```
        - id: upf
          name: upf
          description: UPF
          count: 1
          # Flavour of the VM to be instantiated for the VDU
          vm-flavor:
            vcpu-count: 1
            memory-mb: 512
            storage-gb: 5
          # Image including the full path
          image: "ubuntu"
          interface:
            # Specify the external interfaces
            # There can be multiple interfaces defined
            - name: eth0-mgmt3
              type: EXTERNAL
              virtual-interface:
                type: VIRTIO
              external-connection-point-ref: vnf-mgmt3
              position: 1
            - name: eth1-in10_0_0_3
              type: EXTERNAL
              virtual-interface:
                type: VIRTIO
              external-connection-point-ref: in10_0_0_3
              position: 2
            - name: eth2-in11_0_0_3
              type: EXTERNAL
              virtual-interface:
                type: VIRTIO
              external-connection-point-ref: in11_0_0_3
              position: 3
```

Figure 4-10: Snippet of a Network file descriptor

VNF relocation due to latency: after the first step, both network slices are orchestrated with all their VNFs (namely the higher layers of the RAN stack and the gNB) instantiated and running in the central cloud. The MANO system continuously collects data about the network parameters of the virtual network (i.e. latency, throughput, available and used radio resources). This is especially important for the URLLC slice, which has very stringent requirements on the E2E latency between the UE and the VR server running in the cloud (i.e. less than 80ms for perfect operation, or less than 200ms for a not ideal but still usable experience). For this reason, the delay is constantly monitored to avoid operational glitches caused by a sudden delay increase due to external factors, such as additional incoming network slices causing congestion in the radio and transport, or internal ones, like a high number of UEs connected to the VR application. In these cases, the orchestration framework triggers a relocation of the NFs and VR application to the edge cloud, to benefit from the reduced latency. For this purpose, we use a new paradigm, as discussed next.

Currently, NFs are not implemented with re-orchestration requirements in mind. As a result, once instantiated they cannot be easily relocated to a different node than the one where they are running. The only existing practical approach to relocate NFs is to run them inside a VM and perform a live migration. However, such a live migration (between different infrastructure deployments, that are not managed by the same VIM) can be very costly in terms of resources and time (can take several minutes depending on the size of the VM) causing service disruptions that are completely unacceptable for networking applications.

The main reason for the high cost of performing a live migration is the need to move the NF and its entire environment. Given that NFs and their environment can be sizable, this involves moving large amounts of data and starting a new VM in the new location, which is a complex and lengthy process. When performing the function relocation, to provide an uninterrupted service it is essential that the status of the NF before and after moving is exactly the same. However, this can be achieved by simply keeping the current state of the NF, which is much smaller than the entire NF and hence much easier to move to the new location.

Following the above discussion, we employ a novel implementation paradigm for NFs: the context-execution split, hereafter referred to as the c/e split. The key idea is to separate the *context* of a NF, consisting of the values of all the variables employed by the function, from the "engine", which we refer to as the execution part of the function. Then, when relocating a NF, it is sufficient to move the context to the new location, as the current state of the NF is contained there, so we can then instantiate the NF in the new location simply feeding it with the data corresponding to the context extracted from the previous location.

As the context represents the minimum amount of data that needs to be relocated (as it does not require the relocation of memory originally allocated to the operating system or libraries), this simplifies the relocation process very much, and allows to perform it in a very fast way. In this testbed, the c/e split is used for the relocation of the RAN higher layers and the UPF.

4.3.2.4 AI algorithms

The usage of AI-based algorithm is one of the main motivations behind the touristic city testbed deployment. Besides the undoubtable innovative value behind the implementation of such algorithms (as discussed in [5GM19-D42]), another important aspect of this activity is the insights on the definition of the architectural interfaces that are needed to support the automated decision on selected management operations such as i.e., scaling or admission control. The feedback on these activities has hence been reported to ETSI ENI, which will make use of it for the definition of e.g., the interaction between the ENI module and the ETSI NFV MANO architecture which empowers our testbed.

CPU/RAM based re-orchestration: the AI algorithm (such as the one described in [5GM18-D41], Section 4.2.2.2) running in the ENI module and the VNFM constantly monitors the cloud resources usage through specific probes, and takes orchestration decisions in case of new slices coming in. As explained in ([5GM18-D41], Section 4.2.2.2) an actor/critic algorithm used to derive the right CPU utilisation thresholds that trigger a scaling up or out of the VM running the VNFs. For example, when a new eMBB slice is coming in, the intelligent orchestration framework must take a decision about where to run the associated VNFs and what resources assign to them. Due to the scarcity of resources at the edge, the algorithm may take one of the following decisions, based on the requirements of the incoming eMBB slice and the constraints imposed by other slices:

- Horizontally scaling the higher layer RAN stack (i.e. creating a new instantiation of the running VNF) to support the additional load, if it is permitted;
- Vertically scaling the VNF, if there are resources available on the same server;
- Relocating the NF to a new location. In the latter case, the URLLC-related functions could be kept at the edge and just the eMBB ones are moved.

Admission Control: throughout the network operation, the management system is constantly running an AI/ML based algorithm, implementing selected features of the algorithms described in [5GM18-D41] (Sections 4.3.2.4 and 4.3.2.6). These algorithms take as input both the available physical resources in terms of spectrum and cloud, and the current orchestration patterns. In the scenario envisioned by the demonstrator, the algorithm considers mostly the resources available in the edge and central cloud, computing a maximum number of slices that are actually allocable into the system. Then, according to the subsequent instantiation requests, the admission control takes decision about whether accept or reject network slices, possibly triggering a scaling, as discussed above. The goal is to optimise the monetisation of the network and the overall performance, also through periodic re-orchestrations.

4.3.3 VR application

The combination of eMBB and URLLC network slices provide the connection to the VR application. As already mentioned the eMBB slice delivers the 360º video from the Madama Reale chamber, while the URLLC slice is responsible to transmit object's position and haptic interactions among VR avatars.

In order to show the need and benefit from the network services provided by the respective slices, one of the key components of the testbed is the VR application itself, which was implemented by 5G-MoNarch. The VR application's environment consists of a 360º video that envelops the work space, the artefacts (as 3D models), the avatars (that are controlled by the users) and some extra 3D models which are part of the testbed scenario.

In the following, we thoroughly describe the VR application. It is important to note that the implementation of the VR application involves some 5G specifics. For instance, some software components had to be implemented as virtualised functions, such that they can be moved close to the end-user by the orchestration framework when required due to the latency requirements. Other parts, like the haptic communication, were implemented relying on the low latencies provided by the testbed, which may not be available in other environments.

Note that in Section 4.5.2 the performance of the VR application is evaluated based on the application-layer KPIs required by this application. Such performance is achieved thanks to the service provided by the underlying network slices and could not be achieved without a network technology such as the one developed by 5G-MoNArch.

4.3.3.1 Creation of the VR environment (video server)

The first step in the development of the VR application is the proper creation of the VR environment. For that purpose, an Insta360 Pro [Ins19] camera was used to live stream from inside the Madama Reale chamber. Comparing with [5GM18-D51] a kind of different approach was chosen for the final application. One difference was that a panorama live video was chosen instead of 3D Stereo option. The main reason was the better quality, mostly due the resolution of the streaming video (in Stereo video, the top-bottom feed implies that each feed loses the half of vertical resolution). Furthermore, the final result, with the artefacts inside, doesn't show a big difference regarding the 3D stereo effects compared with the panorama video. Therefore, panorama video offers a better quality, without any significant loss in depth effect. Figure 4-11 shows the setup described above.

Another major change was the implementation of a NGINX [Ngi19] Server replacing the VLC Server that was originally used for restreaming camera's feed (the NGINX sends the HTTP Live Streaming - HLS- to the VR application). The image inside the museum, which is captured by the Insta360 Pro camera, is transferred to the NGINX Server as shown in Figure 4-12.

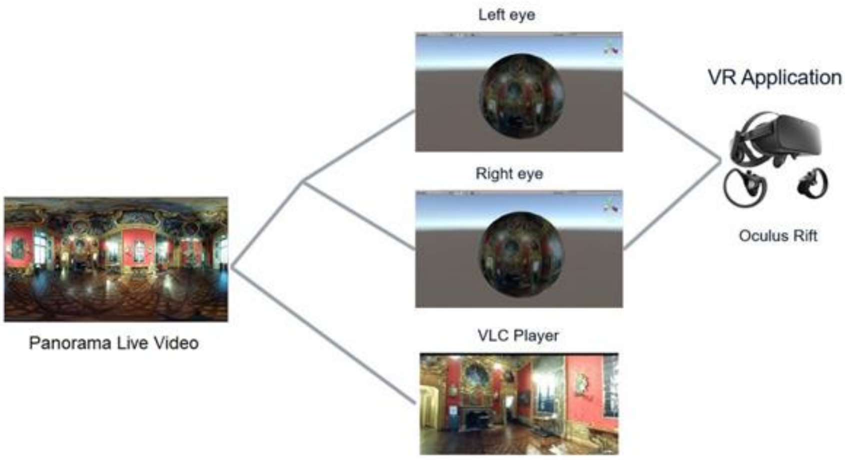


Figure 4-11: Video stream to VR application

NGINX is one of the most popular video servers due to its light-weight resource utilisation and its ability to scale easily on minimal HW. It offers a big scale of optimisation regarding video streaming, and considering the compatibility with ffmpeg transcoding, it provides a complete video server in a customisable single package. In this project, NGINX is used actually as an HLS server to provide an HLS Stream to Unity3D [Uni19].

Figure 4-12: Logical connection of Insta360Pro

An optimisation was performed in order to conform the real-time video stream to the restrictions posed by the 5G bandwidth allocated to the eMBB slice. The video recordings from Palazzo Madama was used as the main video source and new tests were performed, similar to these described in [5GM18-D51]. Some of the basic parameters were optimised was buffer and chunk size of the video stream. Figure 4-13 shows a sample of the code used for this optimisation.

As previously described, the Madama Reale chamber was projected through a 360° camera. On such basis, a 360° sphere is the projection screen, in order to try to simulate the three dimensions. However, since the room to be simulated is not a sphere in reality (it is more like a prism), some curves are distinct (especially in high corners or when approaching the walls). Furthermore, in order to complete the real environment, the artefacts in the room were added as physical objects. In [5GM18-D51] it was described how photogrammetry was used to convert these artefacts in the museum to 3D models in a form compatible with the Unity Editor. Using a video recording as a main feed, the artefacts were placed in their corresponding positions. An optimisation regarding scale and rotation of the artefacts was performed in order the user to have a better visibility beginning from the centre of the sphere. Due to the curves of the sphere a part of each model was overlapped from projection giving the sense of adapting to place.

The usage of the video recording was just to make a first implementation with 3D models to test the scale. In final testbed the camera unlikely will be placed in the exact position, so the lenses will have a different vision angle. As a result, the projection differs, and the artefacts were misplaced. To solve this problem, an editable table alignment was designed to fix the positions to the new video steam with the adequate rotation for each object.

```
rtmp {
        server {
                listen 1935;

                chunk_size 4096;
                allow play all;
                application hls {
                        allow play all;
                        live on;
                        meta copy;
                        #exec_options on;



                        hls on;
                        #hls_nested on;
                        hls_path /tmp/hls;
                        hls_fragment 3;
                        hls_playlist_length 60;
                        #deny play all;


                        #hls_variant _25kbs;

        }
```

```
http {

        sendfile off;
        tcp_nopush on;
        directio 512;
        #include mime.types;
        default_type application/octet-stream;
        #keepalive_timeout 65;

    server {
        listen 8480;
        server_name 160.40.48.102;

        location /hls {
            # Serve HLS fragments

            # CORS setup
            add_header 'Access-Control-Allow-Origin' '*' always;
            add_header 'Access-Control-Expose-Headers' 'Content-Length';

            # allow CORS preflight requests
            if ($request_method = 'OPTIONS') {
                add_header 'Access-Control-Allow-Origin' '*';
                add_header 'Access-Control-Max-Age' 1728000;
                add_header 'Content-Type' 'text/plain charset=UTF-8';
                add_header 'Content-Length' 0;
                return 204;
            }



            types {
                #application/dash+xml mpd;
                application/vnd.apple.mpegurl m3u8;
                #video/mp2t ts;
            }
            root /tmp;
            add_header Cache-Control no-cache;
        }
```


Figure 4-13: Sample of NGINX code

Furthermore, lighting was optimised to enhance the level of realism and simulate the actual interior's lighting, with the main goal to point out the artefacts. Extra lighting was also added where needed. Figure 4-14 shows the final scene inside Unity Editor.

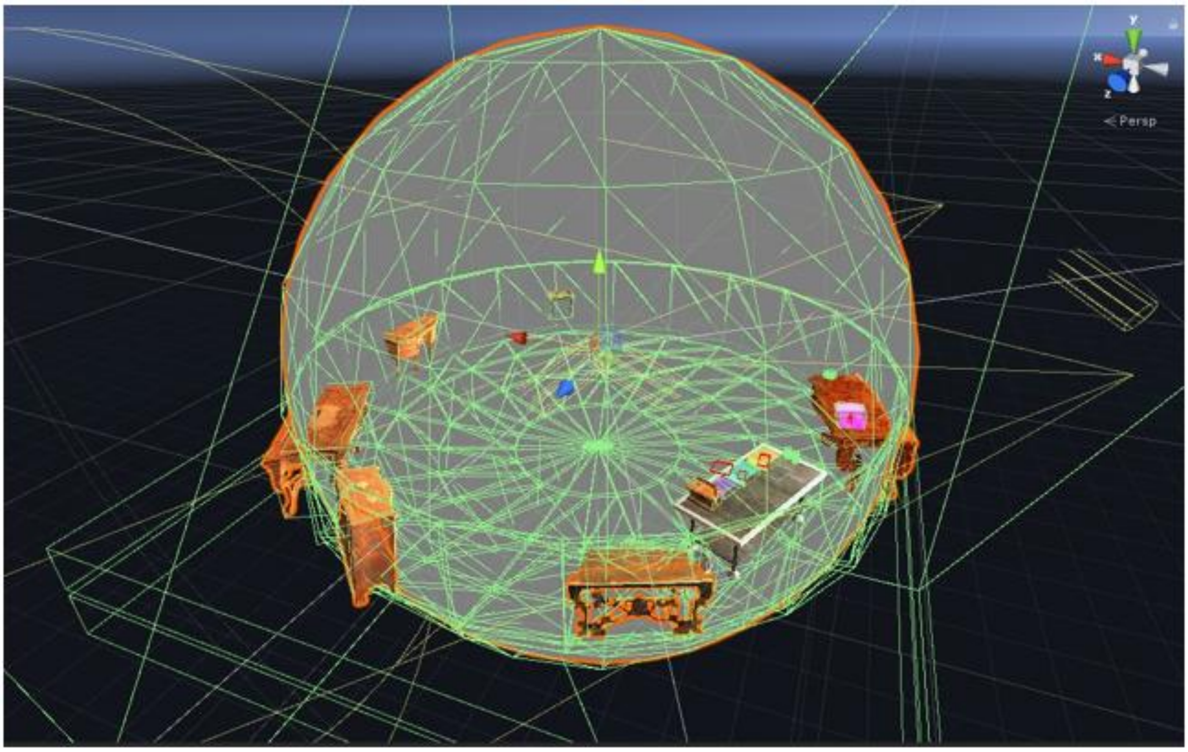


Figure 4-14: Scene inside Unity Editor

As we see, a flat area was added representing the floor in a height lower than the level of the sphere's centre. Additionally, limitations were added to the walkable area where avatars can move.

4.3.3.2 Avatars development

One of the main aims of the VR application is to offer enhanced realism, in order to give the feeling to the users that they actually are inside the museum and interacting with other people. A humanoid avatar with realistic movements is the first and most critical part in terms of latency for a user, especially for one who did not have another similar previous experience with VR applications. The user has to control a humanoid avatar which actually reproduce the exact same moves the user performs using the controllers. Otherwise, it will be confusing, disturbing and most probably will end as an unpleasant experience at least. For the Touristic City testbed, four avatars have been created to play two different roles: a touristic guide of the museum and a tourist, both with male/female options (Figure 4-15). Details on the avatar's development are reported in Annex 7.2.

An inherent problem of VR applications using avatars is their tailoring to fit each user. Human bodies come to plenty different sizes and shapes and a VR application must be flexible enough to provide the same experience to anyone. To do that a calibration process takes place, where the user is requested to keep a straight pose facing the sensors (in order to detect the eye position). Once done properly the synchronised movements add to a more complete realistic experience, otherwise the user would be viewing through the mesh of his avatar, thus obstructing his visual field.

Figure 4-15: Touristic City testbed VR application's characters

Another critical part is the interaction with objects. For this, colliders were added in artefacts not allowing the avatar to pass along them. For each object a specific hand pose was designed around a pivot, offering a suitable and realistic grab from all corners and height. Each bone of the fingers was placed in specific position and rotation angle, to provide a realistic experience (e.g., Figure 4-16 shows the bone structure of an avatar holding a pencil).

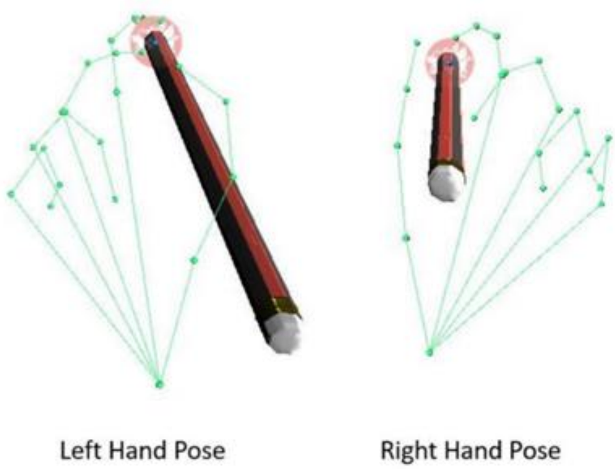


Figure 4-16: Hand poses over an object

4.3.3.3 Restoring tutorial implementation

To enhance the realism of the VR experience Palazzo Madama suggested a use-case scenario to take place inside the museum that meets very well the 5G features to enable low latency real-time interactions. As part of the VR application, the interaction between the tourist and the guide is based on a process where the two users collaborate to perform a restoration of a wooden object. The coordination between the two users essentially reside in two steps: First the guide passes physical tools to the visitor for each step of the restoration process; this means that in the 3D space their hands must be synchronised so that when the tool is passed on from one user to the other the tool is not dropped. Secondly, the tourist-user proceeds with the restoration process guided from the other user receiving instructions in real-time through VoIP. Based on such assumptions, a collaboration with the restorer in charge of taking care of the wooden parts and artefacts of Palazzo Madama has elaborated a restoring tutorial. An example of a restoring process for the reconstruction of the missing portion in a wooden object is reported in Annex 7.3.

The restoring example has been used as basis to elaborate a more appropriate tutorial on one of the wooden tables in the Madama Reale chamber, also taking into consideration the limitations that the interaction among users through the haptic application may have; it would be in fact difficult to implement a collaborative action over small pieces, such as those described in the tutorial. Furthermore, it would be even annoying for a user, that just wants to have a VR experience, to face a so much detailed and difficult approach.

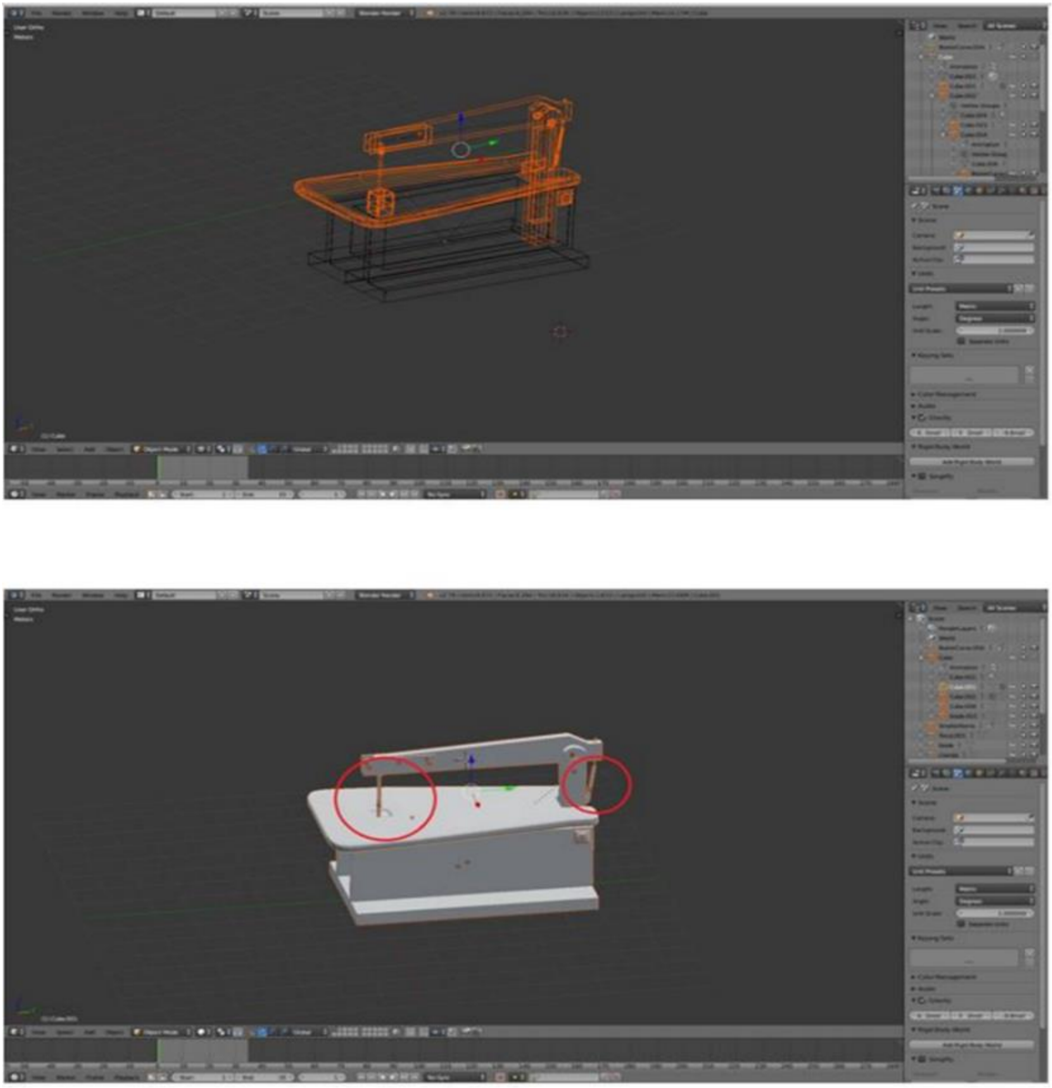


Figure 4-17: Sample of the procedure creating scroll saw with Blender

For the scenario purposes, Blender [Ble19] was used to make new 3D models from scratch. Blender is a 3D modelling, animation, motion graphics and rendering application. It is capable of procedural and polygonal/subdivision modelling, animating, lighting, texturing, rendering and other common features found in 3D modelling applications. A scroll saw, with automated animations made inside Blender, was designed to chop the wood. A sample of the procedure creating scroll saw with Blender is shown in Figure 4-17. Red circles inside the second picture shows where animations have been added. Other models like pencil and brush are shown in Figure 4-18.

