Project: H2020-ICT-2014-2 5G NORMA

Project Name:

5G Novel Radio Multiservice adaptive network Architecture (5G NORMA)

Deliverable D6.2

Demonstrator design, implementation and final results

Date of delivery:	30/11/2017	Version:	1.0
Start date of Project:	01/07/2015	Duration:	30 months

Document properties:

Document Number:	H2020-ICT-2014-2 5G NORMA/D6.2
Document Title:	Demonstrator Design, Implementation and Final Results
Editor(s):	Ignacio Labrador Pavón (Atos)
Authors:	Alessandro Colazzo, Riccardo Ferrari (Azcom), Marco Gramaglia (UC3M), Oliver Holland (KCL), Sina Khatibi, Kunjan Shah, Farid Sheikh (NOMOR), Hassan Osman (RW), Peter Rost (Nokia).
Contractual Date of Delivery:	30/11/2017
Dissemination level:	PU
Status:	Final
Version:	1.0
File Name:	5G NORMA_D6.2.docx

Revision History

Revision	Date	Issued by	Description
1.0	05/12/2017	Atos	Complete version

Abstract

This document is the final deliverable for Work Package 6 (WP6) of the 5G NORMA project (EC project H2020-ICT-2014-2). It describes the design, implementation, achievements and final results of the demonstrators implemented in the context of this project during the Work Package 6 entire lifecycle. As a whole, the document depicts the demonstration platform, the implemented prototypes and the set of experiments carried out on them, putting particular emphasis on the results obtained and their relation with the main 5G NORMA concepts introduced in other Work Packages. The document describes the resources used to get this, and provides conclusions considering the project objectives; it also describes the main problems found and the different strategies to solve them. The document also provides information on the impact that some of the demonstrators presented here have already had on certain relevant events and papers from the beginning of the project so far.

Keywords

Demonstrators, PoC, QoE, QoS, Network Control, Network Slicing, NFV Management and Orchestration (NFV MANO), Security, Economic Analysis, 5G, NFV, SDN, multi-tenant and multi-service networks, software-defined mobile network control (SDM-C), software-defined mobile network orchestration (SDM-O), Service aware QoS/QoE orchestration, virtualised AAA, flexible function (de-)composition and allocation, VNF placement

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

3G/4G/5G	3rd/4th/5th Generation of Mobile Networks
3GPP	3rd Generation Partnership Project
5G NORMA	5G NOvel Radio Multiservice adaptive network Architecture
AAA	Authentication, Authorization and Accounting
AoA	Angle of Arrival
API	Application Programming Interface
AR	Augmented Reality
AuS	Authorization Server (AuS).
BSS	Business Support Systems
CAGR	Compound Annual Growth Rate
CAPEX	CAPital EXpenditures
CN	Core Network
COTS	Commercial On The Shelf
CRAN	Centralised RAN
DL	Downlink
DoA	Direction of Arrival
DRAN	Distributed RAN
E2E	End to End
EC	European Commission
EM	Element Manager
eMBB	Enhanced Mobile Broadband
eNB	Enhanced Node B
EPC	Enhanced Packet Core
ETSI	European Telecommunications Standards Institute
EU	European Union
EuCNC	European Conference on Networks and Communications
FG	Forwarding Graph
GUI	Graphical User Interface
H2020	Horizon 2020
HD	High Definition
HSS	Home Subscriber Server
HTTP	Hyper Text Transfer Protocol
HW	Hardware
IA	Infrastructure Administrator
ICC	International Conference on Communications
ICT	Information and Communication Technologies
IdS	Identity Server

IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IQ	In phase and Quadrature
ITU	The International Telecommunication Union
JSON	JavaScript Object Notation
KCL	King's College London
KPI	Key Performance Indicator
LL	Low Latency
LTE	Long Term Evolution
MAC	Media Access Control
MANO	MANagement & Orchestration
MBB	Mobile Broad Band
MCS	**M**odulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
MNO	Mobile Network Operator
MSP	Mobile Service Provider
MTC	Machine-type communications
mMTC	Massive MTC
MWC	Mobile World Congress
NAS	Non-access stratum
NF	Network Function
NFV	Network Functions Virtualisation
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NS	Network Service
OAI	OpenAirInterface
OPEX	Operating Expenses
OSM	Open Source MANO
OSS	Operations Support System
PC	Personal Computer
PDN	Packet Data Network
P-GW	Packet Data Network Gateway
PHY	Physical
PNF	Physical Network Function
PoC	Proof of Concept
PRB	Physical Resource Blocks
QCI	QoS Class Identifier

QoE	Quality of Experience
QoS	Quality of Service
QR	Quick Response
RAM	Random Access Memory
RAN	Radio Access Network
REST	Representational State Transfer
RB	Resource Blocks
RL	Reduced Latency
RLC	Radio Link Control
RNTI	Radio Network Temporary Identifier
RRC	Radio Resources Control
RRM	Radio Resources Manager
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
SBI	South-bound Interface
SC	Scheduler Controller
SDK	Software Development Kit
SDM-C	Software-Defined Mobile network Control(-ler)
SDM-O	Software-Defined Mobile network Orchestrator
SDM-X	Software-Defined Mobile network Coordinator
SDN	Software Defined Networking
SDR	Software Defined Radio
S-GW	Serving Gateway
SoC	System on a Chip
SP-GW	Serving and PDN Gateway
SINR	Signal-to-Interference-plus-Noise Ratio
SW	Software
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TTI	Transmission Time Interval
TUN	Network Tunnel Interface
UC3M	University “Carlos III” of Madrid
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
URLLC	Ultra-Reliable Low Latency Communication
uMTC	ultra-reliable MTC

USRP	Universal Software Radio Peripheral
V2I	Vehicle-to-Infrastructure
V-AAA	Virtual AAA
VIM	Virtualized Infrastructure Manager
VM	Virtual Machine
VNF	Virtualized Network Function
VNFM	VNF Manager
VR	Virtual Reality
WCQI	Wideband Channel Quality Indicator
WP	Work Package

1 Introduction

1.1 Objective of the document

The main objective of Work Package (WP) 6 is to show the impact of the concepts proposed in other WPs in form of experimental implementations that we usually refer as "Proof of Concepts" (PoC), "Demonstrators" or just "Demos". The main objective of this document is of course to describe the different PoCs implemented in the context of the 5G NORMA project; for each demonstrator detailed information about the concepts intended to prove is given, as well as about their design, implementation (including hardware and software resources) and the experiments carried out on them.

Also, since this is the final deliverable of the WP6, particular emphasis is put on the results and the lessons learned from those demonstrators. An evaluation of the progress and the final results against the project objectives and milestones is provided, as well as about the problems identified and the different strategies used to solve them.

Finally, the document also provides information about the different dissemination activities in the context of WP6, giving details about the relevant events in which the demonstrators (or at least, intermediate versions of them) were already presented.

The document covers the work performed during the complete time span of WP6 in the 5G NORMA project, i.e., from Month 7 (Jan'16) to Month 29 (Nov'17).

1.2 Structure of the document

The following section of this document are structured as follows:

In Section 2 we describe the relationship between the different demonstrators developed in this WP6 and their relationship with three selected 5G NORMA key topics, i.e., network slicing, security and the 5G NORMA economic feasibility. This section also explains how different proposals from other WPs were implemented, e.g., SDM-C, SDM-O, NFV MANO or certain security concepts among others, as well as the different services considered, i.e., Mobile Broadband (MBB), Massive Machine Type Communication (mMTC) and Reduced Latency (RL) services.

Section 3 is focused on the demonstrators themselves. Here, we describe the different demonstrators implemented to verify the concepts introduced in the previous Section 2. It provides a detailed description for each demonstrator, including the hardware (HW) and software (SW) resources selected for their implementation and their relationship with the ideas proposed in other WPs.

Section 4 is about Key Performance Indicators (KPIs) verification. This is a very important section in this deliverable since it analyses the results from the different demos described in the previous Section 3. The section recaps on the main KPIs defined for each demo and the degree of compliance for each one is discussed. Also in this section, we summarize the main lessons learned during the development of this work. Here, the distilled experiences and learnings derived from the practical implementation of the different demos are described to provide relevant information for future projects.

In Section 5, we describe the impact that some of our activities in WP6 have had in certain conferences and relevant events. In this regard, all the activities from the beginning of WP6 works until the end of the project are considered.

Finally, Section 6 completes the document with the main conclusions considering the project objectives and milestones; it also describes the main issues found and the different strategies to solve them.

2 Key Concepts and Implemented Services

In this section, we describe the three main 5G NORMA concepts defined in other WPs that have been selected to be implemented in practice by means of different demonstrators, and also, the services selected for the demonstrator implementations. The key concepts are:

- Network Slicing, covering both: Network Functions Virtualization (NFV) Management and Orchestration, and Network Control;
- Security; and
- The 5G NORMA Economic Feasibility.

In the following subsection 2.1, we describe these concepts in detail; later, in subsection 2.2, the different services are explained.

2.1 Concepts

2.1.1 Network Slicing

The whole 5G NORMA architecture builds around the Network Slicing concept, therefore the project demonstrators are a natural way of showcasing the 5G NORMA innovations in the field of network slicing. As described in [5GN-D33], the different controllers were designed to specifically tackle problems related to:

i) Resource assignment among different network slices, and
ii) Control of these resources while fulfilling the associated KPIs.

In the following, we briefly recall the innovation we introduced for Network Orchestration and Network Control to particularly support the Network Slicing paradigm.

2.1.1.1 NFV Management and Orchestration

The notion of orchestration of Virtual Network Functions (VNFs) is intrinsically linked to the concept of virtualization itself; due to the decoupling between the underlying physical infrastructure and the virtual functions which are implemented on it, the necessity of implementing the functionality to instantiate and coordinate these virtual functions with the existing physical resources emerges naturally.

This decoupling between physical and virtual resources exposes a new set of entities, the VNFs, and makes necessary to consider how they should be dimensioned, configured and connected among them, as well as to the underlying physical infrastructure. Also, considering that individual VNFs could be interconnected to configure specific Network Services (NS), it should be also considered how those network services should be managed as a whole.

So, the concept of NFV Management and Orchestration (MANO) focuses here mainly on the following aspects:

- Management and Orchestration aspects of Virtualised Network Functions; and
- Management and Orchestration aspects of Network Services.

In 5G NORMA, this NFV Management and Orchestration topic has been extensively addressed in the context of WP3 (Multi-service Network Architecture) and WP5 (Flexible Connectivity and QoE/QoS Management). Specifically, in WP3 a dedicated MANO layer has been specially defined for this purpose [5GN-D33], while in WP5 the way this MANO layer can be used for QoS/QoE Management has been described [5GN-D52]. In WP6, the individual demonstrators focus on the different concepts, which will be described later in this document.

Specifically, the framework to make possible the management and orchestration of different network slices has been defined in the WP3, describing also the inter and intra-slice orchestration mechanisms. Figure 2-1 below shows the latest approach of this Management & Orchestration layer.

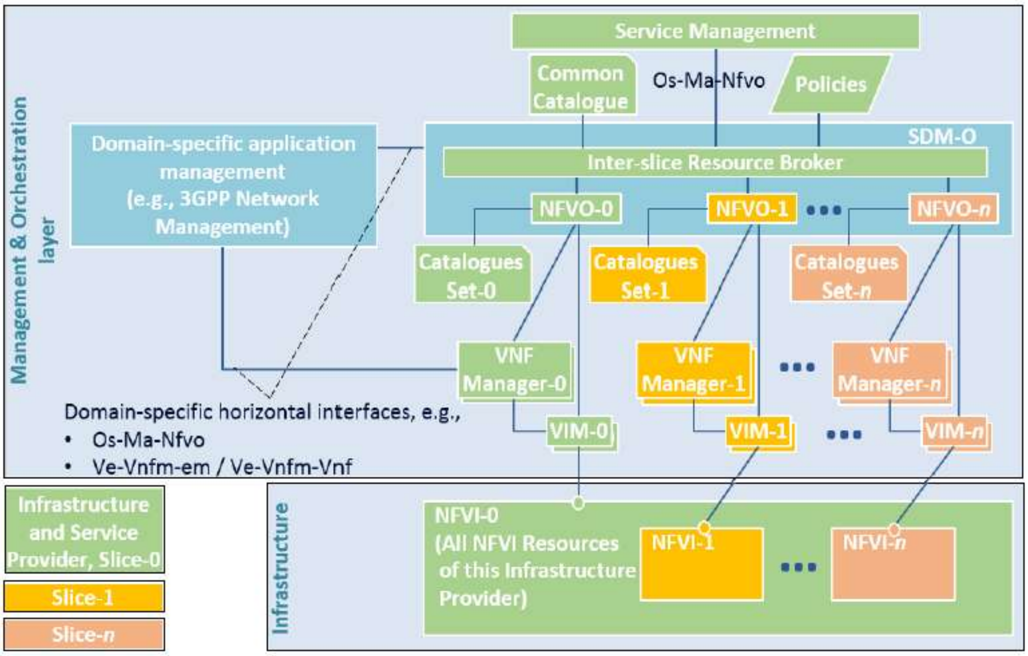


Figure 2-1: 5G NORMA Management & Orchestration Architecture

We are not going to describe in this document the fine-grained details of this architecture (see previous references for that). Important for WP6 is the fact that the architecture defines specific and isolated ETSI NFV MANO [ENM-14] compliant stacks for each slice (orange and rose blocks in Figure 2-1), and also, another specific ETSI NFV MANO stack for the Infrastructure Provider (equivalent green blocks); i.e., the general idea is to deploy specific VNF Manager (VNFM), Virtualized Infrastructure Manager (VIM) and NFV Orchestrator (NFVO) for each slice as they are defined in the ETSI NFV MANO framework. This is aligned with our original approach described in Deliverable D6.1, [5GN-D61], which is based on using the ETSI NFV MANO architecture for our demonstrations.

Although, as described before, the management and orchestration functionalities could cover a wide variety of very specific functions (VNF instantiation, life-cycle management, scaling, etc.), we have decided to focus on three of them that we consider of particular interest for the telco operators in the future 5G networks:

a) Automatic Network Services deployment, driven by network service descriptors where the operational behaviour requirements of each Network Service are captured in a deployment template.
b) Dynamic updating of the VNFs forwarding graph depending on Quality of Service (QoS) and Quality of Experience (QoE) measurements.
c) Placement of VNFs in specific compute nodes, to simulate the placement of NFs in the edge/core networks depending on the services requirements.

So, implementing the concepts defined in WP3 and WP5, the demonstrators show how different slices, associated to different network services, could be deployed and automatically managed triggered by QoS/QoE control events.

2.1.1.2 Network Control

By employing the Enhanced Network Control paradigm as described in [5GN-D33] and [5GN-D52], we allow the tenant to flexibly tune the configuration parameters of the Network Functions that build its network slice. Beside all the advantages in terms of customizability and possible joint access-core optimization that are behind the softwarisation approach proposed by

5G- NORMA and are thoroughly described in the previous deliverables, the enhanced controller paradigm is of particular importance for network slicing especially when used in conjunction with an enhanced orchestration architecture as described in the previous section.

Within 5G NORMA, we especially designed a controller that deals with multiple slices that share the same Network Function (SDM-X); the reader is referred to [5GN-D42] and [5GN-D52] for more details about how this controller can enforce specific KPIs levels for different slices sharing the same infrastructure. Within WP6, we focused on the showcase of a proof of concept controller implementation within a single network slice. That is, in our demos we provide an initial software implementation of the SDM-C, with different extent and different targets, providing a thorough implementation of the SDM-C concept. In particular, the following elements are emphasized:

- The Scheduler SDM-C application, which applies the Software Defined Networks (SDN) principles to a Radio Access Network (RAN) function such as the scheduler. More specifically, the SDM-C application developed in our demos can control a set of eNodeBs (eNBs) to reduce interference and improve the system efficiency in a centralized way.
- The Forwarding Graph (FG) implementation, which is one of the functionalities needed by the enhanced network controller defined in 5G NORMA. When amending the VNF Forwarding Graph, there is the need of reconfiguring also the underlying transport network. This is part of the holistic reach of the SDM-C that aims to provide unified centralized control functionality.

As we will see later, in WP6, we have different demos to address these concepts. One of them is particularly focused on showcasing the Network Control itself, while another one will focus on the interactions between the network controllers that implement the forwarding graphs and the management and orchestration framework, especially when considering the re-orchestration requests.

2.1.2 Economic Feasibility

Certainly, profitability is the ultimate test of whether an operator would invest and deploy a new network or continue with the existing one. In 5G NORMA, and specifically in the context of WP2, a set of evaluation cases to help to evaluate and forecast the 5G NORMA economic feasibility has been defined. This section briefly describes these evaluation cases (for further details see [5GN-D23]). Basically, a set of cost and revenue based metrics has been chosen, which enables to carry out this economic validation. The costs are analysed over a period of 11 years (from 2020 to 2030), capturing the fact that operators will need a suitable period of time for a fair analysis of the investment made. In the following, we briefly describe the different evaluation cases addressed in this work.

- **Evaluation case 1: Cloud RAN**
  This evaluation case is mainly used to compare the costs of a flexible Centralised Radio Access Network (CRAN) versus a Distributed RAN (DRAN).

- **Evaluation case 2: Multi-tenancy**
  This evaluation case analyses the multi-tenant option, i.e., it focuses on whether the multi-tenancy innovation in 5G NORMA brings an economic benefit. The idea is to try to measure the difference in cost between delivering Enhanced Mobile Broadband (eMBB) services over a multi-tenant 5G NORMA architecture compared to delivering the same 5G services over several single tenant networks.

  Since the single-tenant 5G network will have to be consistent with the 5G NORMA architecture (flexible CRAN), we first model a single tenant network for providing enhanced mobile broadband services, and then we model an equivalent multi-tenant network.

- **Evaluation case 3: Multi-service**
  This evaluation case tries to analyse the multi-service concept benefits, in particular the comparison of single and multi-service networks covering MBB, mMTC and Critical Machine Type Communication (uMTC) is shown, of which V2I will be used as an example. In this case, both cost and revenue benefits are evaluated.

2.1.3 Security

Security is a vital consideration in the context of 5G NORMA and the virtualisation of mobile network elements and associated network slices. Security, however, is in conflict with some of the key aspects that are promised in the context of 5G, such as low latency, distribution of management domains, and the slicing concept itself. Considering latency for example, security requires trusted elements to make decisions on aspects such as Authentication, Authorization and Accounting (AAA), whereby those elements are usually therefore limited and centralised within given physical or logical domains. This implies a considerable latency in accessing those elements to, for example, initiate a session. It also affects the distribution of management domains, impeding the slicing concept, which by definition requires management and security to take place in local domains. Indeed, to address latency, in many cases slicing requires management and security operations to take place in *physically* local domains.

The security concept in 5G NORMA is about bringing key aspects, such as AAA, into local management domains to facilitate the realisation of network slicing, localised management, and latency reduction for pioneering 5G Ultra-Reliable Low Latency Communication (URLLC) services. Through this concept, AAA, in the guise of the novel virtualised AAA (V-AAA) proposal, can be distributed to operate both within the scope of localised networks, e.g., reducing latency, and in network slices as largely self-contained management domains. In both cases, the need for the localised AAA to reference the centralised AAA of the mobile network operator (i.e., operator of the computational equipment, bare metal and other elements used for the virtualisation/network slicing concepts) should be reduced as much as possible, while not jeopardising security.

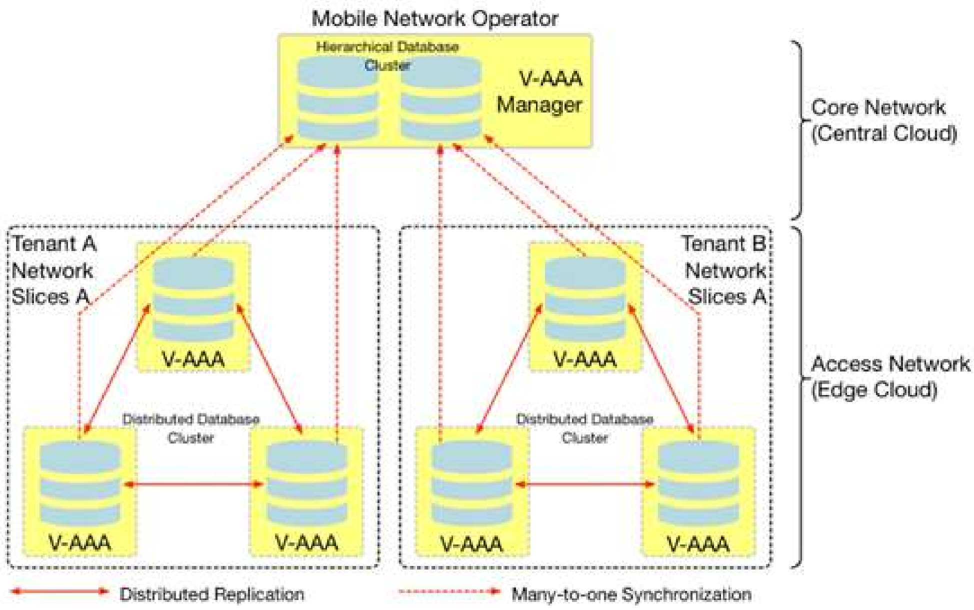


Figure 2-2: 5G AAA hierarchical and distributed illustration

Figure 2-2 illustrates how this concept is seen within the context of 5G NORMA, as a hierarchy of AAA elements with domains existing with the scopes of separate, isolated tenants. In this context, the information arising from the AAA in the separate domains is forwarded to the operator, for monitoring. For instance, if a malicious user within one of the tenants threatens the overall infrastructure of the operator, e.g., through illegal actions or hacking operations, the operator would be able to monitor that and instruct the tenant to appropriately deal with that user.