Figure 4-18: 3D Models of restoring tools created for the scenario

Using the same technique some objects were created in order to be placed in a tutorial room for learning purposes: a ticket booth and an entrance with a ticket validator that can be seen in Figure 4-19. The user has to reach the ticket booth in order to print a ticket which has to validate in the validator. When the procedure will end, user will be able to transport to the main room.

Figure 4-19: Ticket booth and validator

In general, the objects were modelled in Blender, where the albedo[12] and the normal map were baked from a high poly version of the item to a low poly one, which is better suited for a VR application. After UV unwrapping the mesh, it was exported to Unity using the "fbx" format. There the materials

[12] The term "Albedo" refers the reflecting power of the material, so it shows the overall brightness of an object.

were created and assigned using the baked albedo, normal and detail maps and Unity's Physically Based Rendering (PBR)[13] shaders. Regarding the actual scenario inside the application, the tourist-user is guided about how to restore a missing part from one of the wooden tables in the Madama Reale chamber.

4.3.3.4 Interaction between users (haptic server)

The most critical part of the application is the interaction between the two users in the application (the touristic guide and the tourist). Developing a Virtual Reality application which supports multiplayer a lot of effort must be given in synchronisation between players. For example, when a player throws a ball to another player there must not have any delay either to the trajectory or to the hand movement. Otherwise, the ball will pass from second player. 5G network can provide high standards of stability, bandwidth and latency time and therefore is the most appropriate type of network for VR applications. As described in [5GM18-D51] a server was created from scratch based on protobufs [Pro19] and with a fully editable code; however, this approach had several limitations. The need for a physics engine to handle the server-side physics simulation led to the migration to Unity (and C# as programming language), ensuring in this way a better synchronised result, which is considered of great importance. Moreover, the underlying transport protocol was also changed to RUDP (UDP with a layer on top, simulating TCP sliding window and countering the unreliability of UDP on Application layer), which is more suitable for transmitting continuous time-critical data due to no Head-of-line blocking, and provides more flexibility for Wireless Local Area Networks (WLAN) and a greater scalability.

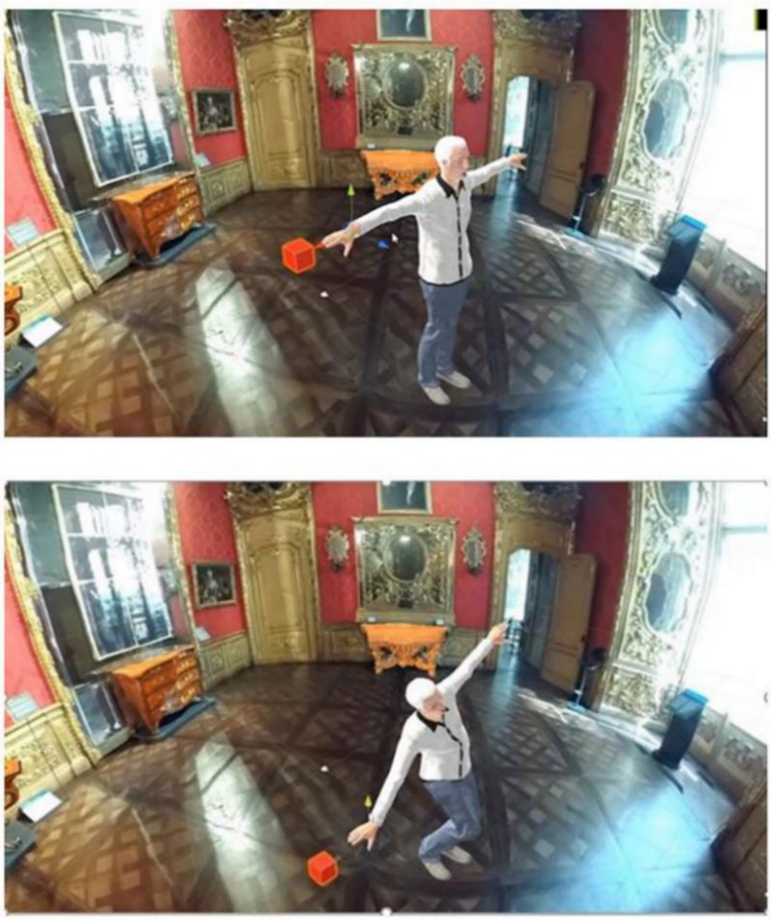


Figure 4-20: Inverse Kinematics

[13] PBR is a lighting model which simulates the natural interactions of light rays and real-world materials. Unity makes it easy to create dynamic materials with excellent contrast and detail, without using any maths.

The VoIP service was also developed inside Unity, i.e., without using an extra server. After short parameterisation two or more users are allowed to talk to each other through the Oculus Rift Microphone (or other equipment) once they are connected (the connection is established right after the tutorial).

Inverse Kinematics (IK)[14] makes use of the kinematic equations to guess the joints position in order to reach a specific point. In our case by moving a single point we can alter the pose of our character. Thus, by broadcasting only the tracked positions/rotations of each user and their central spine bone, we can recreate their movements to the other clients, without having to send the whole skeleton's joints and without depending on the different character rigs. More specific the information is sent has 4 end effectors includes both hands, head and position in space rather than 21 (whole skeleton). The average packet size in an established connection (not just sending a single packet) in a worst-case scenario, where the user is constantly moving at a big degree is 130-140 on the wire with 650BPS Bandwidth out. The package size and bandwidth are highly affected by the health of the connection as well as the alteration of the data between 2 messages. Figure 4-20 shows an example with an avatar trying to reach a red cube.

4.3.3.5 VR application user experience

The application has been designed with multi-lingual capabilities to adapt to the user preference. So, when the application starts, the user can choose between English or Italian. Once the language has been selected the calibration process takes place, asking the user to look forward to a light. Then the user spawns in a tutorial room as shown in Figure 4-21. Regarding the server's IP, they are passed in a configuration file, which is deployed with the application after the user is accepted by the haptic server.

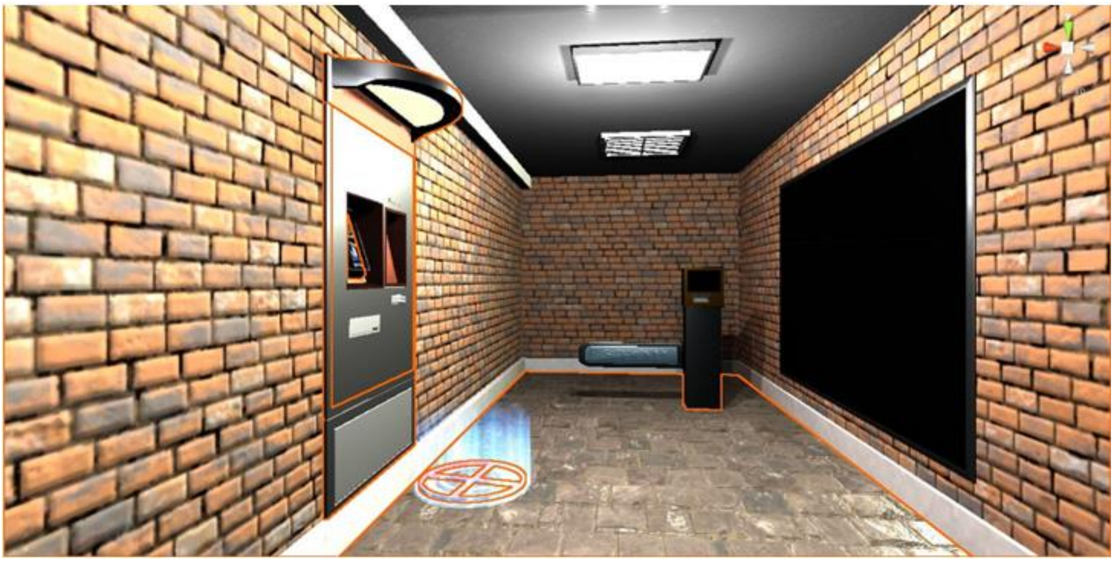


Figure 4-21: Tutorial room

Inside tutorial room the basic movements are described in the screen of a large tv in the middle of the room. Every step is described through an animated picture. The main aim for the user at this stage is to choose a character, print a ticket and proceed to the ticket validation. Figure 4-22 shows character options respectively, while Figure 4-23 shows the ticketing procedure.

[14] Forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters. The reverse process that computes the joint parameters that achieve a specified position of the end-effector is known as Inverse kinematics.

Figure 4-22: Role option

In order to avoid unwanted situations for the user (e.g., discomfort or even motion sickness), it is preferred the motion of the user to be done by simulating a teleporting process. For this, by moving the controller, a curved raycast hits the floor, so when the spotlight becomes green it means the user can be teleported to the specific spot by pressing a button. Figure 4-24 shows the raycast.

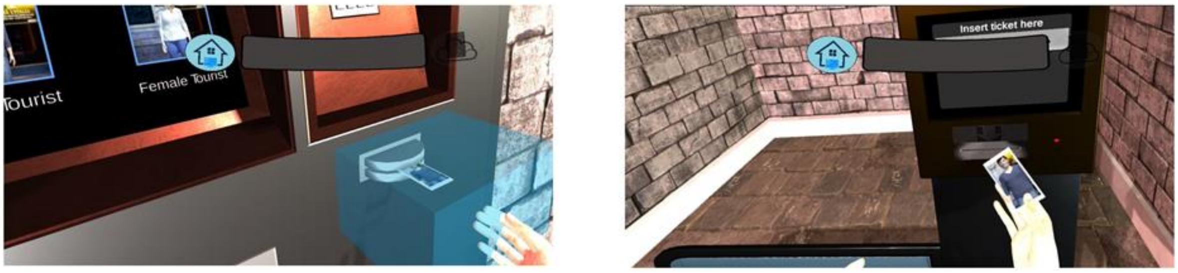


Figure 4-23: Ticketing procedure

The tutorial room provides also beacons which are used to guide the user across the different steps. One of those white beacons can be seen in front of the ticket booth also in Figure 4-24. A guide will already be in the tutorial room when a new user logs in; therefore, only the tourist characters will be available for the visitors to choose. The guide will help the user if he cannot successfully perform some of the tasks. When the ticketing process is completed both users are teleported to the main room.

Figure 4-24: Teleport Raycast

The artefacts of the museum are placed accordingly to match the video as close as possible, however once the projected video is live streamed, minor adjustments have to be made. The server can propagate these changes to the users using RPC to either change the rotation of the objects by a small incremental part, or to set the rotation completely. A configuration file is used as source for these values. Though the artefacts are not shown to the user when first enters the room; they are shown as a reward when the user completes the restoration task. Regarding the procedure, first the guide demonstrates the damage that has being dealt to the artefact and gives some general instructions for the restoration procedure. Then some indicators, used as beacon, are enabled to guide the positions and the tools the user must take to complete the restoration, and a window with animated texture, like a gif, will show the actual movement the user has to do.

For the restoration, the user will need a scrapper from a tool table to cut the damaged wood piece. Once removed, the tourist is asked to cut the appropriate shape from a wooden piece, using a scroll saw located on a workbench. The path of the cut is drawn on the surface of the wooden piece the same way it was drawn on the inlay removal. Moreover, the displacement of the wood due to the cut is being drawn using a tessellation displacement surface shader, which also handles the colouring of the cut depending on how successful it was. Finally, the resulting wooden piece must be inlayed, and taken care of with pad and shellac, then it has to be glued on the table to complete the restoration process. The tourist is requested to complete these steps using a brush and alternating normal, detail and albedo maps on the furniture. A more detailed description can be found in Annex 7.3 and in the video presentation[15]. The different stages of the restoration over the table can be shown in Figure 4-25. In any case each simple step will be shown in the animated texture mentioned before, achieving a high level of simplicity in order to provide a pleasant and easy experience to the user. Figure 4-26 shows two users inside the VR environment.

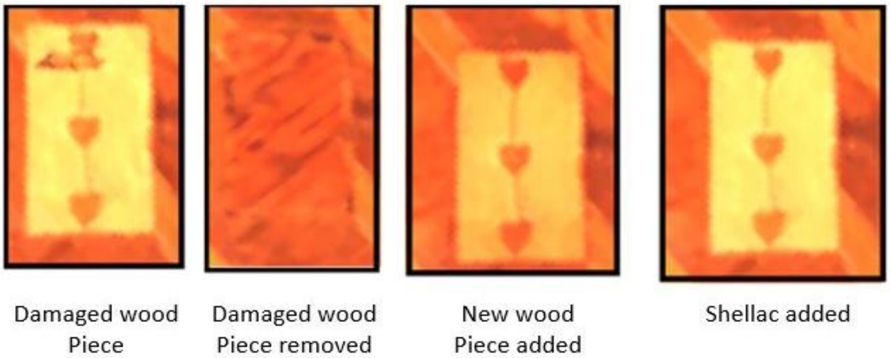


Figure 4-25: Stages of the restoration

Figure 4-26: Avatars in the VR application

[15] https://www.youtube.com/watch?v=hLCkgdOhVJ4&feature=youtu.be

A migration process was also implemented, in order for the users to change server as seamlessly as possible. The migration process consists the following steps which are transmitted using restful services:

- First the currently active server is prompted by the orchestrator, to pass its context, containing all the information about its current state, the server also provides to the users the new IP address and prohibits any new connections being made.
- Then, the context is provided to the new location (in the case of this scenario, the edge cloud) where the same scene is setup. Once the edge server is ready, it responds to the orchestrator that it is ready to proceed to the actual migration and is forwarded to the main server.
- The server pushes the notification to the user to display the migration. A bar indicator is shown in front of the user which inform when the migration of the network functions to the edge cloud is happening and in which status the network is. During this process all disconnect events are disabled to provide a seamless feeling. Once the users are connected to the new server the process is completed.

Figure 4-27 depicts the procedure through a sequence diagram.

4.4 Testbed integration, setup and deployment

In this section, we describe the integration of the different components of the testbeds, whose implementation has been described in the previous section. We also describe the setup and deployment of the integrated solution in the premises of the Touristic City testbed, the Palazzo Madama of Turin.

The integration activity of the Touristic City testbed followed a step-by-step approach. The different modules were implemented stand-alone and subsequently integrated based on the HW/SW architecture defined in [5GM18-D51]. The integration activity was done through the organisation of ad-hoc workshops at the partners' premises to perform the various tests and evaluate the preliminary performance. Afterwards, a complete testbed setup was deployed in Palazzo Madama.

4.4.1 Integration of PHY/MAC with higher-layer PS

The integration of PHY/MAC with the higher-layer protocol stack (PS) took place at the MOBICS premises. Before integrating the PS, a pre-test has been performed by establishing a standalone radio connection between gNB and just one UE. Both slices (eMBB and URLLC) have been tested on air in an indoor environment.

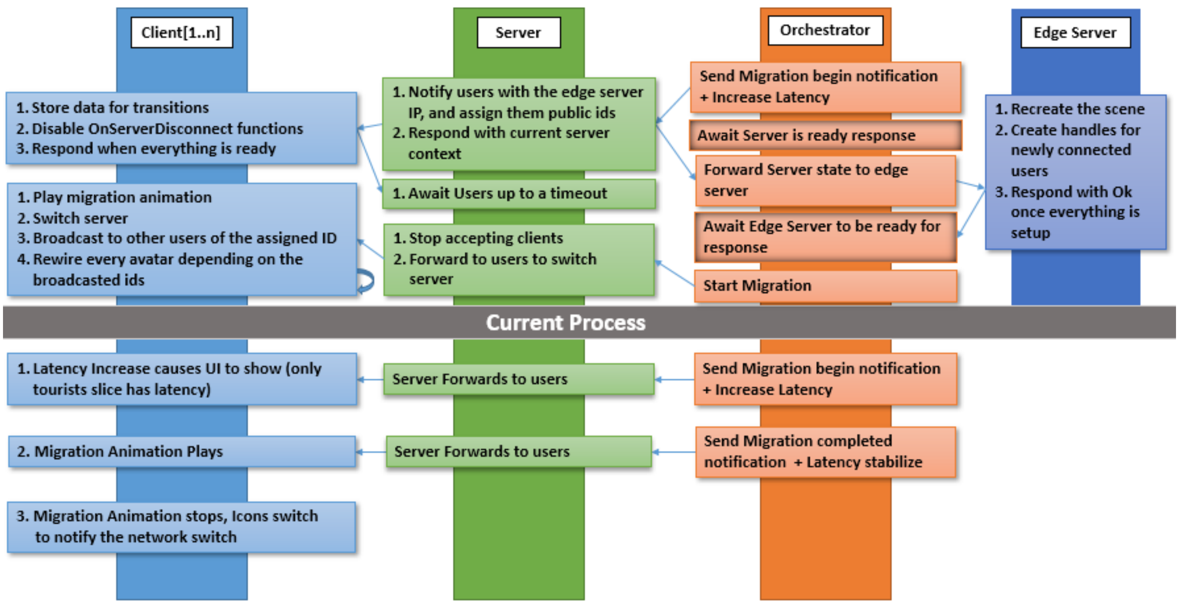


Figure 4-27: Migration sequence diagram

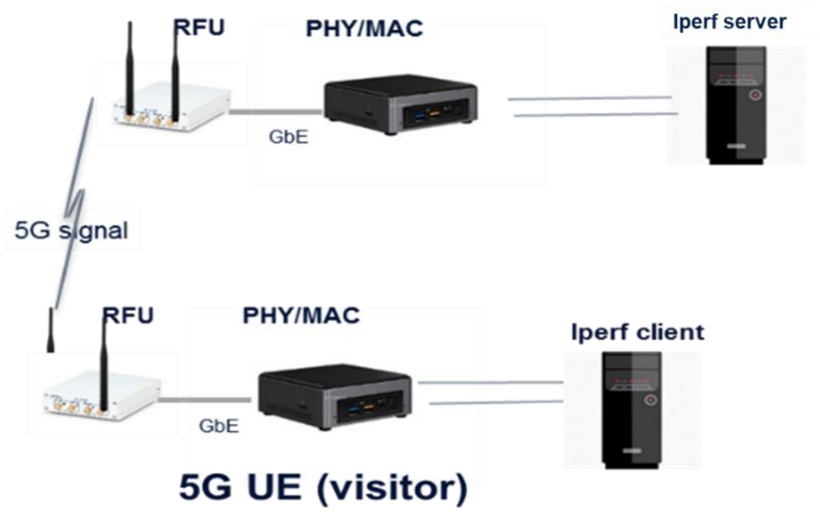


Figure 4-28: Integration and testing of PHY/MAC

As shown in Figure 4-28, two application PCs have been connected to the BBUs at both sides via an Ethernet connection. No routing functionality was needed for the PHY/MAC interface to the application, since the MAC interface was made in a way to accept any traffic coming into the physical port.

Figure 4-29 shows the setup of the integrating higher layers with PHY/MAC. A PC running the PS of the gNB was connected via Ethernet to gNB BBU; also, another PC running the PS of the UE was connected to UE BBU.

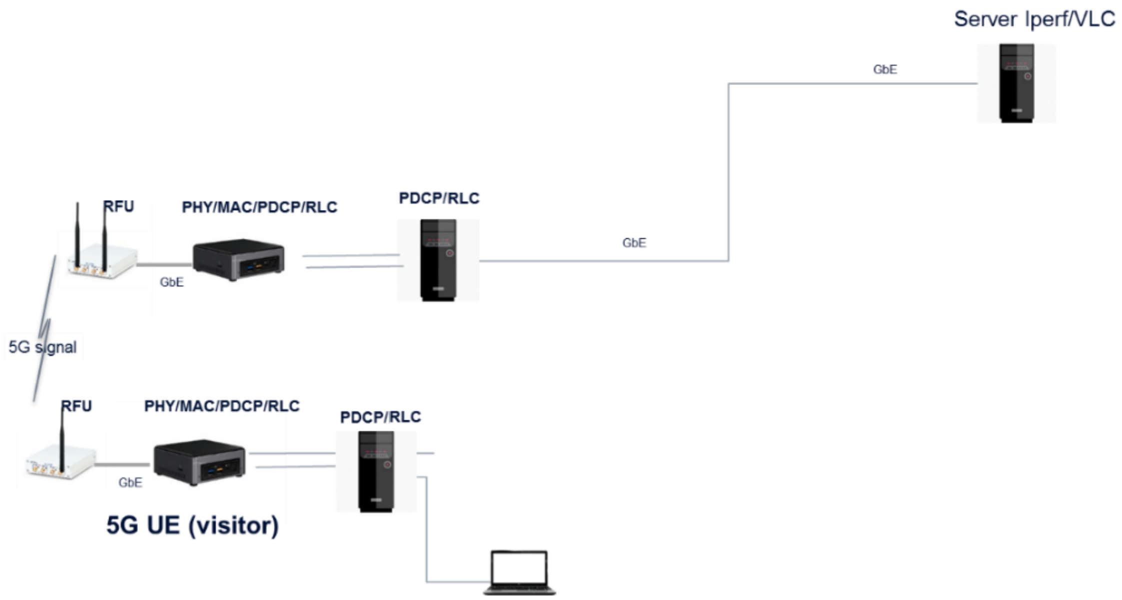


Figure 4-29: Integration of PHY/MAC and higher-layer PS

4.4.2 Integration of PHY/MAC/PS with VR application and tests

The integration of PHY/MAC/PS with VR application took place at MOBICS premises. The test setup is illustrated in Figure 4-30. HW and SW equipment that realises 5G BBU/RF was provided with two different configurations, one per slice (eMBB and URLLC). An advanced antenna has been added to the gNB part of the BBU/RF. The 360° camera stream was fed in the NGINX Server (depicted in Figure 4-30 as the 360° video server) connected to PS PC. At the UE side, 3D glasses with a VR app client was connected to the UE PS using an Ethernet connection.