It is noted that Demo 3, which focuses on Security, aligns well with other WPs, intending to show some key concepts therein. It aligns with use cases in WP2, whereby the concept and its realization (e.g., the tools involved) developed in Demo 3 can be applied to those uses cases. Demo 3 also aligns with WP3 V-AAA design, and indeed is heavily based on that design and has interacted with its development throughout the project. It aims to show the hierarchical and distributed architecture of V-AAA and unified data management (UDM) to maintain central governance in the core network (central cloud), while achieving tenant data isolation and tenant network isolation at the edge.

Demo 3 also aligns with WP5 in protecting and preventing the network slices (e.g., IoT network slice, vehicle network slice, etc.) and network entities (e.g., VNF, PNF, VIM, EM, SDM-C, SDM-X or SDM-O) being compromised during the network resources provisioning and deployment taking place at MNOs or tenants. The protection and prevention technique has already been presented in a 5G NORMA publication [CAFG-17].

2.2 Services

The International Telecommunication Union (ITU) classifies the 5G mobile network services into three baseline categories: Enhanced Mobile Broadband, Ultra-reliable and Low-latency Communications, and Massive Machine Type Communications [ITU-REC]. Traditional Mobile Broadband addresses the human-centric use cases for accessing multi-media content, services and data. This “enhanced” scenario considers new application areas extending the already existing MBB applications by improving performance and increasing users experience and focuses on services having high requirements for bandwidth, such as High Definition (HD) videos, Virtual Reality (VR), or Augmented Reality (AR) applications. Apart from that, URLLC has stringent requirements for capabilities such as throughput, latency and availability; some examples include wireless control of industrial manufacturing or production processes, assisted and automated driving, and remote management among many others. Finally, mMTC is characterized by a very large number of connected devices typically transmitting a relatively low volume of non-delay sensitive data; examples of this are the so called “smart city” or “smart agriculture” applications.

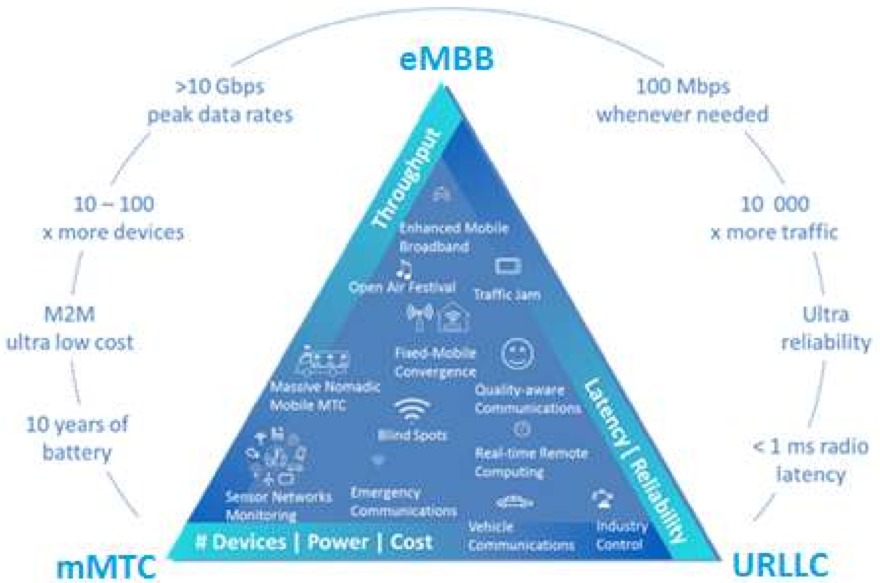


Figure 2-3: ITU 5G Services Classification

However, this classification is not just strictly ternary. Future 5G systems are envisaged to cover a large number of different possibilities of which these three main categories are just the most representative example. This is illustrated in the previous Figure 2-3 where eMBB, mMTC and URLLC are just the vertices in a triangle, but other intermediate cases with in-between features are also considered.

In order to verify the key concepts introduced in the previous Section 2.1, WP6 selected a set of four demonstrators implementing various services that could be represented by different spots in that triangle. They are the following:

- **Demo 1: Native Multi-Service Architecture.** This demo focuses on the Network Slicing concept, considering the network control capabilities of the 5G NORMA architecture, such as the Software-Defined Mobile Network Control and the multi-service and context-aware adaptation of network functions which are defined in WPs 3, 4, and 5. This shows a complete set of services, providing a pure eMBB service (near the top corner in the triangle), a Low Latency service (near the bottom right) and also an mMTC example (bottom left). This demo has been developed jointly by AZCOM and NOMOR.

- **Demo 2: Multi-Slice Service Aware Orchestration.** This demo also focuses on the Network Slicing concept, although from the NFV's Management and Orchestration perspective. It addresses some of the main concepts defined in the WPs 3, 4, and 5, such as the notion of adaptive (de)composition of network functions with Software Defined Mobile Network Orchestration (SDM-O) and the novel concepts of network sharing and network slicing. It is designed to cover two services: an eMBB service (top corner in the triangle) and a Reduced Latency service (near the bottom right). This demo is a joint effort of the University "Carlos-III" of Madrid (UC3M) and Atos.

- **Demo 3: Secured Multi-Tenant Virtual Network Resources Provisioning via V-AAA.** This demo relies on the AAA aspects of the architecture as they are defined in WP3. The main concept covered by this demo is of course the Security concept as it is defined in the 5G NORMA architecture, so this is a backbone common feature that all the services in the triangle should implement. The demo has been developed by the King's College of London (KCL).

- **Demo 4: Online Interactive 5G NORMA Business Cases Evaluation Tool.** This demonstrator is one of the outcomes of WP2. This differs from the previous demonstrators since it is not hardware-based or demonstrating functionality. Instead, this is a demonstration and evaluation tool developed by Real Wireless (RW) that can be used to enable the assessment and dissemination of the economic impact of the 5G NORMA innovations. The three main services in the triangle (eMBB, URLLC and mMTC) are analysed in this demo.

As we see, these four demos are covering a wide spectrum of the different possibilities considered for the future 5G networks. Although these four demos will be described in detail in Section 3 below, a high-level description is provided in the following subsections.

2.2.1 eMBB

Enhanced Mobile Broadband services are specifically addressed by three different demos: Demo 1, 2 and 4. Although the service is the same, the aim of these demos is complementary, since they show different aspects of the 5G NORMA architecture. In the following paragraphs, we explain how this eMBB use case is approached in each of these demos.

Demo 1

This demo shows a high data rate service delivering 4K HD streaming videos. The purpose here is to highlight the role of the Software-Defined Mobile network Controller (SDM-C) component defined in the WP3, showing how the network can be controlled by this key component. It is composed by two parts: Hardware and Software, as it is described in the following sections, having each part its own use case.

In the HW part, the demo is focused on the ability of the SDM-C to react and adapt the network to the required services and to guarantee QoS. In a slice already set up, a user requests a 4K HD video, which requires high throughput. To satisfy the requirements of the service, the SDM-C triggers consequently the relocation of a specific network function, the Serving and Packet Data Network Gateway (SP-GW) component, from the Edge Cloud to the Central Cloud. After this, inter-cell interference is generated to induce a QoS reduction. To remedy this, the SDM-C reacts again providing a new scheduling policy to restore a good QoS level.

Regarding the SW part, the demo focuses on how the SDM-C works in a 5G scenario considering hundreds of users connected to different eNBs. In this use case, the users are connected to different eNBs, but only some of them are requiring 4K video. To meet the QoS required by the eMBB users the SDM-C component, triggers a load balancing of the network to concentrate the users with similar requirements on the same group of eNBs, providing them the most adequate network configuration to satisfy their requirements.

Demo 2

In short, this PoC demonstrates how the NFV orchestration capabilities can be used to include contextual information in real-time video streams to improve the user's experience. For this, a specific eMBB slice has been designed to provide a service consisting of enriching a video streaming signal with context-based add-ons (e.g., subtitles or other graphical elements) depending on the user preferences (colour, size, language…), conditions (e.g., to help people with a hearing impairment) or environmental conditions (e.g., ambient noise). These profiles or environmental conditions can be understood here as QoS/QoE influence factors: the final user could generate a trigger to explicitly request the service according to its preferences, or also, certain QoS measurements could automatically activate the service without an explicit user request. In any case, the idea would be to activate the service by means of the ETSI-based NFVO component which dynamically could add the necessary VNFs to the VNFs forwarding graph while the video signal is being provided.

Demo 4

The eMBB services are addressed in this demo in the three Evaluation Cases previously described in Section 2.1.2.

Regarding Evaluation Case 1 (CRAN vs. DRAN cost comparison), the demo shows that the cost of the network is mainly driven by eMBB traffic demand, which impacts on the design of the network itself, i.e., the dimensioning of the network in terms of macro-cells, small cells and edge cloud sites (applicable to CRAN only) is driven by the eMBB demand.

On the other hand, in Evaluation Case 1, the eMBB traffic growth is also an input to the demonstrator. In this case, the demo shows the cost comparison of CRAN vs. DRAN for several values of traffic growth (Compound Annual Growth Rate values from 20% to 40%). This way, the effect of the eMBB traffic on the network cost is analysed. In addition, the demo also shows detailed cost elements of Capital Expenditures (CAPEX) and Operational Expenses (OPEX) and the effect of the eMBB traffic growth on these cost elements.

Regarding Evaluation Case 2 (multi-tenancy cost benefits), the cost of the network is driven by the eMBB traffic demand. In this case, the benefit of multi-tenancy is analysed for several traffic growth inputs.

Finally, in Evaluation Case 3 (multi-service), we see how the cost and revenue of the network are driven by the eMBB traffic, although it is considered in addition to other traffic types such as utility and V2I traffic.

2.2.2 URLLC

Although the demos implemented in WP6 are not pure URLLC services, we provide two demos (Demos 1 & 2) where a reduced latency is important, so we could place them close to the bottom right corner in the ITU service requirements triangle. Also, Demo 4 analyses the economic impact of URLLC services. In the following paragraphs, a high-level introduction is provided for each case.

Demo 1

In this case, Demo 1 shows an interactive live streaming use case. Of course, the most critical parameter here is latency, which must be kept low and stable. To achieve this, the network slice should be orchestrated in such a way that the core network functions are located close to the end user, i.e., into the edge network cloud. Like in the previous case, Demo 1 is split into two parts (HW and SW) which are described in the following.

In the HW part, the main focus is also the WP3 SDM-C control component. A real-time video streaming is produced using a webcam, so the user can see their own movement replicated on a Personal Computer (PC) screen in real time, which requires low latency. Also, at the same time, an MBB service is executed to show the video streaming. The intention with this demonstration is to show how the SDM-C can recognise two different service types: Reduced Latency and eMBB services, and trigger the relocation of only those Network Functions (NFs) associated to the eMBB service, with no impact on the remaining NFs in charge of the RL service (to avoid degradation in the RL side). Also, when an interference signal is added, we can see how the SDM-C can manage the scheduling policy keeping the QoS requirements in both services. The novelty here is the coexistence of these two different services, and the different management done by the SDM-C component for each one.

On the other hand, the software part shows how the SDM-C can meet the QoS required by the reduced latency users triggering a relocation in the network to concentrate those users with similar requirements on the same base station group. This is performed by processing the feedback messages received in the SDM-C from the connected base stations.

Demo 2

This demo shows how the NFV orchestration capabilities could be used to implement a mobile service to get real-time measurements in wide area industrial environments. In short, this is also a Reduced Latency use case consisting of an Android based mobile application which has been developed to perform (aided) image recognition using Quick Response (QR) codes. In a real environment, those codes could be distributed in the industrial area close to the equipment where measurements of interest could be obtained (e.g., pipes flow or pressure, electric measurements, tank levels, etc.). On each QR code decoding, the mobile terminal performs a request to get real time information that is shown on the User Equipment (UE) screen. To get these real-time measurements a network function (the SP-GW) could be orchestrated by placing it in the corresponding edge cloud nodes, which are supposed to be small cells distributed across the factory. Together with the eMBB use case described before, this service could be deployed as a separate network slice in this demo.

Demo 4

Part of the Evaluation Case 3 (multi-service) in Demo 4 shows the economic impact of URLLC services. These services are mainly V2I, semi-automated driving and assisted driving. The cost impact of each of these services is analysed in terms of cost penalty compared to an eMBB-only network. In other words, the additional cost that an eMBB Mobile Network Operator (MNO) should invest to provide such services are shown in terms of CAPEX, OPEX and Total Cost of

Ownership (TCO). On the revenue side, the additional revenue for each of these services is analysed on a yearly basis for the study period (from 2020 to 2030). The demo also shows the traffic per year in Mbps/km^2 associated to these URLLC services.

2.2.3 mMTC

Massive Machine Type Communication services are addressed in Demos 1 and 4.

Demo 1

In this case the software part of this demo is used to simulate a 5G scenario considering the mMTC users moving in a specific area. Although in this case the UE does not require high throughput compared to the streaming service, the QoS is highly dependent on the latency. The demo considers the mMTC as one of the Reduced Latency service, for which the End-to-End (E2E) latency has to decrease.

Demo 4

These types of services are addressed in Evaluation Case 3 of Demo 4 (multi-tenancy), which focuses on the multi-service benefits of the 5G NORMA architecture. Within this evaluation case, in addition to other services, we examine the effect of operating with small packet sizes. For this, a number of specific example services have been selected, covering a broader set of categories in the context of a specific geographical study area (the city center of London). The example services were selected based on:

- Being actually applicable to the central London study area; and
- Promising significant incremental revenues and social benefits from the techno-economic assessment in WP2.

The example mMTC service is based on the Smart Cities service, which covers both, business-to-business and business-to-consumer services. More specifically, the demo focuses on smart metering (including also deep indoor smart metering use cases). In fact, in the demo, we analyse the additional revenue that smart meters or deep smart meters could bring to MNOs. The cost impact of mMTC services is shown in terms of cost penalty compared to eMBB-only networks and the traffic per year in Mbps/km2 associated to these services is also analysed.

Social value from smart city services such as smart grids and metering can potentially be identified from the analysis. We expect that direct revenues from smart city services may be limited particularly when considering the incremental revenues over what could be delivered with existing LTE systems. However, the ability of 5G to support network slicing and enable the rapid roll out and trial of these evolving services without the costly roll out of proprietary networks could bring their deployment and any potential revenues from these services forwards in time.

3 Demos Design and Implementation

As we've seen in the previous section, we have four different demos to demonstrate 5G NORMA key concepts by means of the implementation of different services. In the following sections, we provide details about their design and practical implementation.

3.1 Demo 1. Native Multi-Service Architecture

3.1.1 Introduction

5G NORMA supports service flexibility using the decomposition of mobile network functions and allocation of them to the edge cloud or central cloud. It is done adaptively based on the service requirements (e.g., bandwidth and latency) in addition to the transport network capabilities (e.g., available network capacity and transport network latency) [5GN-D31]. This demo aims to present the proof-of-concept for a flexible, adaptive, intelligent, and service-aware (de)composition of NFs and services. The key elements of this demo are shown in Figure 3-1 and described in the following:

- An SDM-C and software based eNB. These two components form the software part of this demo. They have been developed by NOMOR.
- A hardware eNB including the implementation of the SDM-C communication protocol, which composes the hardware part of the demo. This is provided by AZCOM.

The SDM-C is in charge of allocating NFs to optimise the network for high data rate services (e.g., HD video streaming) as well as reduced latency services (e.g., video live streaming). The SDM-C is connected to the eNBs using a bi-directional link to:

- Receive relevant KPIs, such as the received signal strength, the active service-type, and load in both downlink and uplink directions.
- Re-configure and trigger the placement of NFs in the edge cloud (i.e., deployed at eNB with low latency and limited processing resources) and central cloud (i.e., placed in the core network with higher latency and more processing resources).

The reports generated by the eNB are legacy network parameters, such as wideband channel quality indicator (WCQI), buffer occupancy, number of the Resource Block and the Modulation and Coding Scheme (MCS). The SDM-C sends reconfiguration commands to the scheduler of each eNB using a dedicated communication protocol (described in the following sections) via a southbound interface (SBI). More details about this SBI can be found in [5GN-D51].

The scheduler of the eNB defines the real-time scheduling policy on a Transmission Time Interval (TTI) basis (1 ms). On the other hand, the SDM-C takes a service-aware scheduling decision based on the feedback received from the eNB as explained in details later on. The scheduler of the eNB has been extended to support the control commands received from the SDM-C. Also, the HW eNB can support two different service types for the LTE dongles in the figure, which are:

- A Reduced Latency service, which can be considered as an URLLC-like service, and
- An enhanced Mobile Broadband service.

The SDM-C performs the reconfiguration of the network functionality between edge and central clouds based mainly on the QoS Class Identifier (QCI) and the service type. The reconfiguration leads the improvement of QoS, latency and throughput. SDM-C maps the relative QCI and throughput/latency parameters and communicates the corresponding reconfiguration commands to the eNBs via a dedicated message. Consequently, the throughput and latency may change and the eNBs report these changes periodically to the SDM-C[1].

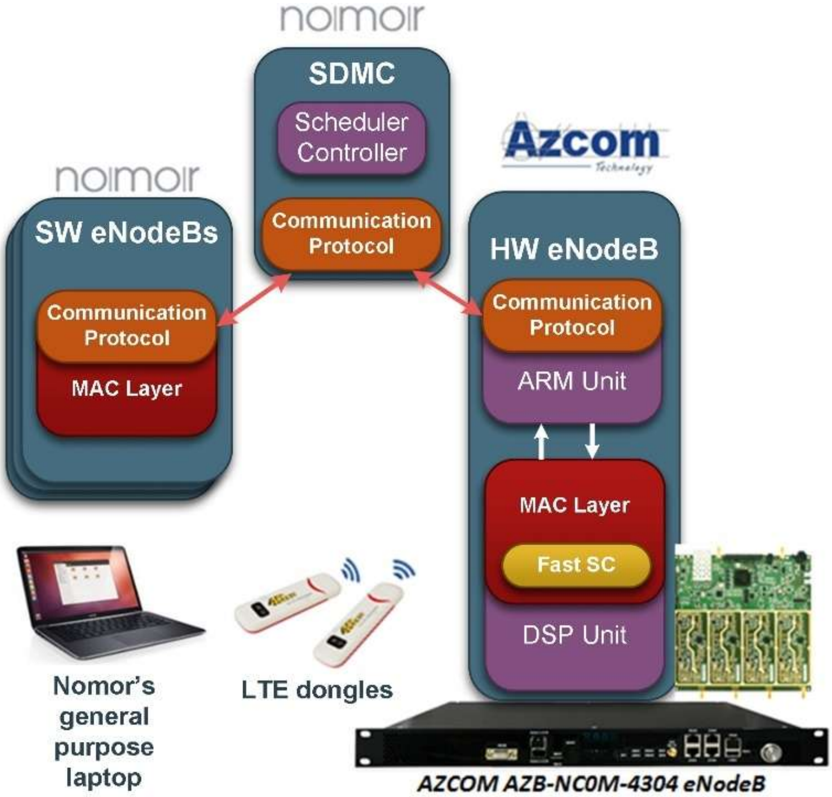


Figure 3-1: Key elements in Demo 1

3.1.2 Hardware Part

As we see in Figure 3-1, the hardware demo consists of the following elements:

- SDM-C,
- eNB,
- LTE dongles (two of them), and
- A Personal Computer (where two SP-GW instances are running).

The SDM-C is one of the key components in the 5G NORMA architecture, defined in both WP3 and WP5. In this case, the eNB works as a Physical Network Function (PNF) that has been adapted to be controlled by the SDM-C through the SBI interface, allowing network programmability. So, in this demo, the integration of PNFs into the 5G NORMA architecture as defined in WP3 is also evaluated [5GN-D33].

The demo has been designed to be shown in two phases:

Phase 1

In the first phase, there is a slice already set up, with two instances of the SP-GW component deployed in the edge cloud. It is assumed that, at the beginning, there are already two end-users requiring two different services:

[1] Regarding the orchestration of NFs, according the definitions in WP3 and WP5, it should be ideally handled by the SDM-O component. Since in this demo this part is not under analysis, it is emulated only the final effect of the orchestration, not considering any orchestration element. However, the orchestration functionality is addressed in Demo 2.

- A 4K video streaming service, which requires high throughput and can be categorized as an eMBB-like service, and
- An interactive live streaming service, which requires low and stable latency, which could be categorized as a Reduced Latency service.

Both of them are connected with a SP-GW placed in the edge cloud.