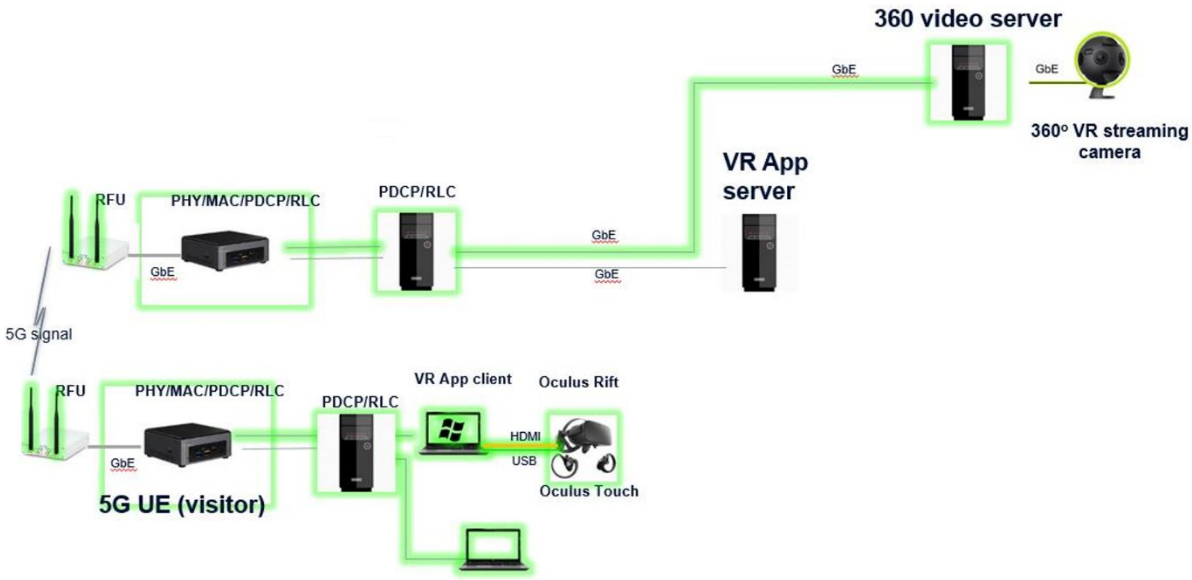


Figure 4-30: Picture of one multi-connectivity device used in the testbed

For each target slice (the eMBB and the URLLC one), a related application was developed. Focusing on the eMBB slice app, the output of the 360° camera is fed in NGINX Server. The NGINX server provides the VR app client with video content using the HLS and the Real Time Streaming Protocol (RTSP) protocols. Considering the need to showcase the gain of elasticity in service provisioning, the NGINX server simulates additional users that enter the virtual room, i.e., additional traffic that flows in the network, as a way to gradually degrade the performance and trigger the relocation of the server (enabling of elasticity mechanism). Furthermore, a basic implementation of PDCP/RLC functionality in virtual machines separated from the BBU/RF equipment was developed and used as part of the functions that can be relocated in the context of elasticity enabling. The test of the eMBB Slice showed a good quality for both a 2-Dimensions (2D) screen and from QoE perspective with oculus. While the eMBB slice includes video data that set a virtual room, the URLLC application was developed to realise the exchange of information coming from user haptics and VoIP communication between the objects in the virtual room.

4.4.3 Testbed integration in Palazzo Madama

4.4.3.1 Setup scheme and deployment

The 23-24th of October 2018, a first Touristic City testbed setup took place in Palazzo Madama. The scope of this activity was essentially to test the equipment setup and their complete integration in the location in which the testbed has been opened to the public. This first setup didn't include any orchestration function to focus on the troubleshooting of the basic configuration; the main scope of the activity was then to check the correct communication between the two users in the VR environment taking advantage of the slice's implementation for the 360° video stream, and the interaction through the Oculus Rift sets. For this first setup it was agreed with Palazzo Madama to proceed with the activity during the closure day of the museum, to have the complete availability of the spaces and no time restrictions.

A preliminary testbed setup scheme was drafted in advance, based on the planimetries provided by Palazzo Madama. The testbed was foreseen to be deployed in three rooms: the Madama Reale chamber, the Small Wardrobe and the Bookshop (see Figure 4-2 in Section 4.3).

More in detail, a 360° video camera would have been placed in the middle of the Madama Reale chamber (Figure 4-31) in order to provide a real time high quality video stream used to build the VR environment in which the tourist and the guide would interact (more details on the VR application can be found in Section 4.3.3). The Small Wardrobe was decided to be the location of the tourist guide setup, consisting of a PC running the VR application client connected to an Oculus Rift set; a switch would have provided the connectivity between the 360° video camera, the VR application client and the rest of the equipment of the testbed placed downstairs in front of the Bookshop. The connection between the equipment's on the two floors would have been provided by an optical fibre due to the lack of a reliable Ethernet connection. The space in front of the bookshop would have been used to setup the core of the testbed consisting in three main parts: 360° video server and VR application server, the gNB and the UE, where the location of the tourist would have been setup by a second VR application client and the related Oculus Rift set.

Based on the set-up described above, the testbed setup took place the morning of the 23rd October 2018 by placing all the needed devices in their corresponding location. With respect to the initial schemes, the final setup has been slightly modified to accommodate in the same place (i.e. the Small Wardrobe) the 360° video server and VR application server for practical reasons. This modification in the setup didn't impact on the testbed functionalities since the connection between all the devices was performed through two switches and mapping the connections over pre-defined IP addresses.

The final setup has not differed too much from the one that was tested in Palazzo Madama on that date. Additional equipment has been used for introducing the orchestration function as well as the central and edge cloud parts of the architecture (see Section 4.3.2 for the details). Such equipment has been placed at the Bookshop level, where the radio part of the testbed has been placed. Additional work has also been required to respect the security measures that have been requested when the testbed was open to the public (for instance, all cables were covered, and all the supports such as tables, chairs, etc. had to be provided made of fireproof materials). Annex 0 includes some pictures taken during the workshop.

Figure 4-31: 360° video camera in Madama Reale chamber

4.4.3.2 Radio deployment details

Focusing on the radio part, the testbed provides an indoor coverage. Since Palazzo Madama is located in Turin downtown other services operating in that area must be protected from co-channel or adjacent channel interferences. On this issue, the Italian Regulatory Authority provided a temporary license on the 3.4-3.6 GHz band (B42) for experimental use. In particular, it has been was possible to operate on the band portions allocated to the military services in FDD mode from 3.415 GHz to 3.437 GHz for the UL, and from 3.500 GHz to 3.537 GHz for the DL. Since the testbed was requiring a total band of 25 MHz in Time Division Duplex (TDD) mode consisting of 15 MHz for the eMBB slice and 10 MHz for the URLLC slice, it was not possible to allocate the entire band in the lower portion of the B42 (3.415-3.437 GHz); therefore, the radio part was configured to separately transmit the eMBB slice over the lower portion of B42 and the URLLC slice over the higher portion of it, thus guaranteeing also good frequency margins to avoid interference on the adjacent channels. Some measurements were anyway performed outside the building to check any co-channel interference; the low power transmission of the gNB and the attenuation of the building walls allowed to guarantee that the military service was not subject to interference outside the testbed area.

4.5 Experimental evaluation

In this Section, we describe the experimental evaluation based on the KPIs measurements in the Touristic City testbed. In [5GM18-D62] we described the relevant KPIs for the testbed, along with the methodology used to evaluate them in a broader context. In this section we evaluate the performance of this KPIs.

In addition to evaluating the different performance KPIs (at the network and application layers), the Touristic City testbed has also been very useful to get feedback from the users, as well as the vertical player involved in the trial, about their satisfaction with the use case enabled by the testbed and the corresponding application. Indeed, this has provided very valuable information about the value that the applications enabled by 5G-MoNArch technology has for the corresponding customers and end-users. This part of the evaluation has been reported in [5GM19-D63], along with the stakeholder analysis and validation contained in that deliverable, while the focus here is on the performance KPIs.

This section is organised in two main areas: Network KPIs (Section 4.5.1) and Application KPIs (Section 4.5.2). Among the Network KPIs analysed in the context of the touristic city testbed, we mostly focus on four categories that are important both for the overall user experience and to showcase the network elasticity algorithms developed in the project. Specifically, we focus on the following KPI categories:

- The *pure RAN performance* in terms of bandwidth and achieved minimum latency, in order to guarantee that the two slices included in the testbed have the needed QoS parameters.
- The *orchestration related aspects*, driven by (i) latency, which is a needed feature for the testbed to guarantee the very low latency requirements for the haptic communications, and (ii) elasticity, which is one of the outcomes of the elastic network management discussed in [5GM19-D42].
- The *service creation time*, showing that 5G-MoNArch technology can meet (and even substantially improve) the requirements set by fora such as 5GPPP on this metrics.
- The *network data analytics processing*, which is a fundamental enabler for the KPIs corresponding to the above three categories.

In addition to the Network KPIs, in this section we also report the performance in terms of Application KPIs, which are the final metrics perceivable by user through the VR applications. These non-network related KPIs serve as a validation for the underlying network algorithms developed by 5G-MoNArch.

4.5.1 Network KPIs

In this Section, we evaluate the pure network KPIs measured in the testbed. We remark that these measurements are collected without real users in the testbed, but in dry runs while validating the performance of the network. As discussed in Section 4.3.2.1 the main metrics that are improved by the elastic network orchestration algorithms and the 5G-MoNArch overall architecture are two: (i) the reduced latency obtained by re-locating VNFs (the UPF in this case) from the central cloud to the edge cloud, described in Sections 4.5.1.3 and 4.5.1.4, and (ii) the reduced service creation time obtained using the 5G-MoNArch slice blueprint onboarding as described in Section 4.5.1.5. Moreover, in this section we also discuss the pure radio performance (Section 4.5.1.2) and functionality/decisions of the data analytics framework (i.e., the Measurement tool described in Section 4.5.1.1) which impact these metrics.

4.5.1.1 Measurement tool

The main KPIs for the Touristic City testbed are the latency, the CPU load and the memory (RAM) consumption. Before providing the KPIs measured for this testbed, in this section we describe the Measurement tool that we have employed to perform such measurements, which thereby represents a central component of the data analytics framework: all the data exchanged across VNF and orchestration with the relevant KPIs (delay, bandwidth, CPU and memory consumption) are eventually shown through a GUI, and the Measurement tool furthermore serves as a data analytics database (DB) which is used by the Orchestrator (see Section 4.3.2.3) to fetch the relevant information required to perform actions such as relocation and scaling.

The KPIs mentioned above are measured at user plane RAN functions, since these functions directly affect the processing of users' data and, thus, the users' experience. To this end, a tool that implements the task of measuring, storing and visualising the data has been implemented. The measurements are depicted in a GUI, with a front-end web application that, in addition to visualising the metrics, is used for the user interaction with the testbed. As depicted in Figure 4-32, the percentage of the CPU and RAM load is depicted in charts, with the option to depict the total consumption as well as the consumption per slice or per domain (edge/central node). The latency is illustrated in a horizontal bar, including a threshold value that can be manually set.

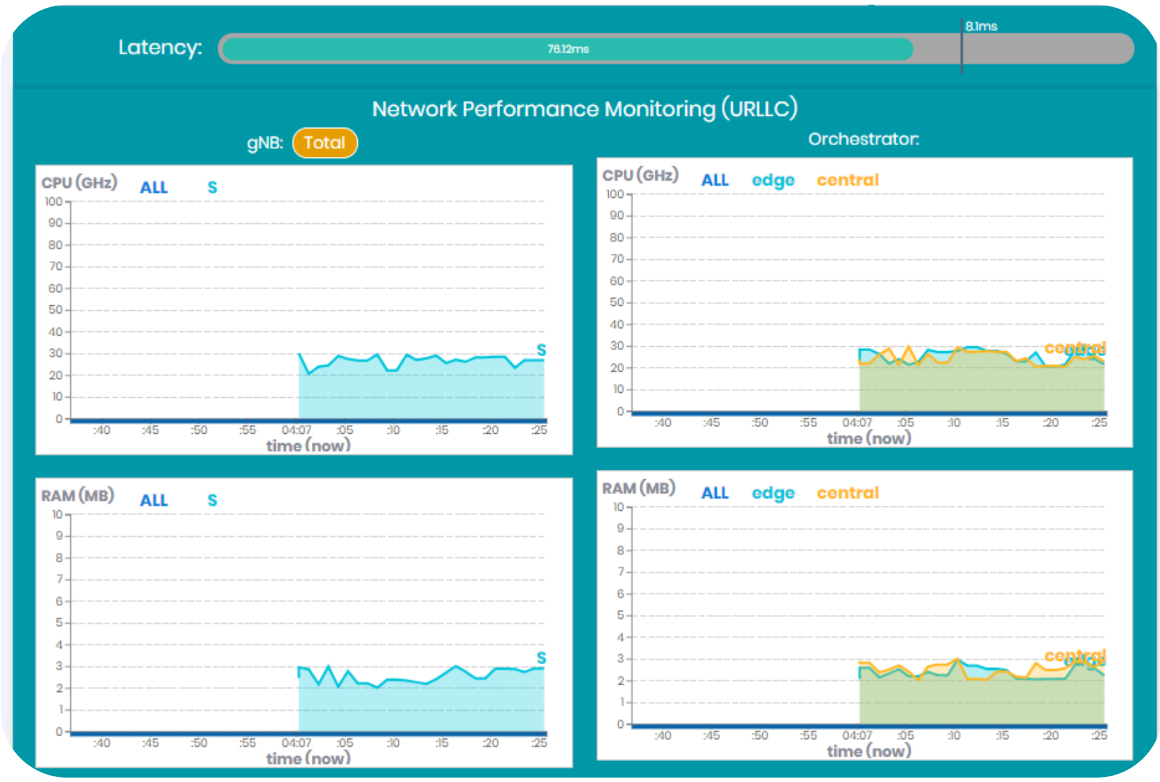


Figure 4-32: GUI with the KPIs of the Touristic City testbed (latency and CPU/RAM usage)

For retrieving "live" metrics, from the gNB node and the node (virtual machine) that hosted the 5G layer 2 stack of the testbed, a python script has been developed. It encodes the data in json format and then performs http requests towards a back-end service of a mongo database where the "live" data are stored. This procedure is performed every second and accordingly, the back-end service of the web application of the GUI polls the database at the same time interval and fetches the metrics to the front-end which then are presented in the form of plots. Considering the target KPIs, the HTTP request contains, as mentioned above, a JSON encoded message, which formed by three fields, namely, the node's id (e.g. gNB node, central cloud upper layer node, etc.), the CPU percentage and the RAM percentage.

4.5.1.2 PHY and higher layers KPIs

The novel service and use case described in Section 4.2 relies on two network slices: a URLLC slice and an eMBB one. These slices need satisfy the requirements corresponding to the types of traffic in the radio access network (RAN). In this section, we thus evaluate the RAN KPIs' results of the testbed to show how these requirements are met.

The test has been performed in multiple phases. The first test performed was an Iperf stand-alone test for PHY/MAC for both eMBB and URLLC slices. This test has been done as part of the integration work, where the radio capabilities of both slices (eMBB/URLLC) were tested with traffic artificially generated. Slice capabilities (maximum throughput and latency) were measured by iperf/ping for both slices. For the BLER performance, the channel conditions were perfect (with LOS communication).

The test showed a throughput of 29Mbps for the eMBB using UDP streams, and 5.8Mbps for URLLC using the Transmission Control Protocol (TCP). The latency test showed a value of 2.7 ms Round Trip Time (RTT) for URLLC, and around 4ms for eMBB. No packet loss was detected in any of the slices. Also, an offline video has been streamed between the two PCs at 17Mbps. The test main parameters are summarised in Table 4-2. We conclude that URLLC requirements in terms of delay are met. In terms of throughput, the values attained are sufficient for the needs of this use case, while higher throughputs could be achieved by using more bandwidth.

Table 4-2: Test parameters

Test parameters	**eMBB**	**URLLC**
TX Power	~1 dBm	~1dBm
gNB-UE Distance	2 m	2 m
Modulation scheme	16 QAM	QPSK
Code Rate	0.86	0.67
TTI length	1 ms	0.5 ms[16]

For testing the integrated higher layers with PHY/MAC, a PC running the PS of the gNB was connected via Ethernet to gNB BBU; also, another PC running the PS of the UE was connected to UE BBU. The test consisted of the following steps:

- The first step was a successful ping test between the gNB and UE PSs (this is just PHY/MAC RF+BB standalone gNB to UE test with no application connected and no protocol stack; the ping and iperf is done from gNB BB machine to the UE BB machine).
- The second step was an Iperf test showing similar throughput performance as for the BBU to BBU test, although with higher latency due to the independent machines of PHY/MAC and higher layers and the Ethernet interface (this is the PHY/MAC connected to the application PCs running a video stream with no protocol stack PS).
- Finally, the third step consisted in the PHY/MAC integrated with the protocol stack.

Figure 4-29 shows the test setup. Table 4-3 summarises the test results of the above-mentioned tests. Finally, Table 4-4 shows the performed tests and the observations of the integrated RAN of the testbed with the VR application.

[16] Please note that no HARQ was used in the radio (ARQ was not under test in the higher layers).

Table 4-3: Test results of the integrated PHY/MAC/PS

Step	Test scenario	Latency (ping RTT)		Throughput		Packet loss	
		eMBB	URLLC	eMBB	URLLC	eMBB	URLLC
1	BB/RF to RF/BB	4.8ms	2.7ms	Not tested	Not tested	0%	0%
2	App./BB/RF to RF/BB/App.	5ms	2.8ms	UDP:29Mbps TCP:13Mbps	5Mbps	0%	0%
3	PS/BB/RF to RF/BB/PS	12ms	Not tested	UDP:29Mbps TCP:13Mbps	Not tested	0%	Not tested

Table 4-4: Test observations and outcome of the integrated PHY/MAC/PS and VR

Test	Configuration description	Observations - Outcome
#1	eMBB slice – data flow from the camera to the client • Live capture from 360° camera • Transcoding via VLC or ffmpeg (HLS over HTTP / TCP) • gNB Higher layers • (bypass Air interface – no RF part, ethernet able is used) • UE Higher Layers • Simple VLC client or Oculus Rift	• Very good quality, ffmpeg is used as transcoder instead of VLC. Decision to continue the tests by feeding the output of the NGINX Server to an ffmpeg transcoder. Optimisations regarding the app's configuration w.r.t. to the peculiarities of the other components are available either through the camera SW or with ffmpeg parametrisation. • The user experience using the Head-mounted Display (HMD) is more attractive compared to the projection in a 2D screen. However, the video quality seems to be lower in the HMD due to the capabilities of the HMD equipment itself – screen analysis etc.
#2	eMBB slice – data flow from the server to the client • Stored video depicting Palazzo Madama (1min) • Transcoding via ffmpeg (HLS over HTTP / TCP) • gNB Higher layers • (bypass Air interface – no RF part, ethernet cable is used) • UE Higher Layers • Simple VLC client or Oculus Rift	• The results are similar to those extracted in the case where live content from the camera is streamed. • Small issues in the use of ffmpeg (e.g., looping) are for further study and do not affect the real implementation in the Palazzo Madama
#3	eMBB slice – data flow from the camera to the client • Live capture from 360° camera • Transcoding via ffmpeg (HLS over HTTP / TCP) • gNB Higher layers • 5G air interface on • UE Higher Layers • Simple VLC client or Oculus Rift	• Good quality (requirement: to limit the server data rate to the value that the RF part can support)
#4	URLLC slice – app only test • Server and client SW installed and configured • Two virtual clients	• The app behaviour for the basic functionality (no avatars and haptics at the presentation level of the app) was validated

	Data flow • Server CERTH device – Ethernet - Client device MBCS	successfully. The app is to be finalised – work in progress with Unity. • Logs for throughput and latency are extracted. The interface with GUI is for further study
#5	URLLC slice • Server and client SW not installed and configured Data flow • Not checked	• The install of the URLLC Server in a new PC (at MBCS) takes a lot of time (there are dependences, versions, libraries to be checked etc.). Due to this, the URLLC over the higher layers and the RF is postponed for the week 27-31/8/2018 (remote access at MBCS PC has been provided to CERTH). The issue should be considered for the establishment in the PC at the Palazzo Madama (Internet access is required during the URLLC server set up).

Due to BBU/RF set up the UDP-based streaming is preferable if the target is to show high bit rates (bidirectional communication, i.e., TCP-like, according to the BBU/RF tests achieves maximum 13 Mbps, while unidirectional communication, i.e., UDP like, according to the BBU/RF tests achieves maximum 29 Mbps). However, 4K/8K video streaming over RF with UDP provides bad user experience, since it requires dynamic adaptation (which is provided only over TCP (i.e., over a bidirectional link).

The TCP based app development is selected. The option of applying data rate limits directly to the server has been investigated. Therefore, more optimisation of the radio interface was needed. To increase the DL throughput, an asymmetric configuration of UL/DL slot length has been applied to achieve ~20Mbps DL and 2 Mbps UL for control feedback.

Higher layers consume negligible CPU resources. However, when a MAC-RLC split is applied, the interfacing required for the MAC-RLC communication (it is ethernet-based) burdens highly the CPU consumption. To reduce the latency of the PS, functions such as RLC/PDCP have been embedded in the BBU equipment (notice that these tests are just for integrating PHY/MAC with the higher layers, no orchestration or virtual functions movements are considered).

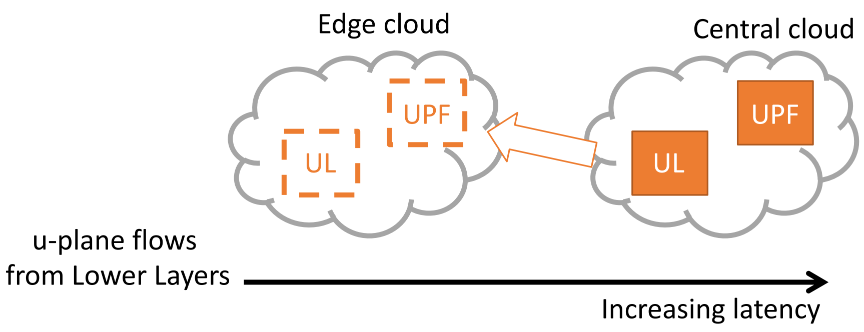


Figure 4-33: The VNF relocation scenario (c-plane functions are omitted for the sake of clarity)

4.5.1.3 Reduced latency KPIs

One of the fundamental functionalities needed by the touristic city testbed is to guarantee low latency for the user attached to the URLLC slice. Otherwise, the haptic interactions between the tourist and the guide may rapidly degrade to unacceptable levels, harming the overall VR experience. We have thus evaluated the improvement in the latency KPI on the URLLC network slice.