Based on the feedback from the eNB, the SDM-C recognizes that the users are requiring two different services. Since the eMBB requires more processing capacity at the central network than the RL service, it triggers a reconfiguration of the network to satisfy the eMBB user. Consequently, the SP-GW for the eMBB user is moved from the edge to the central cloud. This way, the adaptive (de)composition and allocation of mobile network functions depending on the service is performed, which is a key concept described in WPs 3, 4 and 5[2].

Phase 2

In the second phase, the behaviour of the SDM-C in an interference scenario is shown. The interference effect is simulated at the eNB side, because it must be kept under control to avoid any misbehaviour in the demo. When the interference increases, the 4K video user experiences a degradation of the quality in the video signal. Nevertheless, the QoS of the interactive live streaming service is not impaired, so the SDM-C, based on the feedback from the eNB, is aware of the reduction of the QoS in the 4K video streaming only. Consequently, it decides to change the scheduling policy of the eNB defined by a service aware scheduling algorithm implemented on it, which is another feature of 5G networks proposed in WP4 and WP5 (and which is described in the following). Since this algorithm requires more or less one second defining a new scheduling policy, the eNB continues using its own policy. This way, the degradation of the video is kept for sufficient time to recognise the impact of the interference. After that, the SDM-C sends the corresponding commands to the eNB, which, after reconfiguration of the scheduler, lets the MBB QoS to be in target again without any impact on the RL service QoS.

3.1.3 Software Part

The software part of the demo has been developed as a complement to overcome the natural limitations of the hardware part. The hardware part is limited in terms of size and coverage area, hence eNBs and UEs are deployed in a reduced-scale environment, such as those usually available in conferences and relevant events to show the demonstrations. Also, the software part helps to explain the overall concept of functional decomposition in a more general way. So, the main motivation for this software demo is to highlight the effects of the 5G NORMA innovations in a larger scale than just using the hardware components, and to investigate the gains at network-level, which could not be easily observed using only the hardware part.

Figure 3-2 below illustrates the main components of the software demo considering just a simple example consisting of a couple of small-cell eNBs and three end-users, which are differentiated based on the respective services running on them. As we see, two of the end-users are demanding HD video streaming, while the other one represents an autonomous (self-driving) car.

For the sake simplicity, at least one video streaming UE is connected to each eNB at all times, while the autonomous car may connect to either one of the two eNBs. Both eNBs are connected to the SDM-C and send feedback to and receive commands from the SDM-C.

[2] In fact, the re-orchestration part itself is only emulated here, since this part is out of scope in this Demo 1. The VNFs re-orchestration mechanisms themselves are actually addressed in Demo 2.

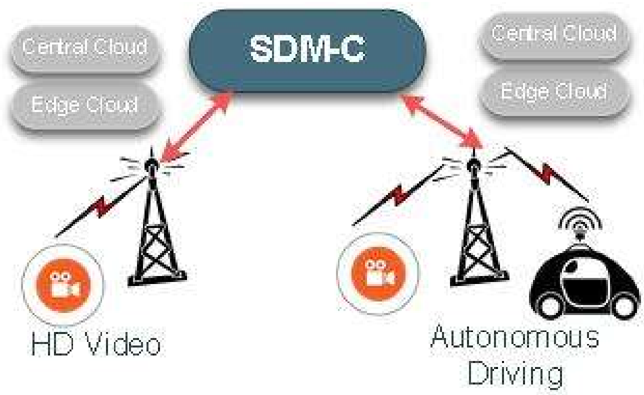


Figure 3-2: Demo 1 Scenario Layout

It is assumed that the autonomously driving UE can move in and out of the coverage area of these two small-cells. Upon arrival of the car into cell coverage, the SDM-C recognises that there is a delay-critical service, so a network reconfiguration should take place to cater for this change. It places the NFs in the edge cloud, as shown in Figure 3-3. As a result, the connected users can get an improvement in E2E latency. Consequently, the autonomous car can be controlled in a more responsive way, which provides an improvement in the end-user experience.

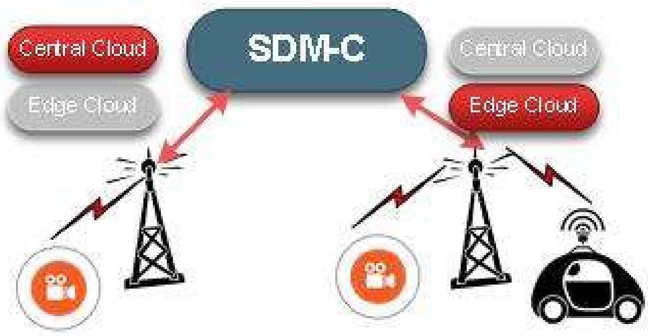


Figure 3-3: Network Reconfiguration into Edge cloud

On the other hand, when the autonomous driving service is not active in the cell (for example when the car user leaves the coverage area of eNBs), the SDM-C recognises this change and places the NFs in the central cloud, as shown in Figure 3-4 below. This change of the NFs' placement helps to enable centralised coordination, and hence, provides an improvement in throughput (the video streaming users can experience a better throughput, so they can receive the HD videos with higher quality).

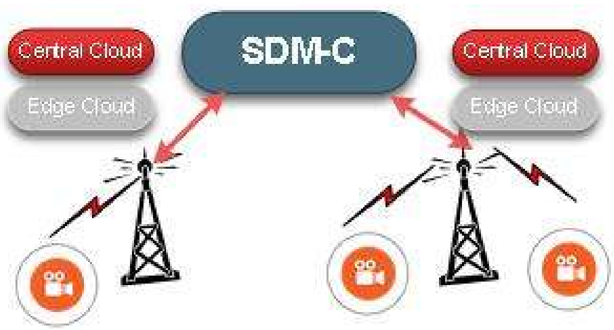


Figure 3-4: Network Reconfiguration as Central cloud

Additionally, an important feature of the software part is based on the possibility of executing the software simulations in parallel with the hardware eNB. Hence, the software part demonstrates how the SDM-C could be actually coupled with a real mobile network component, i.e., the same SDM-C application can communicate with hardware and software simulated eNBs.

SDM-C and eNBs (hardware implemented or simulated) exchange UDP packets on an IP socket (the structure of these packets is based on the communication protocol described in Section 3.1.5 below). The protocol is the same regardless of whether the connection is with a HW implemented eNB or with a simulated one. In both cases, the SDM-C will control eNBs and reconfigure their network functional elements into central or edge cloud based on the network parameters and the SDM-C's decision logic itself.

The overall layout of the software demo user interface is shown in Figure 3-5.

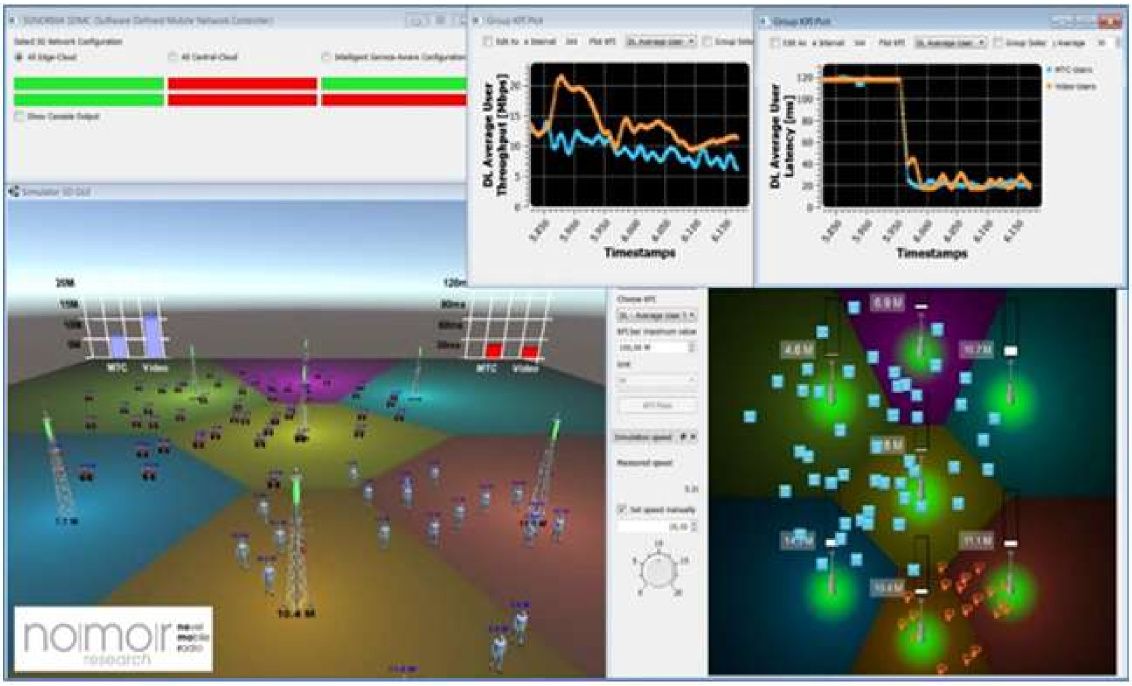


Figure 3-5: Software Demo GUI

It comprises three main parts.

a) The top left of the figure represents the SDM-C's Graphical User Interface (GUI).
b) The network simulator, which is divided in two parts:
    - The network simulator itself, on the right bottom of the screen. This simulator provides a 2D view of the simulated UEs, the eNBs, and their respective cell layouts. It also provides some controls to configure the simulation.
    - The plots visible on the top right corner.
c) The third part is visible on the bottom left part. This is a 3D-GUI which is used to provide a better and more realistic view of the simulated network layout, together with the simulated eNBs, UEs and some important KPIs.

The SDM-C GUI shown in the demo is used to control the SDM-C application and to switch between its different modes of operation. It has three main modes:

1. As it has been explained earlier, the main envisioned task for the SDM-C is to place specific eNBs' network functionalities into central or edge cloud configuration. The SDM-C can perform this task intelligently, based on the network parameters. This can be selected by the "Intelligent Service-Aware Configuration" button in the SDM-C GUI.
2. "All Edge cloud" mode; in this case the SDM-C is forced to switch all eNBs processing into the edge cloud configuration. This can be activated by selecting the "All Edge cloud" button in the GUI.
3. "All Central cloud" mode, to force all eNBs to the central cloud.

In the demonstrated simulation scenario, there are seven eNBs considered, as shown in Figure 3-5. These eNBs are serving a number of UEs, which have been divided into two groups, based on the traffic demand types of most of the terminals:

- The first UE group represents the video streaming users demanding HD video streaming. The increment of the throughput is the key parameter to achieve higher level of QoS. These users and their relative KPI-plots are shown with orange colour in the simulation, as can also be seen in the figure.
- The second group of users represent the machine-type communication (mMTC) users; their KPIs are presented in blue colour. These users are running a service with relatively small packet size while requiring lower latency. An example of such type of users could be mobile factory robots or autonomous cars.