In particular, we have evaluated latency performance in the scenario depicted in Figure 4-33, where we have emulated the different latency between the two clouds by inserting a delay on the TN with the Linux traffic shaper command. In this scenario, we leverage the orchestration functionality of 5G-

MoNArch to place those functions which have an impact on latency, in the edge cloud (i.e., close to the end-user), with the goal of meeting the user's stringent latency requirements. In this context, we measured the E2E delay between the UE and the UPF, when the latter is re-located from the central cloud to the edge cloud after a trigger from the re-orchestration algorithm, as explained in Section 4.3.2.3. The result, averaged over 50 repetitions, is depicted in Figure 4-34. The results show how the algorithm described in Section 4.3.2.3 (triggered 60s after the beginning of the test) can achieve the goal of restoring the latency to the required level (indicated here at < 20ms, but it can be tailored according to the service). Also, the VNF here is relocated across the two NFVI clouds that compose the Touristic City testbed, although the orchestration setup can be changed adapting to the underlying infrastructure.

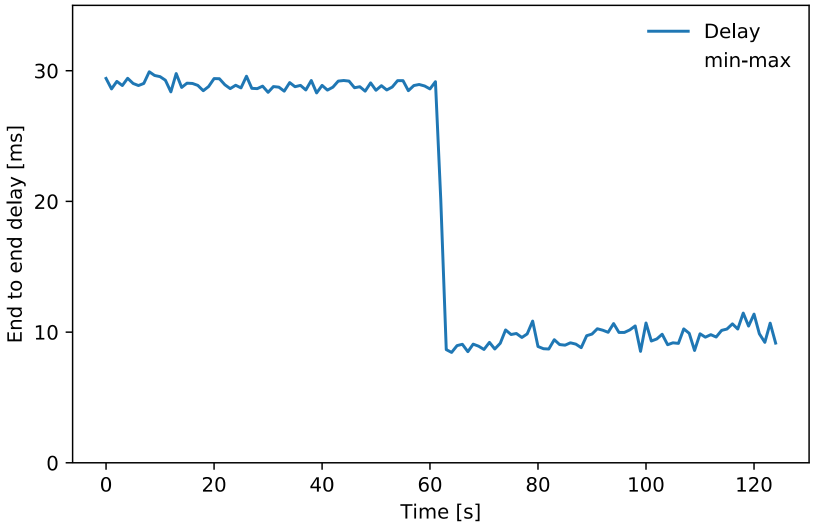


Figure 4-34: The overall E2E delay (UE-UPF)

4.5.1.4 Relocation delay KPIs

One of the key functionalities addressed by 5G-MoNArch is the one of orchestration, which allows to relocate VNFs from one node to another depending on the slices' requirements and the availability of resources in the various nodes. This functionality has been implemented in the testbed (see the algorithm described in Section 4.3.2.3). In the context of this functionality, a very important KPI is the delay in relocation VNFs. Indeed, it is of upmost important to be able to perform such relocation without causing any disruption to ongoing services.

To evaluate the performance of 5G-MoNArch from that viewpoint, we measure here the relocation delay of a VNF. Results are included in Table 4-5, and are obtained with an UPF VM running on an Ubuntu 18.04 VM with 5GB of hard disk and 2 GB of RAM. The different results are given for (i) four variants of our algorithm corresponding different status of the target machine, and (ii) OpenStack (OS), which corresponds to current state-of-the-art technology and thus provides a benchmark.

As discussed in Section 4.3.2.3, our algorithm has different variants depending on the flavour of the relocation algorithm as well as the availability of a cached VM image. Each variant provides a different performance, as depending on the various aspects the algorithm can take more time to instantiate the vanilla VM in the new location. In particular, our algorithm has the following four variants: i) already running (run in Table 4-5), where the engine is already bootstrapped, ii) pool, where the engine is already created in the new location, but not started; iii) non cached (Non-C.), where the image of the engine is available at the new destination but has to be created and bootstrapped for the first time; and (iv) cached, where the target engine has already been started in the destination machine in the past.

From the results, we observe that the performance obtained by OpenStack (OS in Table 4-5) is several orders of magnitude above our solutions, as moving functions across nodes located in different datacentres requires a full copy of the engine, including memory and disks. In contrast, delays are much smaller with our c/e-based solution (all of them but the OS column in Table 4-5), as this solution allows for a more dynamic setup and does not require any shared element across nodes, nor costly copy of the storage. Note that the timings with our solution (the c/e split) could be further reduced by employing a more lightweight engine such as Containers or Unikernels. We also recall that such values are the ones obtained after the re-location of the VM in the new position and the subsequent context update. The service is never disrupted, as the actual UP flow are moved just after the new VNF is fully re-located.

Table 4-5: Relocation delay

VNF	Variant				OS
	Run	Pool	Cached	Non-Cached	
UPF	70ms	28.3s	71.2s	149.3s	74 m 30s

4.5.1.5 Service creation time KPI and network slice

The service creation time is one of the most relevant KPIs in 5G Networks. The transition from a monolithic and rather static network architecture to a flexible one composed by different NFs, orchestrated into a heterogeneous environment composed by a hierarchy of cloud infrastructure needs means to automatically instantiate them. As described in Section 4.3.2.3, this is achieved through a MANO SW implemented by extending the OSM platform. However, the instantiation of such services requires a certain time to deploy the corresponding NFs and allocate the required resources, and it is very important that this time is not too long. In this context, the service creation time is a very important KPI for 5G.

In the following, we list the timing KPIs achieved within the testbed, organised into the 4 main areas of the Life Cycle Management. Timings are obtained using an ad-hoc SW written in Python. Measurement have been done several times (at least 30 repetitions) with very small confidence intervals (below few seconds) that are omitted here. Timings are summarised into the main categories that have been listed as relevant by the 5G-PPP KPI WG, to which the project is heavily contributing in this respect. The total onboarding time (7 minutes) is actually the sum of the two individual contributions on the NSD and the VNFD.

Table 4-6: Timing KPIs in the Touristic City testbed

What	Measurement Conditions	Time
Phase 1: Preparation		
Network Slice Template (NEST)	T_{start} = Request for onboard received in the system T_{end} = NEST fully onboarded in all the catalogues and available for LCM actions	7 minutes
Network Service Descriptor (NSD)	T_{start} = Request for onboard received in the system T_{end} = NSD fully onboarded in all the catalogues and available for LCM actions	2 minutes
VNF package (VNFD)	T_{start} = Request for onboard received in the system T_{end} = VND fully onboarded in all the catalogues and available for LCM actions	5 minutes
Phase 2: Instantiation, configuration and activation		
Instantiate Network Slice (NSI)	T_{start} = Request for instantiate received in the system T_{end} = NSI fully instantiated, ‘alive' and functional and available for monitoring actions	30 seconds
Instantiate & Activate Network Service (NS)	T_{start} = Request for instantiate received in the system T_{end} = NS fully instantiated, ‘alive' and functional and service performance (QoS) metrics meeting or exceeding the target performance (Ps) of the Network Service. Measure the service performance (QoS) metrics periodically (recommended once every 100 ms)	45 seconds

Instantiate & Configure VNFs in service chain (VNF)	T_{start} = Request for instantiate logged in the system T_{end} = VNF fully instantiated, configuration complete, VNF 'alive' and functional	10 seconds
Configure other NFVI elements	T_{start} = Request for any other NFVI related configuration logged in the system (e.g. HW acceleration, fast-paths, private networks, gateways, etc.) T_{end} = NFVI configuration complete, 'alive' and functional	10 seconds
Configure SDN infrastructure	T_{start} = Request for any SDN related configuration logged in the system T_{end} = SDN configuration complete, 'alive' and functional	5 seconds
Phase 3: Run-time modification		
Modify Network Slice configuration	T_{start} = Request for modify received in the system T_{end} = NSI configuration modified and functional	2 seconds
Modify Network Service configuration	T_{start} = Request for modify received in the system T_{end} = NS configuration modified and functional and service performance (QoS) metrics meeting or exceeding the target performance (Ps) of the Network Service. Measure the service performance (QoS) metrics periodically (recommended once every 100 ms)	5 seconds
Detect scale out/in decision	T_{start} = Scale out/in condition generated in the system T_{end} = Scale out/int decision detected and trigger to scale (warning or auto scale action) issued	< 1 second
Implement manual scale out/in	T_{start} = Request for manual scale out/in logged in the system T_{end} = Service scale out/in completed and service functional	< 1 second
Implement autoscale out/in	T_{start} = Auto scale request logged in the system T_{end} = Service scale out/in completed and service functional	< 1 second
Modify VNF configuration in service chain	T_{start} = Request for modify logged in the system T_{end} = VNF configuration modified and functional	< 1 second

The table above, which decomposes the service creation time into different sub times, shows that we can achieve the full onboarding within few minutes, which is in the order of magnitude (and even below) the KPI usually requested for this aspect (i.e., 90 minutes).

The main bottleneck for our case is the computational capacity of the servers hosting the edge and central cloud VIMs. They are portable devices and not telco-grade servers: with a fully data centre alike deployment, those figures can be easily lowered to numbers similar to the ones obtained by the Hamburg testbed.

4.5.2 Application KPIs

The final goal of the Network KPIs analysed above is to satisfy the requirement of the applications. In order to assert that this goal is satisfied, in this section we evaluate the performance of the application based on application-level KPIs. The application-level KPIs considered here correspond to standard KPIs in the domain of computer-user interfaces. Specifically, three application-specific KPIs have targeted for evaluation of the VR Application [Eve19], which correlate application usability metrics to network performance.

Note that the usability of any tool or system is not a quality that exists in any real sense. Instead, usability has to be viewed in terms of the context in which it is used, and its appropriateness to that context. In general, it is impossible to specify the usability of a system without first defining who are the intended users of the system and the tasks those users will perform with it [BKM08]. Defining "goal" as the intended outcome and "task" as a set of activities undertaken in order to achieve a specific goal, there can be some general classes of usability measure as [ISO9244-11-18] suggests:

- *Effectiveness*: accuracy and completeness with which users achieve specified goals
- *Efficiency*: resources used in relation to the results achieved
- *Satisfaction*: extent to which the user's physical, cognitive and emotional responses that result from the use of a system, product or service meet the user's needs and expectations

The precise measures to be used within each of these classes of metric can vary widely. For example, the types of the tasks could be varied and as a result, measures of effectiveness will be affected. Networked multi-user applications (when two or more users in remote locations perform a cooperative operation over a network connection) entail specific metrics which cover the classes of usability that are affected by the network performance. In general, the most critical issue for a networking application is that all the users share the same information in real-time so there needs to be no discrepancy between local and remote entities inside the common application environment. In the case of VR multi-user applications, this common application environment is in the form of a 3D virtual world that all users occupy and interact with at the same time, known as Collaborative Virtual Environment (CVE) [PK99].

The Palazzo Madama VR Museum application also belongs to last category. In any case, when there is high latency, the actions are transmitted after some amount of delay to all users. Beyond the obvious issues of synchronisation between users, a usual result of this delay is a jerky presentation of the environment, called jitter, as the network tries to transmit all information continuously. It is also known that low latency and high reliability are conflicting requirements [SMP+14]. Changes over the latency and low bandwidth during high load cases and actions can affect the users negatively in performing the requested tasks. These issues become more apparent in VR applications as the visual feedback in real-time is important in order to perform the tasks. With regards to the network performance, the main aim of the Quality of Service is to provide high standards regarding the bandwidth, latency and jitter. Considering the three usability measures of ISO 9244-11 mentioned above, KPIs like task completion rate and time on task can be metrics which follow the requirements of effectiveness and efficiency respectively. Furthermore, a questionnaire has been prepared in collaboration with WP6 and provided to the users after they complete the VR procedure to assess their overall satisfaction regarding the application's usability. Participants have to rate the quality of their experience during the use of the VR application while it is served over a mobile network with legacy technologies compared to the application being served in a network that uses 5G technologies developed in the project. The correlation between the answer scores and the actual KPI values is investigated to showcase the relation between network gains due to enabler use and QoE improvements in a quantitative manner.

In the following, we describe the relevant Application KPIs that we have measured in the testbed:

- ***Frame rate judder***: Flickering is a common issue in VR using headsets such as Oculus Rift. This issue has been defined as judder and it occurs when the frame rate drops below 75 fps. This judder can induce motion sickness and general discomfort in VR applications. The following metric is a standard and widely accepted metric to measure the judder:

$$\textit{Frame rate judder} = \frac{n/75}{\sum_1^n t_n \in (t > 1/75)}$$

  where t is the time required for each frame to render and n the total number of rendered frames. The formula above represents the percentage of time during a VR application where the framerate was less than 75 frames per second. Minimising this time reduces the probability of motion sickness. Inside the application, a number indicator is added in order to provide the frame rate. During the restoration procedure and when the user cuts some wood the simultaneously projection of different shaders may reduce the frame rate size.

- ***Task success rate***: Also known as task completion rate [Eve19], task success rate is the percentage of correctly completed tasks by users. As long as the task has a clearly defined goal or end point, such as grabbing an object, combine two objects, etc. the success rate can be measured. So before collecting data, it is important to define what constitutes success. The formula used for task success rate is as follows:

$$\textit{Task success rate} = \frac{CT}{A}$$

where CT is the number of correctly completed tasks and A is the total number of attempts. It is also important to track the first-time users' success rate and then track the progress: how the rate changes through time when users gain more experience with the service. The higher the task success rate, the better. Inside the application first a tutorial takes place regarding the movement and some basic interactions for the user. After the connection, the user has to complete a restoration tutorial of an artefact. There are many interactions where the task rate can be measured as pressing a button or grabbing an object from a table, i.e., when user tries to press an object and succeeded with third time, the task success rate is 1/3.

- ***Time on task***: This metric is basically the amount of time it takes the user to complete the task, expressed in minutes and seconds. Time on task data can be analysed and presented in different ways, but the most common way is to present the average time spent on each task. Generally, the smaller the time-on-task metric, the better the user experience. In the Touristic City testbed, the user has to make a restoration of an artefact. But before the connection be established an easy tutorial is starting in order a user with no previous experience in VR to get accustomed with the environment and the controllers. Right after the connection the guide describes the procedure step-by-step so as not to be confusing. Though, some tasks, like chopping, need more attention and detail to complete. Furthermore, an amateur user needs some time to get used the two hands interactions. Considering that the application takes place inside the museum an estimated time of 6 minutes should be enough to offer a complete and realistic experience with all tasks completed.

In general, testbed visitors used the application actually under two scenarios: a) with the network having free resources and a low load and b) with the network under a high load, so the elasticity algorithm must be triggered to optimise VNFs to serve the VR application with a high QoS. In the high load case, the extra network load was triggered in the beginning of a specific action, where the user had to cut a piece of wood used to restore an antique as described above. In all cases, the application had enough resources to operate in the optimal manner after the application of the elasticity algorithm. However, when the KPI values are compared for the period when the network is loaded, statistically significant differences are discovered for the Time on Task and Task success rate. A p-value of 0.0007706 and 0.009994 was measured for Time on task and Task success rate respectively, using the Kruskal-Wallis non-parametric statistical test. So, the users in the highly loaded scenario required more time and more tries to accomplish the task at hand when compared to the users of the light load case. These differences stop being significant if we use the heavy load scenario data only after the VNF migration. More details about the results can be found in [5GM19-D63].

The main conclusion that we draw from the analysis of Application KPIs conducted here is that the network service provided by the Touristic City testbed is adequate to satisfy the requirements of such a demanding application.

5 Summary of the experimental evaluation

In general terms, the main conclusion from the experimental evaluation reported in this document is that the 5G-MoNArch network architecture is technically feasible and is able to support applications with heterogeneous requirements by deploying specific network slices. This has been proven in both the Smart Sea Port and the Touristic City testbeds. This conclusion is supported by different experiments conducted for the two testbeds, reported in Sections 3.5 and 4.5, respectively. In this section, we first summarise the results obtained in terms of KPIs (Section 5.1), we then provide an overview of the functionality results (Section 5.2), and finally we report on the highlights and lessons learned from the experimental work (Section 5.3).

5.1 Summary of KPI results

In the following table, we summarise the main KPIs obtained from the experimental evaluation, providing a global view of the results obtained across the two testbeds. While these KPIs were defined in WP6 [5GM17-D61] [5GM18-D62], for convenience we provide in Annex 7.4 a summary of the KPI definitions. For each KPI, we show the results obtained from the testbeds and compare them against the objectives defined by the project[17]. Along with the results, we explain how the experiments contribute to showing that the project satisfies its objectives.

It is important to note that the objectives provided refer to the whole project evaluation and not only to the experimental results. Therefore, while experimental results may show that we meet some of these objectives, for other objectives this may be shown via simulation results. Indeed, for those KPIs that deal with large-scale scenarios or require tailored deployments, simulations are a more appropriate tool to evaluate their performance. The reader is referred to deliverable D6.3 [5GM19-D63] for a comprehensive evaluation of the project results. D6.3 puts together the experimental results reported here with the ones obtained via simulations and other means, and it shows how the project meets the targets for all KPIs.

Table 5-1: 5G-MoNArch performance KPIs

KPI	Testbed[18]	Project Objective	Testbed Measurements	Testbed Contribution to Project Objectives
Area traffic capacity	TC	Improvement by factor ~10	Experimental results show that we can dynamically assign the communications resources to a network slice according to the demands of the network slice. This can be seen from the throughput measurements performed in [GGS+18], in line with those shown in Section 4.5.1.2. Similar conclusions are obtained for compute resources; as a proof of concept, we can scale the higher layers and UPF machines of the eMBB slice to the double just when needed, duplicating the capacity on demand.	By dynamically assigning resources to slices, the efficiency in the usage of wireless area capacity is improved, which increases the effective area capacity: the same infrastructure can support more demand. Based on this functionality shown in the testbed, absolute figures on the gains are provided in [5GM19-D63].
Service creation time	TC	< 90 min	Both slices are successfully onboarded and configured in about 7 minutes.	The project objectives are not only met by the testbed results, but even substantially exceeded.
	SSP		Delay for Slice deployment with pre-reservation is < 1min, and for Blueprinting through GUI ~5 min.	

[17] The targets were identified in the DoW as well as in deliverables D6.1 and D6.2.

[18] SSP (Smart Sea Port) or (TC) Touristic City.

E2E Latency	TC	< 5 milliseconds	We can achieve latencies below 5 ms in the radio, as shown in Table 4-3. Figure 4-34 further shows that, when the elasticity/orchestration algorithm is applied, the E2E latency can be restored to 10 ms for the round-trip delay (i.e., 5ms one-way delays, meeting the project objective).	The project objective is met by the testbed results. Indeed, as long as a network slice uses the appropriate radio layer and leverages orchestration, it can meet the objective. In the Smart Sea Port testbed, delay is slightly larger; however, this objective was not set for this testbed, which has more relaxed latency requirements.
	SSP		Latencies down to 15 ms are achieved for a setup in cells of up to 10 Km and 1 ms TTI numerology. Using numerologies with shorter TTIs would reduce the E2E latency below the target requirements. Multi-connectivity and orchestration are shown to improve latency.	
E2E Reliability	SSP	99.999%	The testbed implements multi-connectivity and shows that it works. Results (Section 3.5.6) show the improvements provided in terms of latency. Also, they show that with multi-connectivity a significantly larger number of packets can be delivered within a given delay budget over handovers (50%-ile of the latency is reduced from 45ms to 23ms). Due to hardware limitations, packet losses cannot be directly measured.	The testbed serves to validate the multi-connectivity technology which is essential to provide reliability. By leveraging this technology and employing a suitable deployment, [5GM19-D63] shows via simulation that this objective can be met.
Relocation Delay	TC	No disruption	As shown in Section 4.5.1.4, the orchestration implementation can relocate VNFs within very few milliseconds, avoiding any service disruption	The testbed meets the 'no disruption' project objective.
Isolation	SSP	Avoid side-effects between slices	The results of Section 3.5.5 show that the performance of a slice is not affected by the behaviour of another slice. Indeed, user planes are decoupled and operate independently, and in case of failure of one user plane, the other is not impacted.	The testbed meets the project objective of providing effective isolation.

5.2 Summary of functional results

In addition to the measurement and validation of KPIs reported above, one of the major outcomes of the testbed has been the verification of the feasibility of the technical concepts addressed by the project, by implementing the associated functionalities and verifying their operation. The table below provides a summary of the key functional results of the testbeds. These functionalities do not only provide the basis for evaluating the KPIs reported above, but they also provide the technology enablers needed to evaluate the project key innovations.

As we can see in the table, the two testbeds implement the necessary functionality for the deployment and orchestration of network slices, relying on common enabling technologies such as network slicing, orchestration and virtualisation. Furthermore, when deploying customised network slices to address the respective use cases, each testbed implements additional functionality specific to the testbed: (i) the Smart Sea Port testbed implements multi-connectivity and security in order to evaluate the reliability concept, and (ii) the Touristic City testbed implements VNF relocation and dynamic resource allocation in order to evaluate the elasticity concept. Furthermore, the Smart Sea Port testbed also provides a network slice management tool, to enable the vertical tenant to manage its slice, and the Touristic city testbed provides a low latency radio layer to achieve the latency requirements.

Table 5-2: 5G-MoNArch functional results

Functionality	Testbed	Project implementations and achievements
Network slicing	TC	Two different network slices have been deployed which provide URLLC and eMBB services, respectively. Slices are isolated both for radio and compute resources and provide different latency performance.
	SSP	Three different network slices are implemented, one for each use case. Different slices employ independent user planes, are isolated and provide different latency performance. Multi-slice terminals have been deployed, which can connect to multiple slices simultaneously.
Orchestration	TC	Dynamic orchestration is provided wherein VNFs can be relocated on the fly based on slice requirements and load conditions. Re-orchestration can be triggered, for instance, when latencies grow too much and VNFs need to be relocated close to the end-users to reduce latency.
	SSP	Different orchestration deployments are provided for different slices. Specifically, the user plane functionality is located at different nodes for different slices, namely in the edge and the central cloud, respectively. Experiments show that smaller latencies are achieved in the former case.
Virtualisation	TC	The upper layer functions of the protocol stack (RAN Higher Layers, UPF and application servers) are virtualised. They are VNFs that can be orchestrated on demand for the different slices.
	SSP	The user plane is virtualised and can be orchestrated at different locations for different slices.
Multi-connectivity	SSP	This is a fundamental technology enabler for reliability. This functionality has been implemented, and experiments have been conducted to assess its operation as well as its impact on improving latency as well as handover performance.
Security Trust Zone	SSP	The security Trust Zone mechanism has been implemented in a simulated Smart Sea Port testbed to demonstrate security protection capabilities, and the performance of this mechanism has been evaluated for a number of different attacks. It is an essential feature in order to deploy reliable slices.
Network slice management tool	SSP	The Network slice management tool allows to manage the lifecycle of network slices. In particular, it allows for preparing slice blueprints, and to commission and decommission network slices on the fly.
VNF relocation	TC	This functionality allows to relocate VNFs from one node to the other without any perceptible service interruption. It can be employed to relocate VNFs close to the end-users to reduce latency, or to rebalance load in case of outages. It is thus an essential enabler for the key innovation of ‘orchestration-driven elasticity’.
Dynamic resource allocation	TC	The ability to dynamically re-allocate resources is essential for the ‘slice-aware elasticity’ concept. This refers to both computation and communication resources. For compute resources, by performing a continuous and optimal orchestration of the VNF, the system always uses the optimal amount of cloud resources. In case of resource shortage, VNF resources can be scaled up when the resource utilisation reaches critical points. For communication resources, the allocation of radio resources can be performed dynamically based on the varying demands of the slices.
Low-latency radio layer	TC	In order to meet the stringent delay requirements of the URLLC slice, a radio layer providing low latencies has been developed. This adds to the re-orchestration functionality, which allows to place the VNFs impacting

		latency close to the end-users and further contributes to achieving low latencies.

5.3 Other highlights and lessons learned

In addition to the experimental results and KPIs obtained from the testbeds, and the validation of the project's enabling functionalities, there are a number of additional highlights and lessons learned from the testbed. For the Touristic City testbed, we can highlight the following results:

- **ETSI ENI involvement**: This testbed focuses on one of the user cases identified by the ETSI ENI standards groups and is one of the official Proof of Concepts of this group. By means of the testbed, we have provided feedback to the ETSI ENI System architecture, especially on the interface definition and the orchestration call flow. This feedback was provided during a past meeting, while the formal submission to the ETSI ENI document will be performed after the release date of this deliverable.
- **Mapping service requirements into orchestration**: The standardisation of the link between the service requirements defined by the application and the network orchestration is an ongoing activity (by GSMA, for instance [5GM17-D21]). For this testbed we had to iterate several times before finding a correct definition of the needed KPI and the network capabilities. The outcome of this interaction is exemplified in the GUI.
- **Importance of elasticity**: Elasticity and AI can be very important in the future networks. From our tests we understood that the capability of dynamically matching the network slice demand (even in a smaller scenario such as the TC one) will be paramount for the sustainability of future 5G Networks. Allowing operators to save resources while providing very challenging services is key to the success of beyond 5G Networks.
- **Alignment with elasticity and experiment-driven optimisation**: In terms of the main project objectives, it is worth highlighting that slice-driven and orchestration-driven elasticity have been demonstrated. Also, with respect to the support to experiment-driven optimisation, the testbed has been used to gain insight into the performance of different VNFs (UPF and RAN Higher Layers) in terms of CPU consumption.
- **Feedback from Touristic-driven verticals**: In occasion of the Touristic City testbed event organised in Turin from the 22nd to the 24th of May 2019, the Municipality of Turin and Fondazione Torino Musei provided very positive feedback about the testbed itself and the innovations introduced by 5G-MoNarch. The VR application was very well perceived as a means to enhance the traditional visit to the museum. Such positive impressions were confirmed by the request of the Municipality of Turin to have the testbed available for a further period after the event.

For the Smart Sea Port testbed, we can also highlight a number of relevant results beyond the KPIs and functional results:

- **Industrial application of network slicing**: With the Hamburg testbed, the 5G-MoNArch project was able to demonstrate network slicing in a life industrial campus network; it offered the opportunity to show how to integrate mobile networks with industrial IT networks, how network slices can help to extend existing networks, and how 5G technologies will improve the performance of mobile networks.
- **Integration of solution in commercial equipment**: The main challenge has been the integration of novel concepts and approaches with existing hardware and software in order to perform large testbed measurements; as well as the integration across different network domains such as between the mobile network domain and vertical's IT domain.
- **Improving the operation of industrial campuses**: Mobile networks can significantly improve the operations of large industrial campuses, and the pure connectivity of devices is only one aspects of vertical deployment, where the integration into existing IT systems, business models, and new processes within companies are very important.
- **Providing reliability**: In terms of the alignment with the main project Objectives, it is worth highlighting that the Hamburg testbed focused on demonstrating the capabilities of network

slices in a practical deployment, and on testing multi-connectivity for improved reliability. Both objectives have been fulfilled, i.e., using a self-developed life cycle management tool, it is possible at any point in time to activate or deactivate network slices; using multi-connectivity, we were able to show that the average latency can be significantly reduced due to a lower number of dropped packets.

- **Feedback from an industry vertical**: HPA gained great insights in 5G technology by 5G-MoNArch. HPA's business requires different kinds of connectivity: Legacy copper cables (even analogue), fibre channel, Wi-Fi, directional radio and proprietary radio technology for maritime navigation are in production in Port of Hamburg, today. Therefore, HPA exploited 5G network slicing for two reasons. Firstly, with the broad range of communication technologies in production, HPA always evaluates ways to operate the port-infrastructure more economically. Secondly, HPA is looking for technologies, which allow HPA to connect assets which cannot be sufficiently connected with today's technology. For both areas, HPA's involvement in 5G-MoNArch gave a great insight in future capabilities of 5G network slicing in the applications in Port of Hamburg. 5G is seen as a key technology for future use-cases in Port of Hamburg. Knowledge and practical experience on network slice integration into IT-processes were proven, and practical experience with the implementation of multiple applications in a 5G-network was successfully acquired. Network slicing was identified as technology supporting emergency management in Port of Hamburg and manifold ideas for new use cases in Port of Hamburg were created.

6 Conclusions

In this document we have described the testbed activities and the experimental evaluations performed by the 5G-MoNArch project in the context of the WP5. These activities revolve around the two project testbeds: The Touristic City testbed and the Smart Sea Port testbed. These testbeds address the main technical concepts associated to 5G-MoNArch architecture: Network Slicing, Network Orchestration, Resilience, Reliability, Security and Elasticity. For each testbed, in this document we have described the different use cases, their implementation and evaluation results.

Both testbeds are well linked with the work carried out in other work packages in 5G-MoNArch. Indeed, the architectural concepts proposed in the framework of 5G-MoNArch WPs 2, 3, and 4 are closely related to the work performed in WP5. Based on the work conducted in WP2, fundamental aspects such as slice blue print and slice LCM functions were designed and implemented in both testbeds. The main outcome and results of the research activities carried out in WP3 on reliability have been implemented in the Smart Sea Port testbed, and those carried out in WP4 on Network Elasticity have been implemented in the Touristic City testbed.

In the Smart Sea Port testbed three different innovative use cases have been addressed: traffic flow control, enhanced maintenance experience and an improved pollution control. To realise them, we have modified pre-commercial equipment to support network slicing as well as to provide resilient and reliable NFs. To satisfy the high resilience and reliability requirements, the testbed includes multi-connectivity implemented in the radio part and a fault-management approach. The multi-connectivity uses data duplication to increase the link reliability with the objective to prevent service interruption during mobility events; the fault-management techniques are used to identify, trouble shoot and isolate occurring network faults, which includes the monitoring tools for the detection of changes, potential problems and anomalies in the network behaviour.

In the Touristic City testbed, we have realised an innovative use case consisting on a virtual visit to the Palazzo Madama museum in Turin using VR. This requires two different network slices: an eMBB slice for the video streams transmission and a URLLC slice for those activities requiring real-time interactions. These slices have been implemented with research prototypes supporting E2E network slicing and including 5G New Radio. Several functions in the user-plane (such as the higher layer of the RAN and the UPF) have been virtualised and orchestrated in order to provide the desired behaviour. Elastic orchestration and resource allocation have been investigated, with a special emphasis on Artificial Intelligence and Machine Learning techniques (aligned with the ETSI ENI specification) to determine the best placement of VNFs (between central/edge clouds) and to design elastic NFs capable of resource adaptation and utilisation depending on the network conditions.

The testbeds deployed in this project have gone beyond laboratory implementations, such as the ones provided 5G-PPP Phase 1 projects. In contrast, we have provided a much more realistic deployments, as shown by the following facts:

- Both testbeds have involved real vertical stakeholders: the Hamburg Port Authority and the Municipality of Turin together with the Fondazione Torino Musei. This has allowed to interact with the actual customers of the technology and get their feedback.
- The 5G-based applications in the Hamburg port deal with the real port operation that could serve as a basis to develop real-life services in Hamburg.
- The Touristic City testbed has involved real users: the tourists visiting the Palazzo Madama. This has allowed to evaluate the benefits of the technology in a very realistic environment.

It is important to note that we have obtained specific KPIs validating that 5G can achieve the requirements of typical industrial and consumer applications. For many of the KPI targets that 5G-MoNArch aimed to meet, the testbeds have sufficed to show that the developed technology is actually able to fully achieve these KPI targets. For other KPIs involving larger scale deployments, the testbed validates the various mechanisms and enablers, while simulations have been conducted showing that these mechanisms deliver the desired target KPIs (see Deliverable D6.3). Beyond the quantitative results showing low service deployment times or reduced latency, functional results show that the 5G-MoNArch architecture can be used to deploy multiple isolated slices with different features.

It is important to note that both testbeds have received recognition in relevant public forums: The Touristic City testbed was selected as Proof of Concept (PoC) for the ETSI ENI ISG due to its very high innovation potential and for the usage of different AI techniques for the M&O algorithms (this makes the Touristic City testbed an official ENI testbed). On the other hand, the Smart Sea Port testbed received the Global Mobile Award in the category "5G – Industry Partnerships" at the 2019 Mobile World Congress.

Finally, it is also worth highlighting that both testbeds have been shown in very successful events that have served to show 5G-MoNArch technology to the different stakeholders of the future 5G ecosystem. Special attention in these events has been devoted to the vertical players, as these are the customers of the technology and hence essential players for the success of the technology. Very positive feedback from them has been received in these events.

As a main conclusion, we consider that with the WP5 testbeds it has been demonstrated that the 5G-MoNArch architecture can actually give direction for transforming the telco industry and opens the door to a broad spectrum of vertical industries to integrate their services in the upcoming 5G networks in a reliable and secure way.

7 Annexes

7.1 Description of network slices for the Smart Sea Port testbed

The following Table 7-1 lists parameters for the description of the network slices that have been setup in the 5G-MoNArch testbed in Hamburg. This set of parameters has been improved in the course of the project; finally, the terminology and the description of parameters have been adapted to the GSMA GST [GST], where applicable.

For parameters shown in black font colour, the naming has been taken from the GSMA GST, while parameters *in italics* have been defined by 5G-MoNArch and do not exist in the GSMA GST in this form.

Table 7-1: Smart Sea Port testbed slices properties

GSMA Attribute / ***5G-MoNArch Slice property***	**Explanation of slice property**	**Value range**	**Traffic Lights Control**	**Environmental Sensors**	**AR/VR/Video Streaming**
Purpose	*Description that helps the network operator to understand the needs of the tenant and the users*	*Plain text*			
Slice / Service Type (SST) (implicitly by selection of NEST)	*MBB, URLLC, MIoT, other*	*{MBB, URLLC, MIoT, other}* *Ref.: 3GPP 23.501, Sec. 5.15.2*	URLLC	mMTC	eMBB
Tenant-specific slice functionality					
Data flows		n/a			
Support for non-IP traffic	This attribute provides non-IP Session support (Ethernet session and forwarding support) of communication devices.	• 0: not supported • 1: supported	0: not supported (only IP packets)	0: not supported (only IP packets)	0: not supported (only IP packets)
User Data Access	The attribute defines how the slice (or mobile network) should handle the user data. The options are as follows: • The device has access to the Internet • All data traffic is routed to the private network (e.g. via tunnelling mechanism such as L2TP, VPN tunnel, etc.) • All data traffic stays local and the devices	• 0: Direct internet access • 1: Termination in the private network • 2: Local traffic (no internet access)	1: Termination in the private network	1: Termination in the private network	1: Termination in the private network

	do not have access to the Internet or private network				
Gateway (GW) for user traffic	*Node (i.e. internet peering point or tenant host / GW) to which data coming from the terminals are forwarded*	*IP address*	- project-internal -	- project-internal -	- project-internal -
GW for network management traffic and Business Support System (BSS) access	*GW towards tenant's BSS*	*IP address*	n/a in testbed	n/a in testbed	n/a in testbed
Positioning Support	This attribute describes if the slice provides geo-localisation methods or supporting methods.	Availability: • 1: CID • 2: E-CID (LTE and NR) • 3: OTDOA (LTE and NR) • 4: RF fingerprinting • 5: AECID • 6: Hybrid positioning • 7: NET-RTK Prediction Frequency: • 1: per second • 2: per minute • 3: per-hour • 4: conditional, e.g. in case a specific area is entered or left	not required	not required	not required
NSC NFs	An NSC can own some NFs (e.g. UPF, Unified Data Management (UDM)). This attribute provides a list of NFs to be provided by the NSC.	List of NFs, e.g. • UPF • UDM/AUSF • AF • Policy Control Function (PCF)	none	none	none
Data processing capabilities for NSC NFs			no service-related data processing functionality required	no service-related data processing functionality required	no service-related data processing functionality required
Data storage capabilities for NSC NFs			no data storage capabilities required	no data storage capabilities required	no data storage capabilities required

Security functionality	*Security functions that shall be provided by the network operator to the tenant, e.g. encryption, VPN, Firewall, spam filter, fraud detection, …*	*List of security functions*			
Supported Device Velocity	Maximum speed supported by the slice at which a defined QoS and seamless transfer between TRxPs, which may belong to different deployment layers and/or radio access technologies (multi-layer /-RAT), can be achieved.	• 1: Stationary: 0 km/h • 2: Pedestrian: 0 km/h to 10 km/h • 3: Vehicular: 10 km/h to 120 km/h • 4: High speed vehicular: 120 km/h to 500 km/h	1: Stationary: 0 km/h	3: Vehicular: 10 km/h to 120 km/h	1: Stationary: 0 km/h
Generic slice functionality / deviations from generic slice functionality	*Functionality of PNFs or VNFs (e.g. acc. to 3GPP specifications) can serve as base line, if appropriate; deviations (unused or additional functions) should be specified.*				
Shared NFs	*E.g. common PHY, common MAC, common PDCP*		Shared radio resources, PHY, MAC	Shared radio resources, PHY, MAC	Shared radio resources, PHY, MAC
User Management Openness	This attribute describes the capability that NSC could manage their users or groups of users' network services and corresponding requirements. For instance, if NSC Y orders a slice which is capable to support X users of Y, then Y should be capable to decide which X users could use this slice. Hence, Y could manage the users, in terms of add, modify or delete users to receive network services provided by the specific slice.	• 0: not supported • 1: supported			
Multicast	Define whether the slice supports multicast functionality.	• 0: not supported • 1: supported	0: not supported	0: not supported	0: not supported
Group Communication Support	This parameter describes which type of group communication is provided by the slice.	• 0: not available • 1: SC-PTM • 2: Broadcast/Multicast • 3: Broadcast/Multicast + SC-PTM	0: not available	0: not available	0: not available
Location-based Message Delivery	This attribute describes the delivery of information in a particular geographic region.	• 0: not supported • 1: supported	0: not supported	0: not supported	0: not supported

V2V Support (PC5)	This parameter describes if the V2X communication mode is supported by the slice.	• 0: NO • 1: YES-EUTRA • 2: YES- NR • 3: YES -NR and E-UTRA	0: NO	0: NO	0: NO
Mission-critical service support	This attributed specifies whether or not the slice supports mission-critical push-to-talk (MCPTT), mission-critical data (MCData), mission-critical video (MCVideo), Isolated E-UTRAN Operation for Public Safety (IOPS) or mission-critical interworking	• 1: MCPTT • 2: MCData • 3: MCVideo • 4: IOPS • 5: MC interworking	not required	not required	not required
Session and service continuity support	The attribute defines the continuity of a Packet Data Unit (PDU) session.	• 0: none • 1: SSC mode 1 • 2: SSC mode 2 • 3: SSC mode 3	0: none	0: none	0: none
MMTel Support	This attribute describes whether or not the slice supports IP Multimedia Subsystem (IMS) Multimedia Telephony Service MMTel. This parameter describes whether an IR.92 compliant MMTel deployment is supported in the slice.	• 0: not supported • 1: supported	0: not supported	0: not supported	0: not supported
Other	*Examples: Relocation of terminal-related context data between data centres when terminal moves, ...*				
Geography Parameters					
Coverage	This attribute specifies the coverage area of the slice - the area where the terminals can access this particular slice.	• 1: Global • 2: National • 3: Regional • 4: Local (outdoor) • 5: Local (indoor)	4: Local (outdoor) Streets in the area supervised by the traffic management office of the city of Hamburg	4: Local (outdoor) River Elbe from km x to km y	4: Local (outdoor) HPA premises / construction sites in the port area of Hamburg
Terminal Density	This attribute described the maximum number of connected and/or accessible devices per unit area (per km2) supported by the slice.	Number per km²	n/a in testbed	n/a in testbed	n/a in testbed
Geographic limitations for data centres and backhaul connections	*Allows to exclude data centres or backhaul connections from serving the slice, e.g. for data privacy or redundancy reasons*		EU/Germany -> dependent on current	EU/Germany -> dependent on current	EU/Germany -> dependent on current

			EU-DSGVO discussions	EU-DSGVO discussions	EU-DSGVO discussions
Traffic profile parameters					
DL throughput per slice (minimum, maximum)	The overall achievable data rate of the slice in DL that is available ubiquitously across the coverage area of the slice.	Bytes/second	Peak: 256 Kbit/s Avg: 16 Kbit/s	Peak: 1 Mbit/s Avg: 100 Kbit/s	
DL throughput per user (guaranteed, maximum)	The first parameter describes the minimum data rate supported by the slice per UE in DL, which is required to achieve a sufficient quality experience (dependent on the selected service type) and can be seen as a guaranteed throughput. The second parameter describes the maximum data rate supported by the slice per UE in DL.	Bytes/second	Peak: 256 Kbit/s Avg: 16 Kbit/s		Peak: 100MBit/s Avg: 5MBit/s
UL throughput per slice (minimum, maximum)	The overall achievable data rate of the slice in UL that is available ubiquitously across the coverage area of the slice.	Bytes/second	Peak: 256 Kbit/s Avg: 16 Kbit/s	Peak: 1 Mbit/s Avg: 100 Kbit/s	
UL throughput per user (guaranteed, maximum)	The first parameter describes the minimum data rate supported by the slice per UE in UL, which is required to achieve a sufficient quality experience (dependent on the selected service type) and can be seen as a guaranteed throughput. The second parameter describes the maximum data rate supported by the slice per UE in UL.	Bytes/second	Peak: 256 Kbit/s Avg: 16 Kbit/s		Peak: 100MBit/s Avg: 5MBit/s
Deterministic Communication	This attribute defines if the slice supports deterministic communication for periodic user traffic. Periodic traffic refers to the type of traffic with periodic transmissions.	• 0: not supported • 1: supported	bursty	steady raw data stream	Start at purely random points in time, then steady usage of high bandwidth
Periodicity	This parameter provides a list of periodicities supported by the slice.	Seconds			
Max. Supported Packet Size	This attribute describes the maximum packet size supported by the slice and may be important for URLLC and MIoT case, or to	Bytes			

	indicate a supported maximum transmission unit (MTU).				
Slice Quality-of-Service Parameters (5QI)	This attribute defines all the QoS relevant parameters supported by the slice. For some of these parameters 3GPP already defined standard values. By preselecting a 5G QoS Identifier (5QI) these parameters will automatically be filled with the standardised values.	5QI value specified by 3GPP			
5QI Packet Delay Budget	The Packet Delay Budget (PDB) defines an upper bound for the time that a packet may be delayed between the UE and the UPF that terminates the N6 interface. For a certain 5QI the value of the PDB is the same in UL and DL. In the case of 3GPP access, the PDB is used to support the configuration of scheduling and link layer functions (e.g. the setting of scheduling priority weights and HARQ target operating points)	Seconds	Real-Time Protocol (RTP) / Latency must be comparable to fiberchannel	Steady stream: latency is not of relevance	latency < 20 ms. Optimum is 5 ms
5QI Jitter	Jitter is defined as a variation in the delay of received packets. At the sending side, packets are sent in a continuous stream with the packets spaced evenly apart. Due to network congestion, improper queuing, or configuration errors, this steady stream can become lumpy, or the delay between each packet can vary instead of remaining constant.	Seconds			
5QI Packet Error Rate	The Packet Error Rate (PER) defines an upper bound for the rate of PDUs (e.g. IP packets) that have been processed by the sender but that are not successfully delivered by the corresponding receiver. The purpose of the PER is to allow for appropriate link layer protocol configurations (e.g. RLC and HARQ in RAN of a 3GPP access). For all 5QIs the value of the PER is the same in UL and DL. For GBR QoS Flows with Delay	Percentage	not tolerable	dependent on sensor type: tolerable in most cases	tolerable (like Video stream today)