3.1.4 Service-Aware Scheduling

Service-aware scheduling is an important feature of 5G networks that can be also placed "on top" of the SDM-C as an SDM-C application. The software demo also highlights the use of this feature for improving the system performance.

The hardware eNB supports user-selective scheduling based on the QoS requirement, the decision logic at the SDM-C can control and modify this UE-specific scheduling for specific users based on the feedback parameters. The decision logic developed inside SDM-C shall take into account parameters such as the type of service running at the user side, its pending load, the Reference Signal Received Power (RSRP), MCS, etc. Based on these inputs, the SDM-C will then decide how to set the UE-dependent priority metric for each user.

The decision shall also take into account the current network configuration. For example, if the network is configured to be central cloud, the service-aware scheduling should ideally give higher preference to a user running a particular service. In the case of a central cloud, throughput demanding services can be served in a better manner, and hence such services should be prioritized. On the other hand, if the network configuration is set to edge cloud, users with latency-critical service should be given higher priority; this way services that are more latency-critical can be served more efficiently.

The SDM-C sends new commands with a frequency of about one per second, i.e. roughly 1000 times slower than the scheduler. Since the commands do not change too frequently, the refresh rate is kept slow to avoid unnecessary overhead.

As a result of such service configuration and service-aware scheduling, it can be demonstrated that for different network configurations, i.e. central and edge cloud, different services show different performance. This emphasises the importance of a flexible network architecture for supporting a multitude of services in the future 5G wireless networks.

3.1.5 Communication Protocol Design

A communication protocol has been designed for communicating messages between SDM-C and eNBs. This is a proprietary communication protocol, designed only for this demo and it is not part of the concept study. However, this is an example of what a communication between SDM-C and eNB can be, see [5GN-D33] and [5GN-D52] for more detail. The communication protocol takes care of parsing and serializing messages in the correct format. In addition, the communication protocol also takes care of checking for message integrity and also identifying the source and target of different messages.

The different messages exchanged between SDM-C and eNBs can be broadly classified into two main types: feedback and command messages. Feedback messages are sent from the eNB to the SDM-C and contain information about the transmission parameters of the eNB and the UEs connected to the particular eNB. The command messages, on the other hand, contain commands or (re)configuration requests from SDM-C to eNB. These messages incorporate commands like network reconfiguration or a change in priority of the connected UEs.

The generic message structure is depicted in the following Figure 3-6.

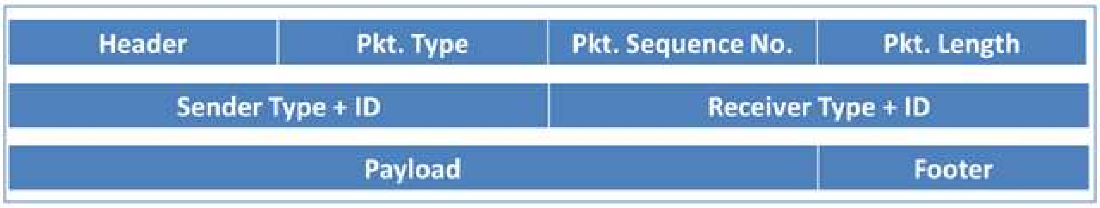


Figure 3-6: Generic eNB/SDM-C Messages Structure

Further details are described in Annex A.

3.1.6 Beamforming with Interference Fluctuation Migration

The beamforming algorithm implemented in the demo is helpful to improve the Signal-to-Interference-plus-Noise Ratio (SINR) of UEs by directing the beams towards the desired UE and reducing the interference to other UEs. This technique is used when eNBs are placed in central cloud configuration. The beamforming mechanism uses a linear array of antennas at the eNB with up to four antennas. Beamforming is used primarily for downlink.

The beamforming model deployed is based on a mathematical model, which is called delay-and-sum beam former [GOD-97]. By means of this, the beams are pre-computed by the eNB for all discrete angles in degrees (360x360). The computed beams are saved and then used by the eNB when transmitting to each UE based on the Angle of Arrival (AoA) information (also known as Direction of Arrival, DoA). The channel model used for simulations is the uniform distribution model [ESC-98]. In real-world LTE networks, AoA can be estimated by the eNB as a part of the uplink channel estimation procedure.

The beamforming method explained above helps to improve the SINR observed at the UE. However, it also increases the interference fluctuations because beams lead to higher level of interference on some PRBs and lower interference on others. To cope with this problem, there have been multiple methods implemented that aim to mitigate the interference fluctuations, which involves forcing persistency in scheduling, interference averaging and beam coordination among cells. Out of these methods, the beam coordination among cells is shown to produce the best results and is therefore used for the demo for interference fluctuation mitigation.

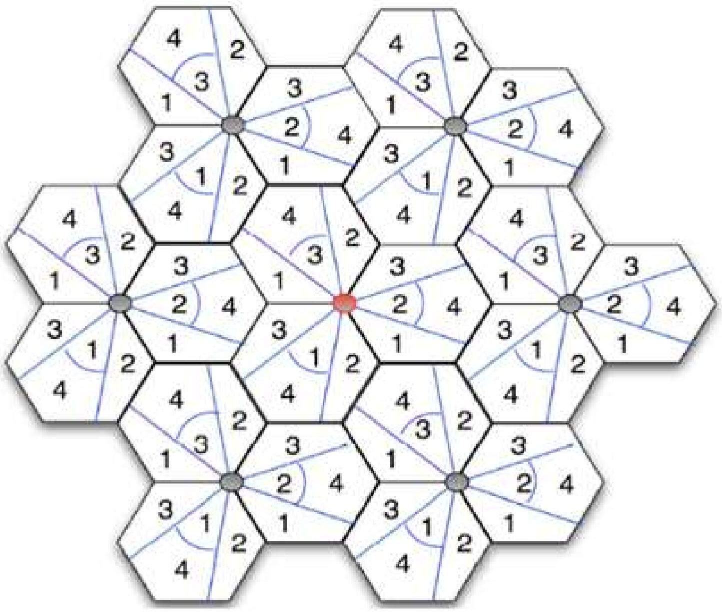


Figure 3-7: The beam coordination method

The beam coordination method divides the entire cellular region into four spatial regions. For each region, a portion of the total resources is reserved. The UEs in a particular region are assigned resources with higher priority from the resource pool reserved for that region. The division of the cells into regions is highlighted in Figure 3-7. It can be seen from the figure that the division of the region of each cell is done also considering the regions selected in the neighbouring cells. The regions are selected in such a way that the regions with the same index from neighbouring cells are not adjacent to each other. This way, two users in adjacent regions are mostly assigned resources on different PRBs, and hence they do not interfere actively with each other. As a result of this beamforming combined with beam coordination, it has been shown that the effective SINR, average throughput and also 5%-ile throughput of users increase as compared to the case without beamforming.

3.1.7 Extension of Demo with Virtual Cells

An extension of the original design of the software demo has been implemented focusing on the development of the Virtual Cells concept. This can be done because the definition of the SDM-C component in the WP3 also provides the concept of network programmability for 5G networks [5GN-D32]. This enables the control and coordination of multiple base stations, which is the ground stone for forming the virtual cells concept [CSSX-16]. The basic idea of this concept is illustrated in Figure 3-8.

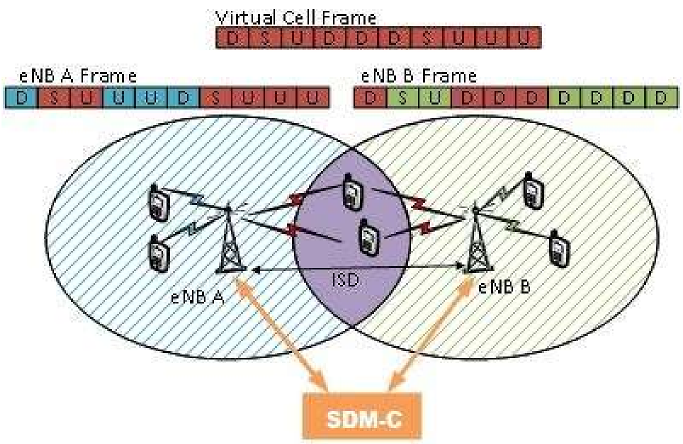


Figure 3-8: The Virtual Cell concept

The Virtual Cell concept tries to answer the dramatically changing traffic properties emerging due to the new mobile applications. To support such dramatic variation of traffic demands, the next generation of mobile networks has to be more flexible. In existing standardised LTE networks, flexibility is achieved by enabling Time Division Duplex (TDD) mode which supports TDD patterns with Uplink (UL) ratios ranging from 10 % to 60 %. However, changing TDD patterns individually per cell can lead to extreme UL-DL cross-slot interference. Formation of Virtual Cells can help to solve this problem.

The concept of virtual cells is proposed in [SSPR-14] and they allow the utilisation of resources (i.e., allocation of sub-frames) from multiple base stations and to meet the best of user's demands. By flexibly combining the uplink and downlink time-slots of both eNBs, a logical cell offering a individualized TDD pattern is offered.

This approach can also enhance the network performance, especially in pseudo congestion situations. Pseudo congestion is a situation when there is congestion in either uplink or downlink, while there are still enough resources in the other direction. This technique can also improve the QoS offered to the users by allowing the terminal to be allocated in another cell and have lower delays or by allocating extra resources to them. However, the transmission to the terminals in the virtual cell increases the interference on the other terminals. Hence, the formation of the virtual cell can be considered as the trade-off between the increase of throughput for the terminals placed in the virtual cell and a decrease of throughput for the other terminals (as the result of increased interference).

Hence, a Virtual Cell is a logical cell offering a different TDD pattern compared to the forming physical cells to enhance the network performance. The frame structure of the virtual cell is formed by the combination of the TDD patterns of different physical cells' (eNB A and eNB B) frame structures. It is apparent that the formation of a virtual cell at the edge of physical cells is beneficial compared to forming virtual cells at the centre. The cell-edge threshold is the difference between RSRPs of two adjacent cells, a key parameter for the formation of the virtual cell. The details of the algorithms for the formation of the virtual cells can be found in [5GN-D42].

Scenario Description

The NOMOR simulator and the SDM-C have been extended to support the virtual cell concept. Figure 3-9 describes the GUI.