	critical GBR resource type, a packet which is delayed more than PDB (but which complies with the GFBR and MDBV requirements) is counted as lost and included in the PER. Delayed packets are not included in the PER if a GBR QoS Flow with a Delay critical resource type is exceeding the GFBR and the Maximum Data Burst Volume.				
Reliability	This attribute describes in percentage the amount of sent network layer packets successfully delivered to a given system node (incl. the UE) within the time constraint required by the targeted service, divided by the total number of sent network layer packets.	Percentage	Highest possible reliability, no packet loss tolerable	dependent on sensor type: low reliability / packet loss tolerable in most cases	Low reliability / packet loss tolerable (like Video stream today)
Availability	This parameter specifies the availability provided by an instance of this slice, expressed as a percentage. This contributes to, and may be different from overall service availability.	Percentage			
Number of Connections	This attribute describes the maximum number of concurrent sessions supported by the slice.	Number			
Terminal devices					
Number of Terminals	This attribute describes the maximum number of terminals supported by the slice.	Number	1	2	1
Supported Access Technologies	This attribute defines which access technologies are supported by the slice.	• 1: GERAN • 2: UTRAN • 3: E-UTRA • 4: NR • 5: LTE-M • 6: NB-IoT • 7: Wi-Fi • 8: Bluetooth • 9: Fixed access, e.g. DSL, Fibre	3: E-UTRA	3: E-UTRA	3: E-UTRA
Radio spectrum	Defines the radio spectrum supported by the slice. This is important information, as some	Frequency band according 3GPP 38.101	All bands permitted	All bands permitted	All bands permitted

	terminals might be restricted in terms of frequencies to be used.				
Security / privacy requirements					
Expected vulnerabilities and threats	*Particular interest should be on vulnerabilities and threats that arise from the specific purpose of the network slice; protection of a slice against "normal" vulnerabilities and threats should be done anyway*	*Plain text*	(Terroristic!) Attack to critical infrastructure	Hacker attack to manipulate data	Attack to breach into HPA's network
Protection mechanisms	*Which mechanisms are needed, and how do they have to be applied?*				
Slice isolation	Isolation is one of the key expectations of network slicing. An NSI may be fully or partly, logically and/or physically, isolated from another NSI [24]. This attribute describes different types of isolation	• 0: No Isolation • 1: Physical Isolation • 2: Logical Isolation	• 2: Logical Isolation critical infrastructure (today, traffic lights are connected in a redundant way)	• 2: Logical Isolation Performance isolation necessary to a certain measure. Application is important for HPA'S business, but not critical	• 2: Logical Isolation Performance isolation not crucial
Slice isolation (logical isolation)	Logical – slices are logically separated		Yes	Yes	Yes
isolation of user traffic (user data)			isolated for each UE	isolated, but sensors may be allowed to communicate with each other (use case e.g. noise reduction by integration other sensor's data)	not isolated; possible application of service chaining in the 5G network itself to perform business logic on the transferred data
isolation of control and meta data	*incl. Mobility support/management of terminals, movement profile, session data*		isolated for each UE	not isolated / future production ready IoT-Sensors with integrated UE could be managed by slice	not isolated; service chaining could also be based on metadata (e.g. position tracking)
isolation of subscriber profile	*incl. data for encryption and integrity protection* *In particular:* *- Is the Home Subscriber Server (HSS)*		Highest security level possible		

	operated by tenant or MNO? *- If HSS operated by MNO: Joint HSS with other tenants possible or isolation between tenants required?*				
Slice isolation (physical isolation)	Physical – slices are physically separated (e.g. different rack, different HW, different location, etc.)		No (Separate VLAN)	No (Separate VLAN)	No (Separate VLAN)
Authentication, access admission					
Encryption			E2E	E2E	E2E / Service chaining must still be possible
Integrity protection			highest level	mid-level	Package loss and failures in streaming data is acceptable to a certain level
Firewall			yes	yes	yes
Operational Aspects					
Frequency of slice activation / termination	*{e.g. yearly … daily, on demand,…}*		always on	slice is constantly deployed / possibly separate slices for different sensor types / more flexible lifecycle for certain use cases possible	possibly separate slices for different construction projects --> suppliers could be integrated into the project's slice
Mission-Critical Capability Support	This parameter specifies what capabilities are available to support mission-critical services. More than one capability may be supported at once. This parameter is optional when “Slice Priority” is set to 1 (meaning mission-critical slice).	• 1: Inter-user prioritisation • 2: Pre-emption • 3: Local control	• 2: Pre-emption		
Pre-emption capability	*In case of resource shortage, is the slice permitted to withdraw resources from other slices? (Example: Slices for public safety communications)*	*• 0: not supported* *• 1: supported*	• 1: supported	• 0: not supported	• 0: not supported
Pre-emption vulnerability	*If another slice shall be setup despite actual resource shortage: Can the resources*	*• 0: not supported* *• 1: supported*	• 0: not supported	• 1: supported (in urgent cases)	• 1: supported

	assigned to the slice be withdrawn by the other, new slice? (Example: New slice for public safety communications to be set up. Is an MBB slice vulnerable in the sense that it has to release its resources in favour of the new public safety slice?)				
Slice Priority	Mission-critical (MC) leads to a priority of the slice relative to others, for C-plane and U-plane decisions. This is relative to a customer provider relationship and to a Public Land Mobile Network (PLMN).	• 0: non-mission-critical • 1: mission-critical	highest	mid-level	normal
Root Cause Investigation	Root cause investigation is the capability provided to NSC to understand or investigate the root cause of network service performance degradation or failure.	• 0: not available • 1: passive investigation • 2: active investigation			
Alarming	*How can the network slice inform the application and the terminals about network problems?*		There needs to be a communication channel between Port Road Traffic centre and the slice LCM.	Sensors are able to buffer raw data for short period of time. After the buffer overflows, data is lost and the real-time status of port environmental measurement is broken.	User Information, if UE is integrated into device; Notification to HPA Business Units by slice LCM?
Performance Monitoring	This attribute provides the capability for NSC to monitor Key Quality Indicators (KQIs) and KPIs. KQIs reflect the E2E service performance and quality while KPIs reflect the performance of the network.	List of KPIs and KQIs to be monitored	overall KPIs on performance, reliability, influences into the radio frequencies by external parties; planned maintenance alerts	overall KPIs on performance, reliability, influences into the radio frequencies by external parties; planned maintenance alerts	overall KPIs on performance, reliability, influences into the radio frequencies by external parties; planned maintenance alerts
Private Operation	This attribute describes the capability to operate private 5G networks/slices in a predefined area.	• 0: not supported • 1: supported			
Manager of the network slice	*Slices can be managed in 3 ways.* *1) slice-provider: The slice-provider requires*		none	slice LCM / UE integration	Slice addition and LCM

	to manage this type of slices for the tenant as the tenant cannot manage the slice by itself. 2) Tenant : The tenant requesting slice would like to be able to manage the slice by itself and requires the slice provider to provide it with proper interfaces and permissions so that it can manages the slice features as it finds it fitted for its services. 3) Tenant and slice-provider: The tenant is requesting the slice-provider to manage some parameters of slice while the tenant stay in charge of the rest.				
Expected slice behaviour					
Resource shortage	*How does the tenant's application react on resource shortage, and what are the consequences for design of the network slice? (resource shortages are expected to last from some seconds to potentially many minutes)*	*Plain text*	Traffic lights control units are designed to run autonomously. Even if the slice breaks down completely, the control unit falls back to a safe mode.	Sensors are able to buffer raw data for short period of time. After the buffer overflows, data is lost and the real-time status of port environmental measurement is broken.	Engineers work is interrupted or legacy planning tools (paper plans) are used. Loss in efficiency of workforce.
Network faults	*How does the tenant's application react on network faults, and what are the consequences for design of the network slice? (network faults are assumed to cause service interruption lasting from some seconds to potentially many minutes)*	*Plain text*	see resource shortage	see resource shortage	see resource shortage
Expected temporal fluctuation of required slice capacity	*Are there major fluctuations in the expected overall traffic volume or number of terminals using the network slice? How large?*	*Plain text*	Constant, possibly mobile traffic lights (<20 available) are set up dynamically	Dynamic movements of ships could lead to temporary clustering in one area of the port	Clustering of engineers on certain sites possible

7.2 Details on the avatar's development

Creating a humanoid avatar from scratch requires the usage of a 3D computer graphics SW. Autodesk Maya [May19] and the open-source Blender [Ble19] are some of the most commonly used SW in the market. The main stages of the development are 3D Modelling and Rigging.

Rigging is the procedure of creating a skeleton consisting of a bone hierarchy. Each bone is usually parented to a previous bone which is influencing its pose, and consists of a peak, a joint and target spatial point. Once created, the appropriate vertex groups and vertex weights for each group, usually via a method called weight painting, are assigned to the bone to have control proportionated to the control a bone or a lever would have on the same object. An end-effector is the free end of the chain of alternating joints and bones and it is not a joint and an avatar can have more than one. Moreover, body parts which stabilise the rigs posture, such are the feet on humanoid avatars, are implemented as IK bones, that do not match the actual body structure, but provide a better control over the avatar. The rigging allows a 3D model to be animated in an articulated manner. An articulation is a rotation/translation of a joint which moves a connected bone. On the other hand, the pose is a set of joint articulations which results in positioning the articulated body. In the case of an avatar the rigging is the skeleton that ties in to the human posture.

As a whole, the development of an avatar includes four basic methods:

1. UV mapping, which refers the 3D modelling process of projecting a 2D image to a 3D model's surface for texture mapping.
2. Texturing, which refers how the avatar surface looks, including colour and resolution.
3. Retopology, which refers the method of recreating an existing mesh with more optimal geometry. For example a background not animated mesh has to have a very light mesh, with as few vertices as possible, while a humanoid mesh requires extra vertices in the joint areas and especially on the heavily animated areas (such as the face)
4. Sculpting, used to add the final details to a mesh, like brushing textures or directly sculpting a heavy mesh to alter the geometry.

The major steps were followed in order to create 3D avatars with high quality texture. Figure 7-1 shows some of the used options.

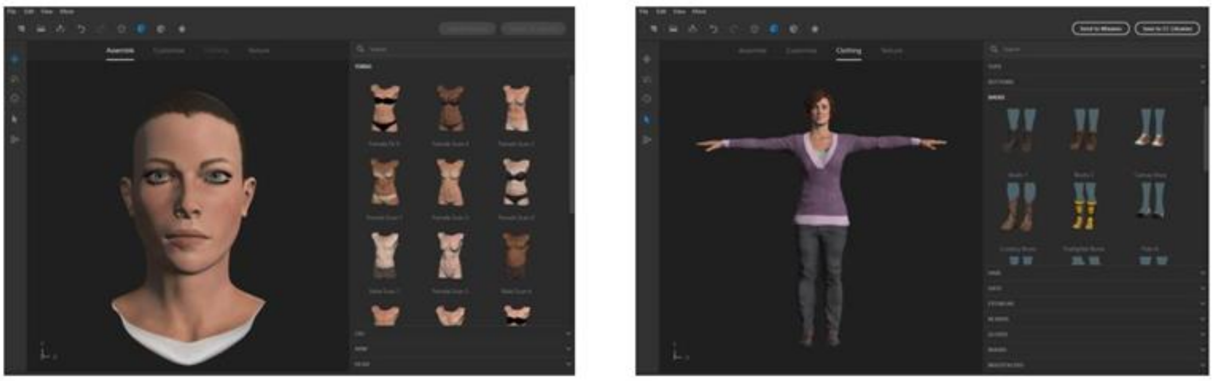


Figure 7-1: Modelling stage

When completed, the humanoid avatar can accept the rigging. The most common skeleton used in the animation industry starts from hips and keeps the most important bones through spine, legs and hands, which are responsible for the movement; they also include the exact bones a human has in hands' fingers, which is used for grabbing objects. Figure 7-2 shows this structure within the avatar.

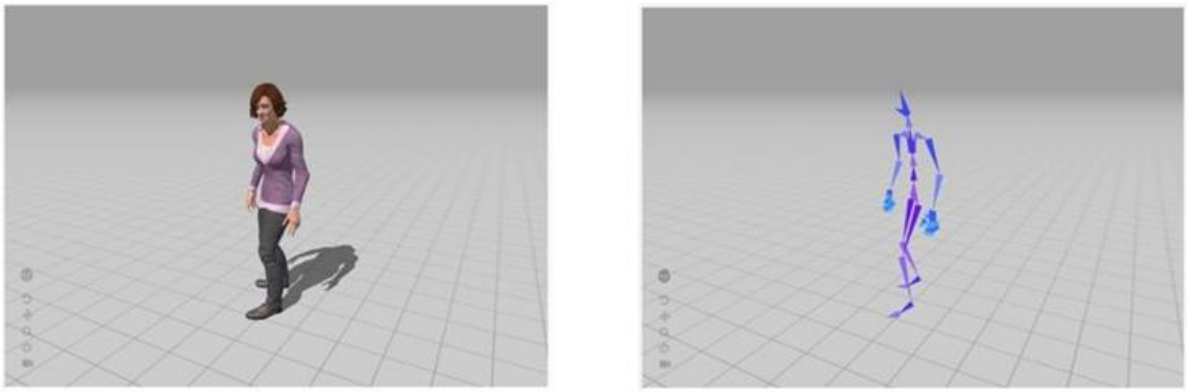


Figure 7-2: Rigging stage

The humanoid rigged avatar can be exported in a so-called “fbx” file, which is ready for implementation in Unity3D [Uni19]. The usage of the final avatar with this method provides a fair moving character in projects where avatar is controlled through scripts and keyboards (e.g. in a third person videogame). Though, in case of VR applications, a further procedure needs to take place considering especially on skinning and the rigging stages as described before. There should be a check on weights and influence that bones have on each other and other parts of the body. Weight is described through the visual presentation of opposing forces. Influence shows how a movement of a joint (bone) affects the other parts of the body (e.g., when the shoulder rotates not only the arm is moving; nearby parts might have also some position changes). Fingers must have the same influence on each stage, otherwise they do not have a human behaviour in interactions, so the avatar could make wrong hand gestures. Figure 7-3 shows a sample of the procedure over the hands. All these connections aiming to realistic humanoid movement and interactions like grabbing.

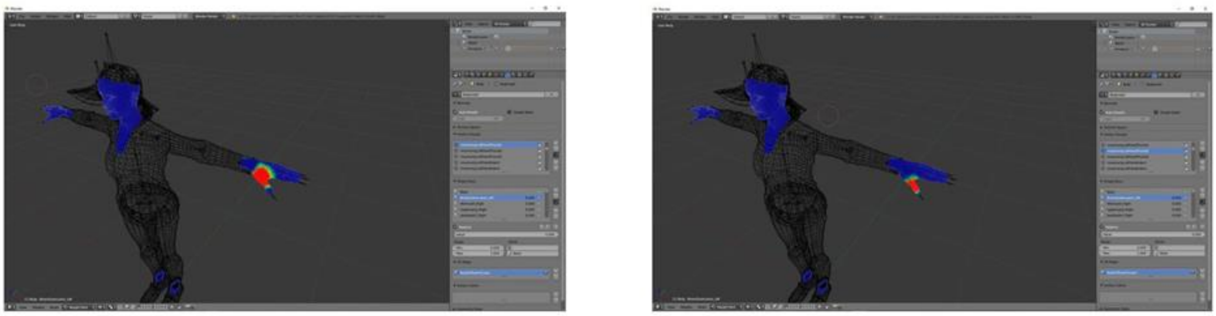


Figure 7-3: Weight painting

The Unity 3D Editor is the development environment where the whole procedure, regarding the humanoid avatar control and visualisation, takes place. One of the main advantages is that Unity provides a great state machine animation system called Mecanim [Mec19]. Furthermore, humanoid animations can be applied to all models with a humanoid (human like) rig[19], just remapping the animations through different characters. Though, specific guidelines must be followed, such as keeping the bone count to the minimum number of fifteen bones. This is a prerequisite for Unity to be able to import correctly the rig and integrate it with its Mecanim system, by assigning an avatar to it. Inside the Unity’s Editor there is the avatar Mapping tab as shown in Figure 7-4.

[19] Rigging is a technique used in skeletal animation for representing a 3D character model using a series of interconnected digital bones.

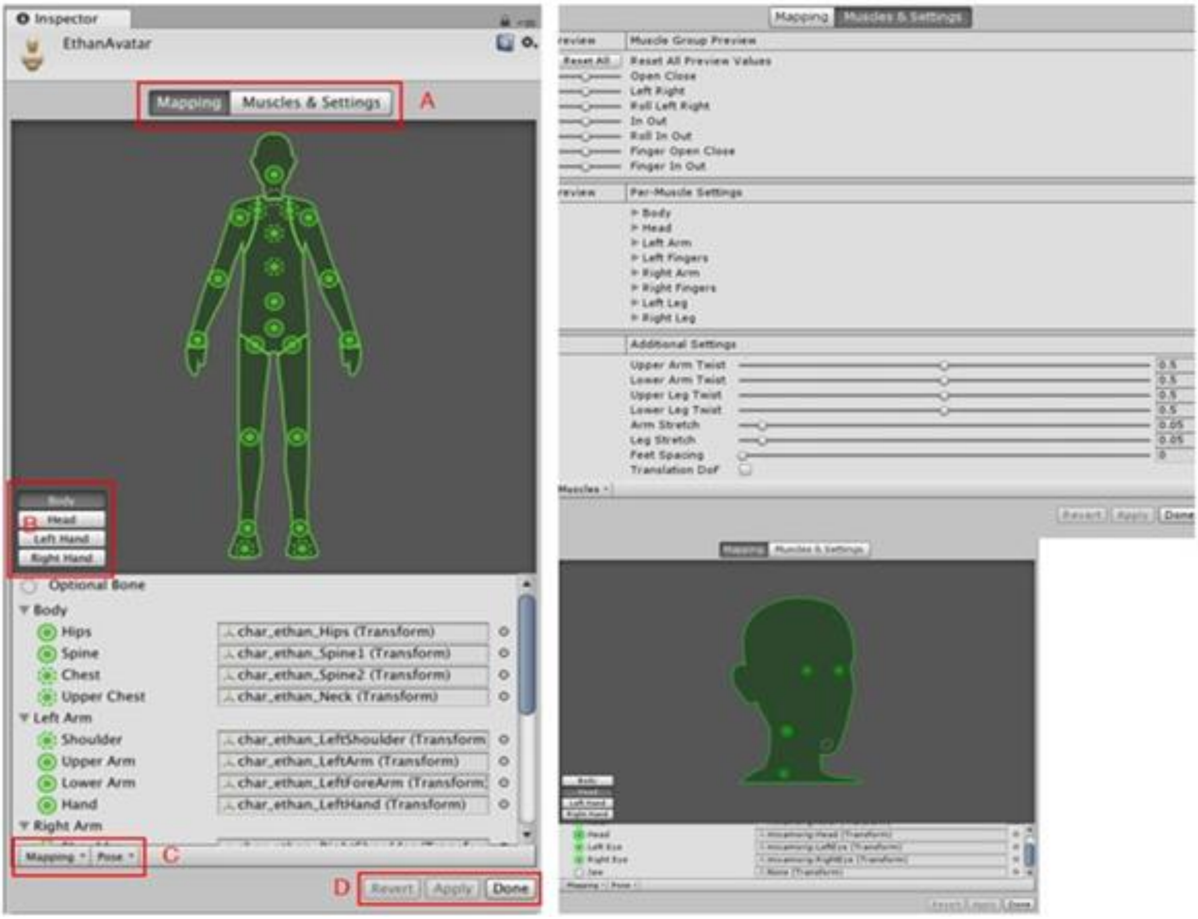


Figure 7-4: Unity's avatar mapping

The avatar mapping indicates which of the bones are required (solid circles) and which are optional (dotted circles). Furthermore, a more detailed image is provided for the head and the limbs, with optional bones. Another important advantage of this system is that the specific mapping can be saved and reused for another similar avatars.

Moreover, Unity provides some muscle settings which directly affect the weight of each bone and are named in a very reasonable way, instead of their anatomic properties. A rig for an impaired character could be adjusted in by tweaking some of these values. Additionally, some basic editing of the rig is provided, which results in direct results inside the scene, such as a character with broader shoulders. Therefore, some restrictions and limits were added to avoid as much as possible out of bounds conditions or abnormal postures (e.g., when the user forces extreme postures with the controllers).

7.3 Example of a restoration tutorial

In the following, the steps to be performed for the reconstruction of the missing portion of an inlay in a wooden object are reported:

First, it is necessary to identify the damaged or missing wooden part. If it were a damaged part it would be necessary to cut it off with a scrapped, and gently move around the damage without destroying any part of the rest of the artefact. Then, a proper wood should be selected for the reconstruction. The edges of the damaged wood are drawn in order to cut out the specific piece with a saw and match the inlays. Some corrections might be needed until the perfect size is reached. Next step would be the reintegration of the missing part by gluing the blocks created in the missing inlay. Finally, a polishing with sellac and pad would be necessary to transform the added piece and match it with the rest of the artefact. Figure 7-5 to Figure 7-9 illustrate the procedure.

Figure 7-5: Identifying the missing part

Figure 7-6: Drawing

Figure 7-7: Position and cut

Figure 7-8: Reintegration of the missing part

Figure 7-9: Final polishing

In the VR environment the user has to accomplish all the above tasks by following the restoration scenario. In order to be a more entertaining experience for the user, rather than an actual professional restoration, the whole procedure has been simplified, although keeping the realism as much as possible. Also, the tourist guide can provide instructions during the whole operation through VoIP. To assist the user a table with the needed tools (scrapper and scroll saw) is presented. Also, an indicator is shown over each tool when it is needed, which makes possible for the more experienced users to make the restoration process by themselves with just minor VoIP directions.

As a first step the user must use the scrapper to cut a predefined piece over the table. When the cut is completed, the piece actually pops up from top and the user is able to grab it. When the broken piece is removed the new one must replace it; therefore, the user must take a bigger square piece of wood and cut it on scroll saw in a similar size as the one being removed. During the cutting process some lines help the user to remain in a desired area (although there is no need to make a precise cut in order to avoid an unpleasant and difficult experience).

When the cut from saw is completed a wooden piece appears in the tourist's hand. Following the tutorial, the piece must have glue on it in order to achieve the reintegration of the missing part. So, before placing the piece to the table, the user must put first some glue on the surface (which can be found in one of the two bottles on the table). After the piece is positioned on its place a further layer of shellac must be added to accomplish the task. Once the whole procedure is completed the 3D artefacts appeared giving the opportunity to the user to examine them with more detail. Figure 7-10 shows the different steps from inside the application.