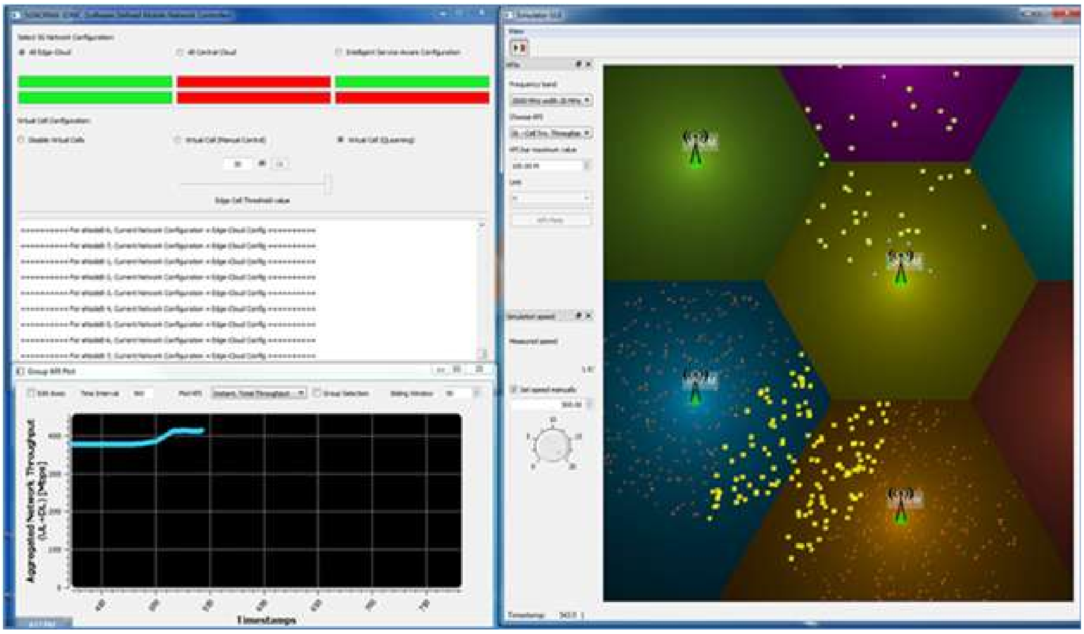


Figure 3-9: Virtual Cell SDM-C Extension

As we see, this virtual cell use case demo is consisting of:

- the simulator itself in the right bottom corner,
- the KPI plot window in the right top corner, and
- the SDM-C application in the left top corner.

The simulator has been extended to support uplink, TDD mode and cross-slot interference. For the demonstration of the gain achieved by virtual cells, pseudo congestion has been created between Cell A and Cell B located in the bottom left corner with blue and orange colour, respectively. Cell A (coverage represented in dark blue colour) has uplink-heavy traffic, generating congestion in uplink direction, while resources in downlink direction are available, and vice versa for Cell B (coverage represented in light orange colour).

The following options are available in the GUI:

- **Disable Virtual Cell**: This is considered to be the baseline/reference scenario. The network deployment is the same as in the previous section. Cell edge threshold is set to zero, which leads to all UEs within the simulator being cell centre UEs. Traffic properties of UEs are configured to create pseudo congestion. Due to the lack of cell-edge UEs and a virtual cell, KPIs of this scenario (system throughput and load factor) serve as a reference.

- **Manual Control Mode**: In Manual Control Mode of the scenario, cell-edge threshold is changed manually for the SDM-C. We refer to this mode as 'Manual Control' mode. The user can change the value of cell edge threshold from 0 to 30 dB. By applying a non-zero cell edge threshold, the virtual cell is formed between cell boundaries of Cell A and Cell B. UEs in the virtual cell have a different TDD pattern compared to Cell A and Cell B eventually helping to resolve congestion and enhancing system capacity. UEs served by the virtual cell are marked with yellow colour. The size of the virtual cell is linearly proportional to the applied cell edge threshold. Based on network response and traffic condition, the cell edge threshold has to be changed periodically. One practical limitation of this manual control mode is that it requires a network administrator to watch the network behaviour and change the threshold manually.

- **Q-Learning Mode**: This mode removes the limitation of manual control mode by introducing machine learning techniques (Q-Learning) [WAT-92] to mitigate the network administrator issue previously mentioned. With this method, a Q-learning agent can modify the cell edge threshold and watch the impact on the system, being able to learn from past inputs and keep modifying the cell edge threshold based on the system responses and the traffic conditions.

3.2 Demo 2. Service-Aware QoS/QoE Control

3.2.1 Introduction

This second demonstrator also focuses on the Network Slicing key concept, but specifically considers QoS/QoE-aware control and NFs virtualization and orchestration aspects. The general idea of this demo is to deploy two different network slices on the cloud infrastructure: one with a reduced latency service and another one with a mobile broadband service.

The demonstration has several goals. The main one is to show how an ETSI NFV Management & Orchestration platform could be used to deploy, manage and orchestrate the network services previously named on different network slices. In this respect, the main objectives are the dynamic re-orchestration of a particular NS forwarding graph, and also, the placement of certain VNFs in the appropriate host (simulating the 5G edge/central nodes). In both cases, QoS/QoE influence factors are used to trigger the re-orchestration function.

Although an initial draft of this demonstrator was early introduced in the Deliverable D6.1 [5GN-D61], relevant changes have been introduced since then. In summary, the main changes are:

1. In order to show the promising features of network slicing, the demo was refocused towards a different scenario involving the creation of two network slices to provide different services. This entailed the inclusion of the RAN stack functions as PNFs to properly show how different network slices are deployed in the core network.
2. To solve the difficulties found trying to use out-of-the-box VNF orchestration platforms, it was decided to focus primarily on an alternative programmatic solution consisting of implementing the VNF management and orchestration functions using the VIM APIs and Java-based libraries. Since the initial configuration of the demonstration was closely linked to using out-of-the-box solutions, this change had a significant impact on the task's plan.

With these changes, the demo set-up was re-defined and clarified. Now the focus is mainly on network slicing, QoS/QoE-control and NFs virtualization and orchestration. Anyway, two slices similar to those originally defined in D6.1 are kept in the design:

a) a Reduced Latency slice, focused on reading real-time physical measurements triggered by QR labels, and
b) an eMBB slice, focused on adding contextual captions to streaming media according to the user profile and surrounding context.

The intention with this demo is to deploy and orchestrate these two network slices on the virtualized cloud infrastructure. The following Figure 3-10 illustrates this concept:

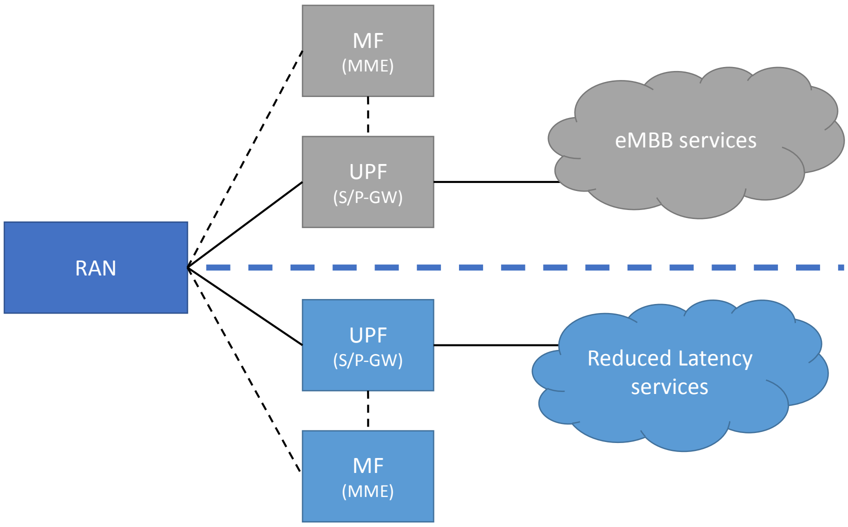


Figure 3-10: Network Slicing in Demo 2

The idea is that both slices are deployed on the same eNB and share the same spectrum. This is aligned with the Slicing Option #3 as it is defined in [5GN-D42]. For the core part, each tenant (i.e., a service) can run its own instance of the protocol stack (i.e., mobility management, gateways and the upper layers), performing the needed QoE/QoS trigger especially tailored for the selected case.

In the following subsections, we describe each slice providing a summary of their design and the showcase for each one, and also, the softwarised mobile network components which are common to both of them. But first, we're going to describe the cloud architecture that has been devised to implement this.

3.2.2 Cloud Architecture

As introduced in subsection 2.1.1.1, the cloud architecture for this demo is based on an ETSI NFV MANO implementation extended by 5G NORMA features. As explained in the Deliverable D6.1 [5GN-D61], the approach to build-up this cloud architecture was, whenever possible, to benefit from already available general purpose out-of-the-box software projects. Hence, we've cross-checked different MANO platforms; the following ones were evaluated:

a. TeNOR [TEN]. This platform was the one initially selected for implementing the NFVO and VNFM blocks. TeNOR is the orchestration and management platform developed for the T-NOVA project [XIL-14], another European Union R&D founded project in the FP7 framework programme that released some of its work as Open Source. TeNOR was specifically developed by two of the partners participating in that project: Altice Labs (former "PT Inovação e Sistemas") and "I2CAT Foundation"; Atos also collaborated developing a Marketplace service for it.

   The reasons to consider TeNOR as a primary option were mainly the following: (i) it was an Open Source project, so we could access the code and modify it if necessary, (ii) it was designed to integrate with OpenStack (a kind of de-facto standard for implementing cloud computing solutions [OPSTK]) and (iii) Atos had previous experience using this platform. Although we knew it didn't meet all the requirements needed for the demo, we thought it could be feasible to introduce the necessary changes to add the missing functionalities. However, this platform was finally discarded due the following main reasons:

- After investing a considerable amount of time and resources we saw it was not possible to deploy the VNFs we had developed at that moment. Of course, the platform was suitable to deploy and manage the sample VNFs from the T-NOVA project, but not the VNFs specifically developed for this demo.
- We found that the platform was not designed to provide the VNF placement functionality, which is a must for this demo. In fact, the platform does not support (at least when we investigated it) the placement of VNFs on specific network nodes.
- The possibility to modify the code to add the missing functionality was out of scope because: (i) there was no official involvement of the partners who developed this platform in 5G NORMA, so it was difficult to get direct support; (ii) the Atos staff who previously had used this platform was finally assigned to other projects, so it was also not possible to get support from them either; (iii) the available technical documentation was poor; this was a serious difficulty when trying to modify the source code to add the missing functionalities.

b. OSM (Open Source Mano) [OSM]. Initially we considered this platform as a backup alternative for TeNOR. Since TeNOR turned out to be not suitable, we evaluated this possibility as well. OSM is a relevant project started by several important founding members, including Mirantis, Telefónica, BT, Canonical, Intel, RIFT.io, Telekom Austria Group and Telenor; now the OSM community includes about 50 different organisations. The project is hosted at the ETSI facilities and targets delivering a MANO platform closely aligned with the ETSI NFV MANO reference.