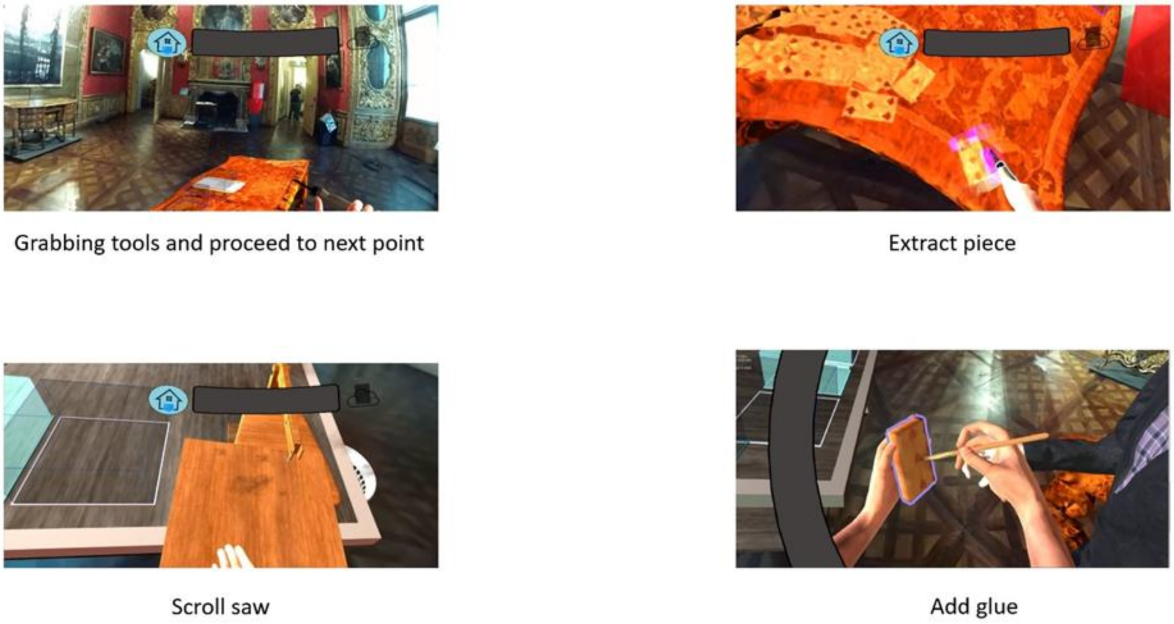


Figure 7-10: Restoration procedure inside the VR application

7.4 KPI definitions and related concepts

In the following, we describe some of the KPIs refereed in Chapter 5 providing also some considerations on the concepts underlying these KPIs:

- **Wireless area capacity**: Quantifies the provided data volume per geographic area (in MB/m^2). This metric can be evaluated under two different traffic models: (i) full buffer model and (ii) non-full buffer model. The first one assumes that the users' buffer is always full, while the second one assumes that users generate data following some traffic model. The traditional full buffer model serves as an upper bound for this KPI.

  5G-MoNArch enablers allow for improved (radio) resource efficiency and hence improve the "effective" wireless area capacity. This captures not only the improvement in the radio PHY but also the upper layer resource allocation mechanisms that improve efficiency in the use of the radio resources.

  Since the wireless area capacity KPIs is only meaningful for deployments at a very large scale (i.e., when the size of the service area is very large), the testbeds only provide limited insights on this KPIs, so this KPIs is more accurately demonstrated by simulations and emulations in the framework of evaluation cases of WP6.
- **Service creation time.** This is defined as the time from the arrival of a request to setup a network slice at the network operator's management system until the slice is fully operational.
- **E2E latency.** Measures the time that it takes to transfer a given piece of information from a source to a destination, measured at the communication interface.
- **Security**. Following traditional security analysis, the security KPI is evaluated qualitatively.
- **Cost efficiency.** This is the most important KPI demonstrating the benefit of network slicing and resource elasticity. To provide this KPI an economic analysis has been done building on the previous work of the 5G-PPP Phase 1 project 5G-NORMA and adapting it to the use cases addressed by 5G-MoNArch. One of the elements considered in this analysis is the savings in terms of computational resources achieved by the resource elasticity concept.

7.5 Security threats and probes

The following tables shows the security threats for the simulated Smart Sea Port testbed assets and the probes to detect identified threats.

Table 7-2: Security threats for the simulated Smart Sea Port testbed assets

Asset	Potential Threats	Attacker	Probability	Impact
Network	Denial of service	Outsider or insider attacker targeting any asset within the network	High, especially with network segments exposed to public (i.e., wireless communications)	High, as many critical services and operations might stop being available
Central Traffic Control Data Centre	Brute force attack	Insider attacker, attempting to log in any service of the data centre	Low for insider attackers assuming network configuration that prevent outsider access attempts	High, as a potential success might give critical permissions, allowing to carry out nasty operations
	Denial of service	Insider attacker trying to reduce service availability. Outsider attacker in case of services exposed to the public network	Low, assuming perimeter security allows to prevent outsider attacks.	High, as many critical services might stop being available
Traffic lights	Brute force attack	Insider or outsider attacker, attempting to get control of traffic lights	High, especially for being exposed to public	High: a successful access would allow to carry out nasty operations, modify configurations or tamper data
	Device tampering	Insider or outsider attack, capable of physically manipulate the traffic lights	High, especially for being exposed to public	High, as it might derive into services not available or manipulated to operate incorrectly
	Man in the middle	Insider or outsider attacks capable of intercepting network traffic to/from the traffic lights and modify data/commands	High, especially for being exposed to the public but low in what respect to the probability to modify the data contained	High, as it might derive into incorrect commands and data entailing incorrect decisions (i.e., changing light from red to green)
VPN server	Unauthorised VPN connection	Mainly insider as it requires capability to access to the VPN server to establish a VPN connection using incorrect configuration (i.e., unsecure)	Low, as it requires low security configuration that allows the establishment of unauthorised connections	High, as it might affect to the rest of the connections established by this server
Streaming server	Denial of service	Insider attacker trying to reduce service availability	Low, assuming perimeter security allows to prevent outsider attacks	High, as it might reduce performance of the service or even stop it
AR-headsets	Anomalous UE behaviour	Insider or outsider, it consists on not common behaviour or headsets when accessing services (for example, accessing	High, especially because many devices might be working at the same time and exposed to	Low, as many verdicts labelled as anomalous might not denote malicious activity.

		to the service at unusual times)	many different types of users	In general verdicts related to this thread denotes just a probability of a malicious activity
	Man in the middle	Insider or outsider attacks capable of intercepting network traffic to/from the headsets, manipulating the content streamed	Low, assuming AR headsets have security mechanisms that encrypt the communication between streaming server and headsets	High. In case the attack is successful, the headset might receive incorrect images, giving the possibility to perform nasty operations due to incorrect information received by the operator.
Database	Denial of service	Insider attacker trying to reduce service availability. Outsider attacker in case of services exposed to the public network	Low, assuming perimeter security allows to prevent outsider attacks	High, as many critical services might stop being available
	Service discovery	Insider or outsider attack that perform port scanning capable of finding services available, the type, or even the version, which might denote vulnerable/exploitable services	High. Most of the attacks start with a service discovery to find out potential targets	Low. The service discovery by itself is harmless. The risk comes when used as preliminary step to carry out more dangerous operations.
Pollution Sensors	Anomalous UE behaviour	Insider or outsider, it consists on not common behaviour or sensors (for example, strange temperature measurements in the case of temperature sensors)	High. In general, the security capabilities of IoT devices such as pollution sensors are very scarce and need to be combined with external security mechanisms	Low, as many verdicts labelled as anomalous might not denote malicious activity. In general verdicts related to this thread denotes just a probability of a malicious activity
	Denial of service	Insider or outsider, trying to reduce availability of the sensors by flooding them with many requests	High, especially for being devices exposed to the public	High, as sensors might stop working
	Device tampering	Insider or outsider attack, capable of physically manipulate the sensors	High, especially for being exposed to public	High, as it might derive into services not available or manipulated to operate incorrectly
Wireless Spectrum	Jammer attacks	Outsider attacker trying to collapse the availability of the wireless spectrum by interfering with radio frequencies	High, especially because pollution sensors are using wireless spectrum to report measurements.	High, as sensors might stop working.

Table 7-3: Identification of security probes to detect identified threats.

Threat	Security probes	Function	Type	Product/project
Denial of service Service Discovery	Network Intrusion Detection System (NIDS)	NIDS capable of sniffing network traffic to detect incidents based on anomalous patterns	HW	Free: Snort, Suricata, Bro IDS, OpenWIPS-ng, Security Onion Commercial: FG-* by netZentry, Sleuth9 from DeepNines, Qradar by Q1Labs, FloodGuard by Reactive Network Solutions, AppScan* by WatchFire, IntruShield by McAfee
Brute force attack	Authentication, Authorisation and Accounting logging (AAA) Pluggable Authentication Modules (PAM)	Host-based Intrusion Detection Systems capable of logging activities in a machine (such as CPU usage, login attempts or memory consumption)	SW	Free: OSSEC, Tripwire, AIDE, Prelude Hybrid IDS. Commercial: HP-UX by HP
Device tampering	Tampering sensor (TAMP)	Detect unauthorised physical manipulation of devices	HW	Several commercial products exist, which depends on the technology used (magnetic, inductive sensors, etc.). Some examples: TE connectivity, Texas instruments, etc.
Man in the middle	Hardware Security Module (HSM)	Hardware device capable of generating, storing and protecting cryptographic keys to protect communication channels	HW	Several commercial solutions exist by BULL, IBM, ARC, Thales, HP to name just a few
Unauthorised VPN connection	VPN Logging (VPN_log)	Not a security probe by itself. It relies on the capability of VPN servers of logging activities (clients connected, security settings, etc)	SW	Free: Open source solutions such as OpenVPN can report logs using remote syslog. Commercial: All the commercial solutions for VPN provide with logging activities that can be used as source of events for security management.
Anomalous UE behaviour	User Equipment Behavioural Analysis (UEBA)	Machine learning based solutions capable of predicting incidents based on anomalous behaviour of devices and the services they use	SW	In 5GMonarch CERTH provides an implementation of this solution. Other solutions are provided by other vendors, including products such as Splunk, RedOwl, HPE Niara, or by Microsoft
Jammer attacks	Jamming detectors (antijamming)	Wireless spectrum analysers that detect different patterns of waveform distortion	HW	Several solutions exist, such as DoSSensing by Workdsensing for Wi-Fi, Raytheon for GPS Anti-Jam

		that provoke DoS over certain frequencies		systems. Antijamming approaches for 5G are still in research stage [ANB+15] [LLC18]

7.6 Networking dashboard

The role of the networking dashboard

A web-based GUI platform has been developed by MBCS to serve as a networking dashboard of the Turin testbed. This is just an auxiliary component developed for demonstration purposes, so it is not part of the NGPaaS architecture itself.

The first target of the GUI is to visualise real-time diagrams of computational and storage metrics from the nodes that form the overall topology of the testbed. In this context, charts in the front end of the GUI visualise the measurements collected for the following KPIs:

- CPU and RAM usage percentages consumed on the machines that host RAN functionalities at the central and the edge cloud. The target is on user plane RAN functions, since they directly affect the processing of users' data and, thus, the users' experience. More precisely, the measurements that refer to the central cloud, include the CPU and RAM consumptions of the PDCP and RLC functions, which are migrated there, based on elasticity provisioning policies.
- E2E latency of the network.

The second and highly important target of the GUI is to illustrate the impact of resource elasticity to the service performance. Recall that elasticity policies are applied to network functions, realising either a re-location of the functions or a re-allocation of computational resources to the nodes that host the functions. These changes are visualised through animations in the GUI front-end. Additionally, actions that are performed by the orchestrator (where the AI policies for enabling the resources elasticity are executed), are depicted as logs or messages in a dedicated field.

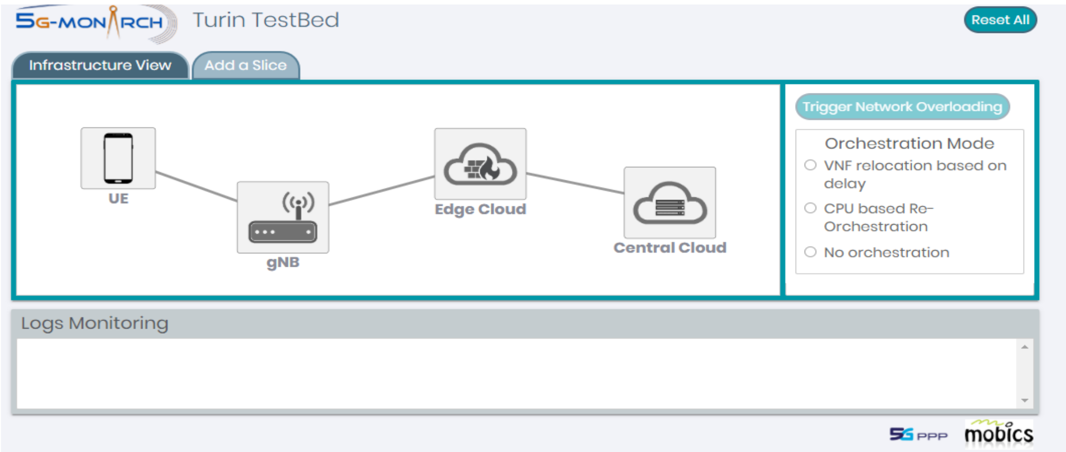


Figure 7-11: GUI initial page

Implementation approach

From the implementation point of view, the GUI platform is built upon the following technologies:

- **NodeJs**: used as the server-side service that feeds the web interface with metrics values that are stored in the database.
- **Javascript**: used as the frontend component of the platform. Along with Hyper Text Mark-up Language (HTML) and Cascading Style Sheets (CSS), it creates the animations of the movement of network functions as well as the conversion of metrics values into plots and graphs that are introduced by the backend (NodeJs) service.
- **MongoDB**: used as the database where live metrics values are stored. The data stored there can be used by the AI algorithms to drive the orchestrator actions.
- **Python**: used to write the scripts which are executed on the nodes to retrieve CPU and RAM consumption values and the E2E network latency.

User manual

The GUI platform at its idle state depicts the testbed infrastructure (*Infrastructure View tab* in Figure 7-11), i.e., it presents an abstract view of the physical architecture consisting of the main components of the E2E service provisioning chain, namely: edge cloud, central cloud, gNB and UE. In order to activate the testbed, the user should select an orchestration mode and press the start/reset button.

The tab *"Add a Slice"* can be selected by the user in order to trigger the set-up of a slice. In line with the purpose of the testbed the user is able to create two slices, namely one eMBB and one URLLC, for the video streaming and the haptic service, respectively (Figure 7-12).

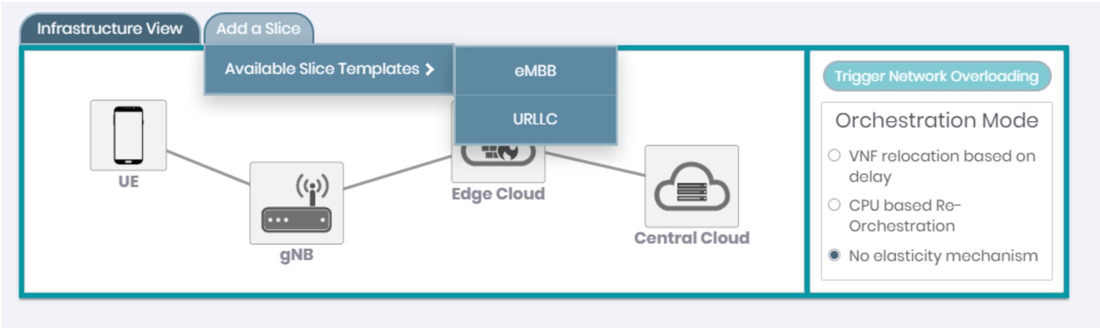


Figure 7-12: Button for activating the monitoring of slices as presented in the GUI

Once a slice is selected, logging messages appear at the bottom of the web page and inside the logging box to reflect the procedures that take place for the slice set up. At the same time signalling between the database that stores the metrics and the backend – fronted services is performed, and plots that represent the metrics being stored in the database are depicted to the frontend as shown in Figure 7-13 and Figure 7-14, for the eMBB and the URLLC slice, respectively. The user can watch the metrics from both slices concurrently by selecting the tab *"Infrastructure View"* while the slices are running (Figure 7-14).

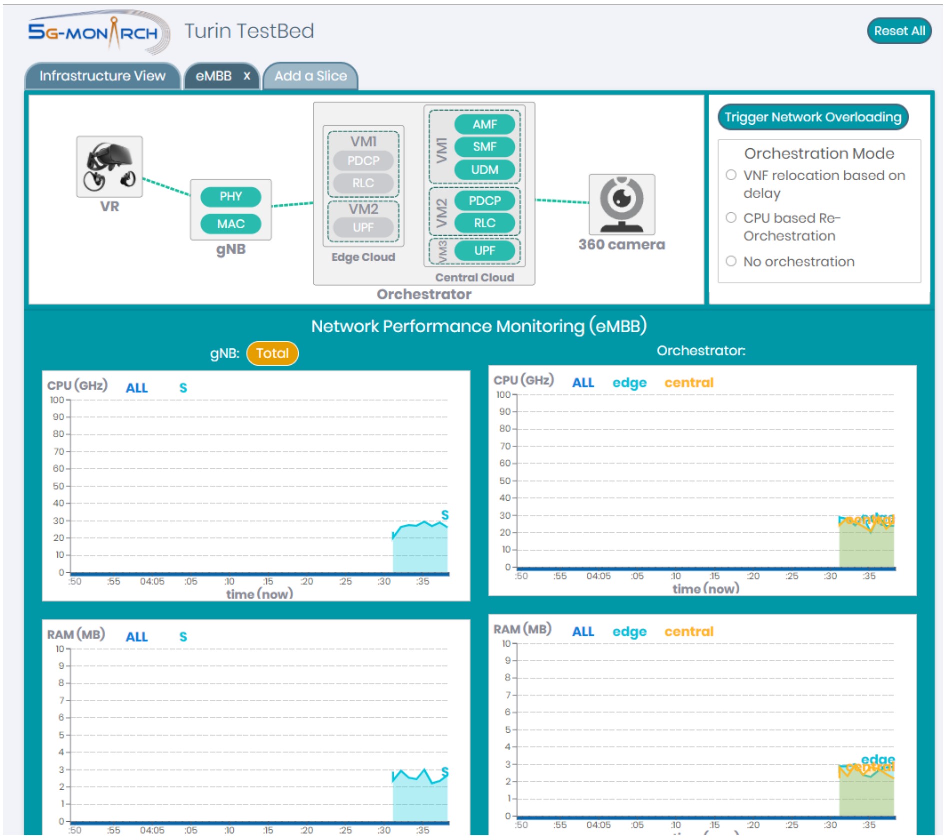


Figure 7-13: The main page with the performance charts of the eMBB slice

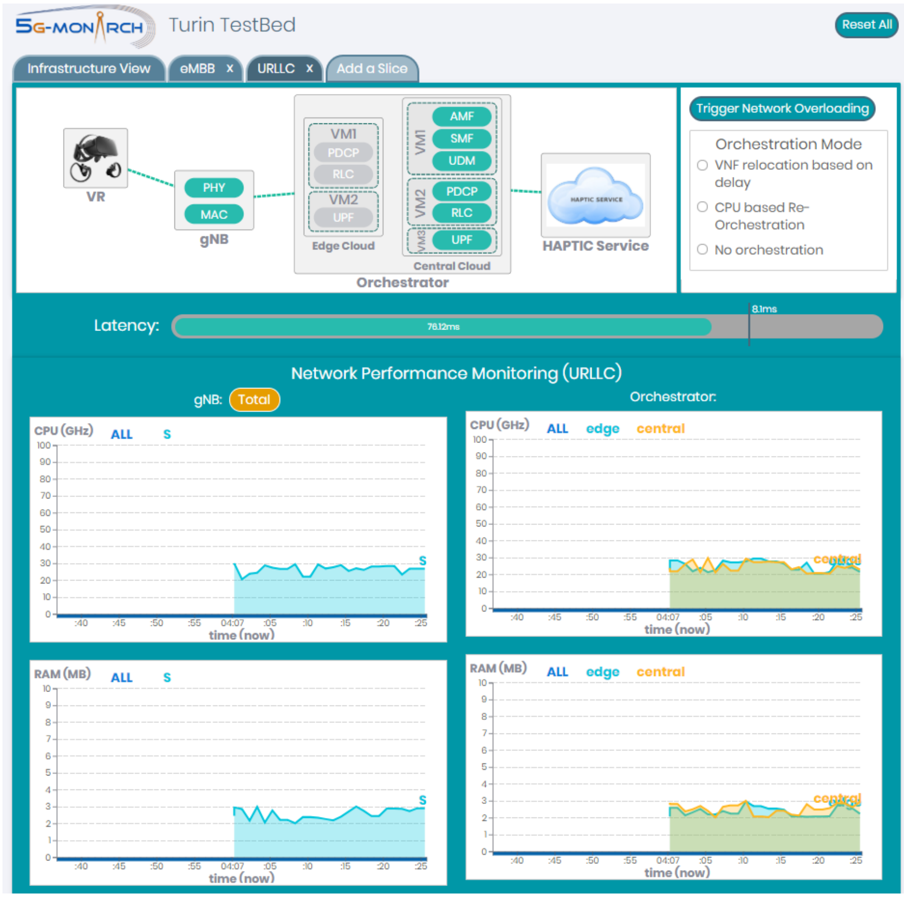


Figure 7-14: Main page with the performance charts of the URLLC slice

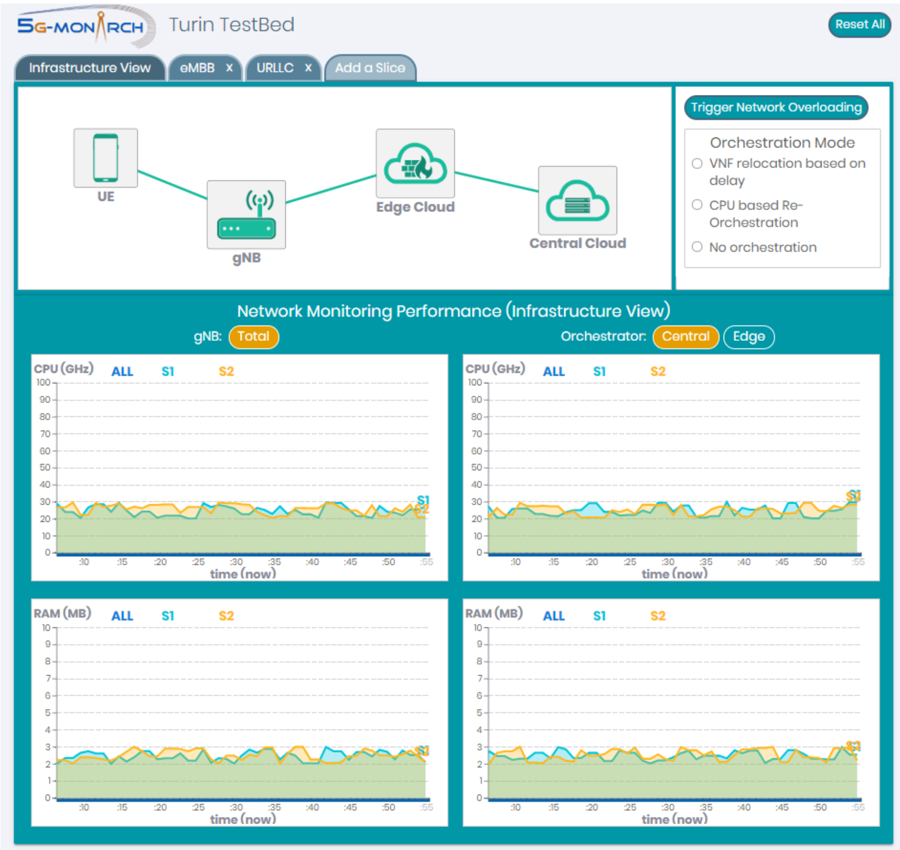


Figure 7-15: Monitoring of the metrics from all the active slices (infrastructure view tab)

Additional traffic can be triggered from the GUI by pressing the *"Trigger Network Overloading"* button (see on the right top of Figure 7-15). This triggers the overloading of the infrastructure (CPU and storage resources), an action that results to the activation of the AI resource elasticity at the orchestrator, and finally, to the migration of the network functions forth or back (to the edge cloud or to the central cloud, respectively). Figure 7-16 shows how the PDCP, RLC and UPF relocation from the central to the edge cloud appears in the GUI.

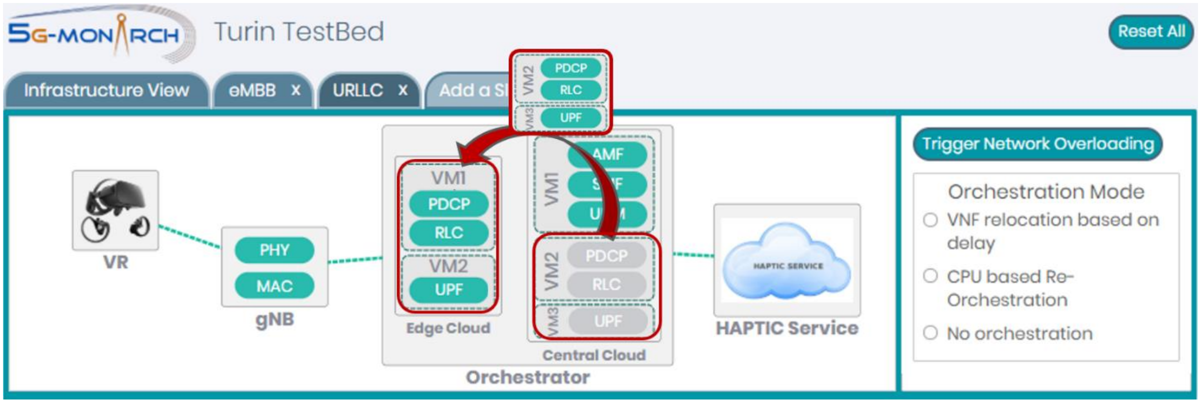


Figure 7-16: PDCP, RLC and UPF relocation in the Touristic City testbed

7.7 Touristic City testbed pictures

In the following we show some pictures taken during the Touristic City testbed workshop which took place in Turin between 21 and 24 May 2019.

Figure 7-17: 360º Camera in the Madama Reale Chamber

Figure 7-18: Installation in the Small Wardrobe (360° video server, VR application server and VR client of the first user)

Figure 7-19: UE installation and VR client of the second user

Figure 7-20: Demonstration room (gNB installation in the foreground)

Figure 7-21: Tourist guide room (May 2019)

Figure 7-22: Workshop presentation at Palazzo Madama (May 2019)

Figure 7-23: Live demo presentation to the workshop attendees (May 2019)

Figure 7-24: Demonstration of the VR application (May 2019)

Figure 7-25: Demonstration of the orchestration functionalities (May 2019)

Figure 7-26: Visitor using the VR application (May 2019)

7.8 Smart Sea Port testbed pictures

In the following some further pictures from the Smart Sea Port testbed installation activities, and from workshops and events are shown which took place in Hamburg in 2018 and 2019.

Figure 7-27: Presentation of the slice lifecycle management GUI (November 2018)

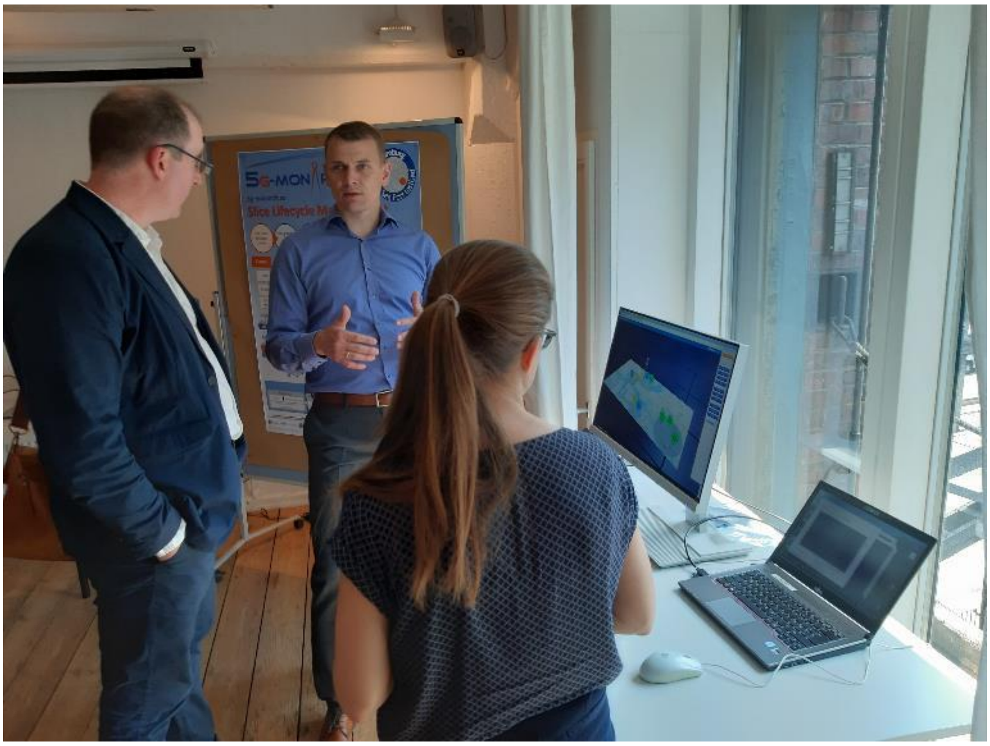


Figure 7-28: Presentation of slice lifecycle management and performance KPIs (June 2019)

Figure 7-29: Hamburg TV Tower where the base stations are installed

Figure 7-30: Mounting of the antennas of the base station

Figure 7-31: Mobile traffic light installation (June 2019)

Figure 7-32: Poster presentation of reliability & security concepts (June 2019)

Figure 7-33: Presentation of the AR application (November 2018)

Figure 7-34: Presentation of the traffic light control (June 2019)

Figure 7-35: Second measurement campaign at location of the traffic light installation

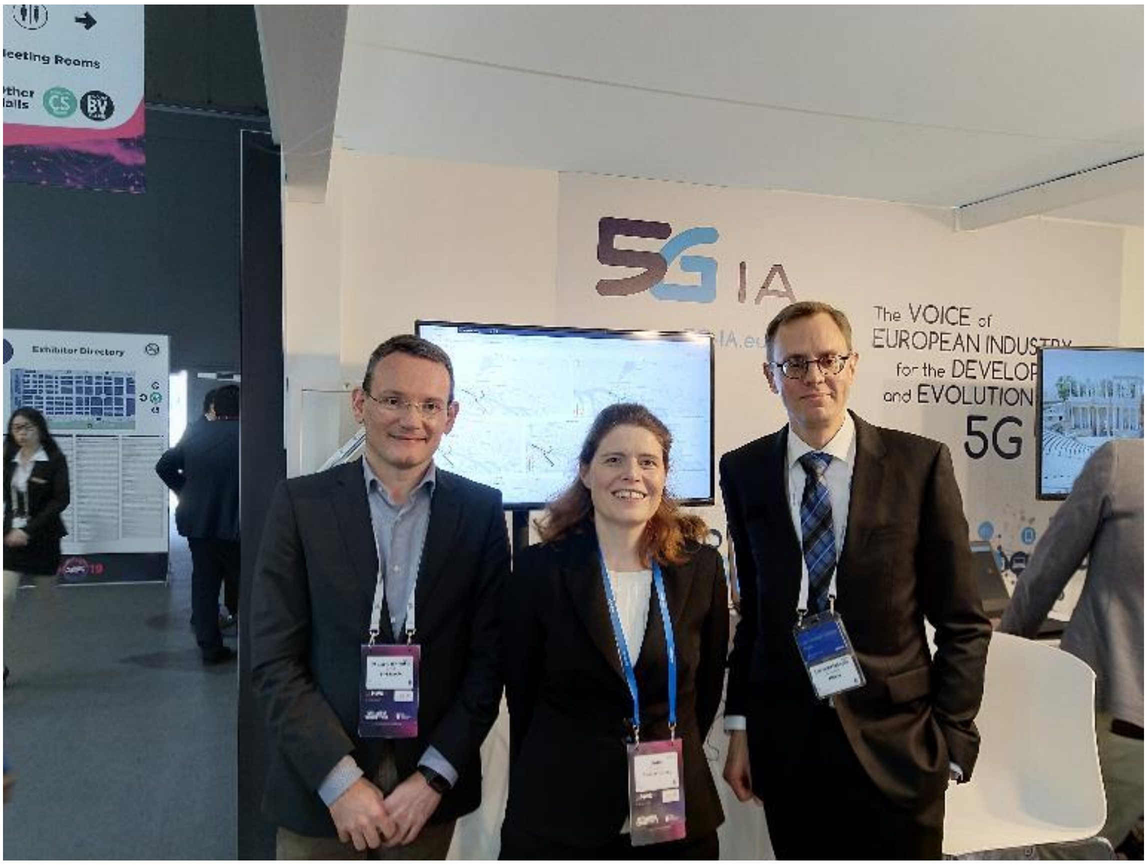


Figure 7-36: Smart Sea Port testbed live demonstration at MWC 2019

8 References

[3GPP18-28530] 3GPP, TS 28530, "Management and orchestration; Concepts, use cases and requirements", Release 15, September 2018

[3GPP18-28801] 3GPP, TR 28.801, "Study on management and orchestration of network slicing for next generation network", Release 15, January 2018

[3GPP18-38211] 3GPP, TS 38.211, "Physical channels and modulation", Release 15, September 2018

[3GPP18-38300] 3GPP, TS 38.211, "NR and NG-RAN Overall Description", Release 15, September 2018

[3GPP19-37340] 3GPP, TS 37.340, "Multi-connectivity; Stage 2", Release 15, v15.4.0, January 2019

[3GPP19-NR_mob] 3GPP Work Item Description, "NR mobility enhancements," RP-190489, Release 16, March 2019

[5GM17-D21] 5G-MoNArch project, Deliverable D2.1, "5G-MoNArch baseline architecture based on 5G-PPP Phase 1 results and gap analysis", November 2017

[5GM17-D61] 5G-MoNArch project, Deliverable D6.1, "Documentation of requirements and KPIs and definition of suitable evaluation criteria", September 2017

[5GM18-D22] 5G-MoNArch project, Deliverable D2.2, "Initial overall architecture and concepts for enabling innovations", July 2018

[5GM18-D62] 5G-MoNArch project, Deliverable D6.2, "Methodology for verification and validation of 5G-MoNArch architectural innovations", June 2018

[5GM18-D51] 5G-MoNArch project, Deliverable D5.1, "Testbed setup and 5G-MoNArch technologies demonstrated", August 2018

[5GM18-D31] 5G-MoNArch project, Deliverable D3.1, "Initial resilience and security analysis", May 2018

[5GM18-D41] 5G-MoNArch project, Deliverable D4.1, "Architecture and mechanisms for resource elasticity provisioning", May 2018

[5GM19-D32] 5G-MoNArch project, Deliverable D3.2, "Final resilience and security report", March 2019

[5GM19-D42] 5G-MoNArch project, Deliverable D4.2, "Final design and evaluation of resource elasticity framework", March 2019

[5GM19-D23] 5G-MoNArch project, Deliverable D2.3, "Final overall architecture", April 2019

[5GM19-D63] 5G-MoNArch project, Deliverable D6.3, "Final report on architectural verification and validation", July 2019

[5GNOR16-D41] 5G-NORMA project, Deliverable D4.1, "RAN architecture components – preliminary concepts", November 2016

[Ama19] Amazon Web Services, Inc. (2019). Amazon Web Services (AWS) - Cloud Computing Services. [online] Available at: https://aws.amazon.com/ [Accessed 11 May 2019]

[ANB+15] Adem Nadia, Bechir Hamdaoui, and Attila Yavuz. "Pseudorandom time-hopping anti-jamming technique for mobile cognitive users." 2015 IEEE Globecom Workshops (GC Wkshps). IEEE, 2015.

[Ble19] Blender, free and open 3D Creation Software (2019) [online] https://www.blender.org [Accessed 14 May 2019]

[BKM08] Bangor A, Kortum PT, Miller JT. An empirical evaluation of the system usability scale. Int J Hum Comput Interact. 2008;24(6):574–94. https://doi.org/10.1080/10447310802205776

[Eni17] ETSI ENI, "Improved operator experience through Experiential Network Intelligence", White Paper, Oct. 2017.

[Eni18] ETSI RGS ENI 005 (GS ENI 001), "Experiential Networked Intelligence (ENI); System Architecture" v0.22, 2018

[Eni19] Eniwiki.etsi.org. (2019). PoC - wiki. [online] Available at: https://eniwiki.etsi.org/index.php?title=PoC#PoC.232:_Elastic_Network_Slice_Management [Accessed 19 May 2019].

[ETSI18-GRENI001] ETSI, GR ENI 001, "ENI use cases", April 2018

[ETSI14-GSNFV001] ETSI, GS NFV-MAN 001, "Network Functions Virtualisation (NFV); Management and Orchestration", December 2014

[ETSI16-MEC] ETSI, "Mobile Edge Computing A key technology towards 5G", 2016

[Ett19] Ettus.com. (2019). [online] Available at: https://www.ettus.com/wp-content/uploads/2019/01/b200-b210_spec_sheet.pdf [Accessed 23 May 2019]

[Ett19b] Ettus Research. (2019). USRP X310 High Performance Software Defined Radio - Ettus Research. [online] Available at: http://www.ettus.com/all-products/x310-kit [Accessed 07 Jun. 2019].

[Eve19] Every Interaction. (2019). 5 UX KPIs You Need To Track - Every Interaction. [online] Available at: https://www.everyinteraction.com/articles/5-ux-kpis-need-track/ [Accessed 27 Jun. 2019].

[Fon19] Fondazionetorinomusei.it. (2019). Homepage | Fondazione Torino Musei. [online] Available at: https://www.fondazionetorinomusei.it/en [Accessed 23 May 2019].

[GGS+18] Ginés García, Marco Gramaglia, Pablo Serrano, Albert Banchs, "POSENS: A Practical Open Source Solution for End-to-End Network Slicing", IEEE Wireless Communications Magazine, October 2018.

[Gra19] Grafana Labs. (2019). Grafana - The open platform for analytics and monitoring. [online] Available at: https://grafana.com [Accessed 18 May 2019].

[GSMA18-5GNS] GSMA, "5G Network Slicing Report: From Vertical Industry Requirements to Network Slice Characteristics", September 18

[GST] GSMA: "Generic Network Slice Template", Version 0.02

[Gue19] GUEANT, V. (2019). iPerf - The TCP, UDP and SCTP network bandwidth measurement tool. [online] Iperf.fr. Available at: https://iperf.fr [Accessed 11 May 2019].

[Ham19] Hamburg, E. (2019). Ein Masterplan Klimaschutz für Hamburg. [online] hamburg.de. Available at: https://www.hamburg.de/bsu/2580042/2010-10-20-bsu-stadt-im-dialog [Accessed 15 Jun. 2019].

[Ham19b] The official website of the Port of Hamburg. (2019). Welcome to the Port of Hamburg. [online] Available at: https://www.hafen-hamburg.de [Accessed 19 May 2019]

[Hpa19]	Hamburg-port-authority.de. (2019). HPA. [online] Available at: https://www.hamburg-port-authority.de/en [Accessed 24 May 2019].
[Inf19]	InfluxData. (2019). InfluxDB: Purpose-Built Open Source Time Series Database | InfluxData. [online] Available at: https://www.influxdata.com [Accessed 7 May 2019].
[Ins19]	Insta360 Pro Camera (2019) [online] Available at: https://www.insta360.com/product/insta360-pro [Accessed 22 May 2019].
[Int19]	Intel.com. (2019). [online] Available at: https://www.intel.com/content/dam/support/us/en/documents/boardsandkits/NUC6i7KYK_TechProdSpec.pdf [Accessed 23 May 2019].
[ISO9244-11-18]	International Organisation for Standardisation, "Ergonomics of human-system interaction -- Part 11: Usability: Definitions and concepts", Edition : 2, March 2018
[Kal19]	Kali.org. (2019). [online] Available at: https://www.kali.org [Accessed 18 May 2019].
[Kub19]	Kubernetes.io. (2019). Production-Grade Container Orchestration. [online] Available at: https://kubernetes.io [Accessed 21 May 2019]
[LLC18]	Lu Xiaozhen, Liang Xiao, and Canhuang Dai. "Uav-aided 5g communications with deep reinforcement learning against jamming." arXiv preprint arXiv:1805.06628 (2018).
[Lin19]	Linux.die.net. (2019). hping3(8) - Linux man page. [online] Available at: https://linux.die.net/man/8/hping3 [Accessed 11 May 2019].
[May19]	Maya 3D Creation Software (2019) [online] https://www.autodesk.com/products/maya/overview. [Accessed 7 May 2019]
[Mec19]	Mecanim is Unity's animation system (2019) [online] Available at: https://docs.unity3d.com/Manual/AnimationOverview.html [Accessed 22 May 2019]
[Mic18]	Microsoft HoloLens (1st generation) device specification [online] Available at https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details#device-specifications [Accessed 9 May 2019]
[Mik18]	MikroTik RouterBoard RB911G-5HPnD technical specifications [online] Available at: https://mikrotik.com/product/RB911G-5HPnD#fndtn-specifications [Accessed 18 Jun. 2019]
[Mik19]	Mikrotik-store.eu. (2019). [online] Available at: https://www.mikrotik-store.eu/media/files_public/1367b823acec2937d144aaea4aaf2270/wAP60GAP-Brochure.pdf [Accessed 7 Jun. 2019].
[Mwc19]	MWC Barcelona 2019. (2019). Hamburg Port Authority, Deutsche Telekom & Nokia for First Large Scale Industrial Commercial 5G Trial - MWC Barcelona 2019. [online] Available at: https://www.mwcbarcelona.com/glomos/hamburg-port-authority-deutsche-telekom-nokia-for-first-large-scale-industrial-commercial-5g-trial [Accessed 15 Jun. 2019].
[Ngi19]	NGINX Server, Open Source (2019). [online] https://www.nginx.com [Accessed 19 May 2019].
[Nma19]	Nmap.org. (2019). Nmap: the Network Mapper - Free Security Scanner. [online] Available at: https://nmap.org [Accessed 14 May 2019].
[Nma19a]	Nmap.org. (2019). Ncrack - High-speed network authentication cracker. [online] Available at: https://nmap.org/ncrack [Accessed 7 May 2019].

[Nok19] Nokia AirScale base station, website [online] Available at https://www.nokia.com/networks/products/airscale-base-station/#overview [Accessed 18 Jun. 2019].

[Oas19] Oasis-open.org. (2019). OASIS | Advancing open standards for the information society. [online] Available at: https://oasis-open.org [Accessed 30 May 2019].

[Osm19] Osm.etsi.org. (2019). OSM. [online] Available at: https://osm.etsi.org [Accessed 30 May 2019].

[Ope19] OpenStack. (2019). Build the future of Open Infrastructure. [online] Available at: https://www.openstack.org [Accessed 30 May 2019].

[Pal19] Palazzomadamatorino.it. (2019). Homepage | Palazzo Madama. [online] Available at: https://www.palazzomadamatorino.it/en [Accessed 8 May 2019].

[PK99] K. S. Park and R. V. Kenyon, "Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment," Proc. IEEE Virtual Reality (VR 99), IEEE Computer Society, Mar. 1999.

[Pro19] Protocol Buffers (protobufs) are Google's language-neutral, platform-neutral, extensible mechanism for serialising structured data. (2019) [online] https://developers.google.com/protocol-buffers/ [Accessed 29 May 2019].

[SMP+14] B. Soret, P. Mogensen, K. I. Pedersen, and M. C. Aguayo-Torres, “Fundamental tradeoffs among reliability, latency and throughput in cellular networks,” in 2014 IEEE Globecom Workshops, GC Wkshps 2014, 2014

[Sur19] Suricata. (2019). Suricata. [online] Available at: https://suricata-ids.org [Accessed 22 May 2019].

[Tur19] Turismo Torino e Provincia. (2019). Turismo Torino e Provincia. [online] Available at: https://www.turismotorino.org/en [Accessed 19 May 2019].

[Uni19] Unity3D, game development platform (2019) [online] https://unity.com [Accessed 29 May 2019].

[Wik19] Wiki.openstack.org. (2019). Heat - OpenStack. [online] Available at: https://wiki.openstack.org/wiki/heat [Accessed 27 May 2019].

[Wrt19] OpenWrt project [online] Available at https://openwrt.org [Accessed June 13th, 2019]