   The main issue with this platform was that its initial release and further development overlapped with our work in the 5G NORMA project. OSM issued four releases which coincided in time with the development of our own demo[3], so when we initially started evaluating this platform, not all the necessary functionalities for the demo were available. For instance, as with TeNOR, the platform did not support the placement of VNFs on specific nodes, which was an important requirement for our purposes. Also, the access to the source code to implement the necessary changes in the platform was not trivial (it was a new platform), so we finally discarded this option as well.

c. OpenBaton [OPB]. This platform is defined as an extensible and customizable NFV MANO-compliant framework capable of orchestrating network services across heterogeneous NFV Infrastructures. Virtually, it could manage a diverse ecosystem of VNFs. It provides good documentation resources. During our testing, we successfully deployed what they call a "hello world" scenario, i.e., a basic testing network service composed of dummy VNFs. However, we did not succeed in deploying the specific VNFs we needed for our demo. We discovered that for deploying our VNFs, it was necessary to modify them by developing what they call a 'VNF Manager adapter'. Although a specific SDK is provided for this purpose, we considered this an inconvenience, because some of the VNFs necessary for the demo were already finished. Also, the uncertainty about whether this general purpose platform could effectively meet the specific needs for the demo once the VNFs were adapted, discouraged us from focusing on this path.

So, after exploring these different possibilities (and also evaluating others with less depth[4]), we finally decided to discard the usage of these kind of general purpose out-of-the-box orchestration platforms and focused primarily on an alternative programmatic solution

[3] Release ZERO (May 2016), Release 1 (October 2016), Release 2 (April 2017) and Release 3 (November 2017).

[4] Of course, many other orchestration platforms are also available (e.g., Tacker, OPEN-O, CORD, Cloudify…) but we consider an exhaustive analysis of all of them to be out of scope of this demo.

consisting of implementing the VNF management and orchestration functions specifically for this demo using the VIM APIs. At the end, the main reason for this decision was the fact that the learning process for using these types of platforms could consume considerable resources, and the dependency on external parties. Hence, we finally considered that greater chances of success could be achieved using an ad-hoc approach specifically for this demo instead of a general purpose out-of-the-box solution.

The main conclusion in our context is that these kinds of platforms seem to be in a young state of development right now, so for practical and specific implementations (like the one we address in this demo), it seems more convenient to mainly rely on the VIM and implement on it the required functionality (see Section 4.2 below for additional details regarding the lessons learned about this).

Hence, the final approach for implementing the cloud architecture has been a hybrid approach with OpenStack (*Pike* release) to implement the VIM, and a programmatic solution to implement the remaining blocks. On one hand, OpenStack is a free and open source out-of-the-box platform (Apache 2.0 license) considered a kind of de-facto standard for implementing cloud computing solutions [OPSTK]. On the other hand, the VNF Manager and the NFVO components have been specifically developed for the demo using the Java™ programming language together with the OpenStack4J open source library [OS4J].

Of course, what we've implemented here is not a fully functional general purpose NFV Orchestrator and VNF Manager (we consider this is out of the scope for this demo) [5], but just the necessary functionality to fulfil the requirements of the demo itself. Figure 3-11 shows this approach in relation to the ETSI NFV MANO architecture:

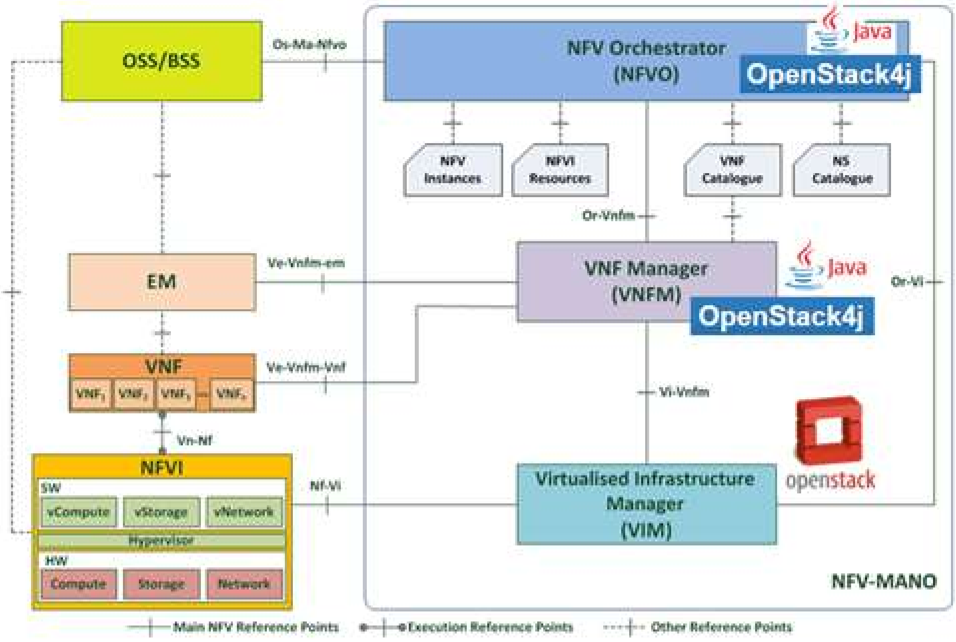


Figure 3-11: ETSI NFV MANO Implementation

The implementation of this demo has been affected to a greater extent than expected by the selection of the orchestration platform, and especially, its further development, which has been particularly challenging. Although the framework and the great majority of the necessary components are already available, while this document is being written, the last integration and

[5] Other Horizon 2020 projects are specifically focusing on developing general purpose NFV Orchestrators (see [XIL-14] and [DPK-16]).

testing tasks on the final HW platform are still in progress, which will be completed by the end of the project.

On the other hand, regarding the physical infrastructure for this demo, besides a reduced testbed environment with a couple of general purpose PCs that was enabled for developing and debugging purposes, a dedicated demo environment has been set-up consisting of the following main elements:

- one Intel 12 Core i7-3930K PC @ 3.2 GHz with 32 GB RAM and 1 TB of local storage, and
- two twin AMD FX 8320 eight-core processor PCs @ 3.5 GHz with 32 GB RAM and 1 TB of local storage.

Beside the CPU, memory and storage figures, the main criterion to select this hardware was to have at least two of these nodes with exactly the same physical features. This is a requirement necessary to perform the VNF live-migration functionality. Figure 3-12 shows the network topology for this set-up.

The two nodes on the left (*buyo* and *zamorano*) are the twin PCs used to work as compute nodes in the architecture (VNFs are executed on them) while the node on the right (*bichobola*) act at the same time as network controller and storage node. Besides, a general purpose 10/100Mbps router is used to interconnect the nodes through a private isolated network (orange), while another equivalent router is used to connect the nodes with the external network through an OpenVPN Server (blue lines). All nodes are executing the Linux CentOS operating system (release 7.2)[6].

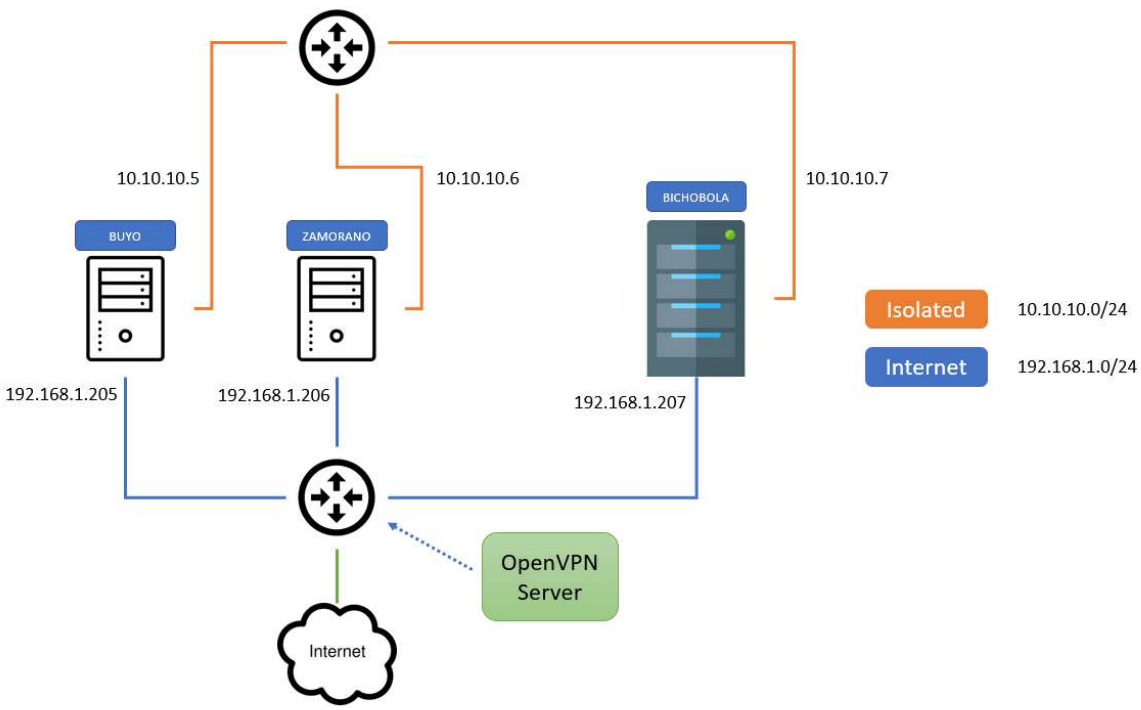


Figure 3-12: Demo 2 Hardware Environment

3.2.3 Softwarised Mobile Network

The mobile network architecture employed within this Demo 2 relies on two well-known software implementations of the LTE Stack: OpenAirInterface (OAI), see [OAI] for more

[6] This infrastructure has been provided by the UC3M.

details, and srsUE [SRS]. The former is an open source software implementation of an LTE mobile network (including RAN and Core), while the latter is an open source implementation of a UE and the eNB.

We describe here the setup that involves the UE implemented using the srsUE software. However, as the demonstrated innovations involve the infrastructure part only, this setup may also be converted to use a commercial of-the-shelf (COTS) LTE modem or a mobile phone.

Figure 3-13: UE Hardware Setup

The hardware elements used for this are depicted in Figure 3.-13. The UE part is configured as a Laptop running the srsUE software, which provides connectivity to the terminals attached through short range wireless interfaces. We selected this setup for using RF cabling in order to not rely on spectrum available.

The radio used in this setup is implemented using two Software-Defined Radio (SDR) USRP B210 Cards by Ettus Research, one on the UE side and the other for the eNB. As the USRP B210 features two antenna ports (one RX and one TX), we cross-connected them using two SMA to SMA cables, adding a 30 dB attenuator loop to each of the cables. Figure 3-14 represents the connected radio setup.

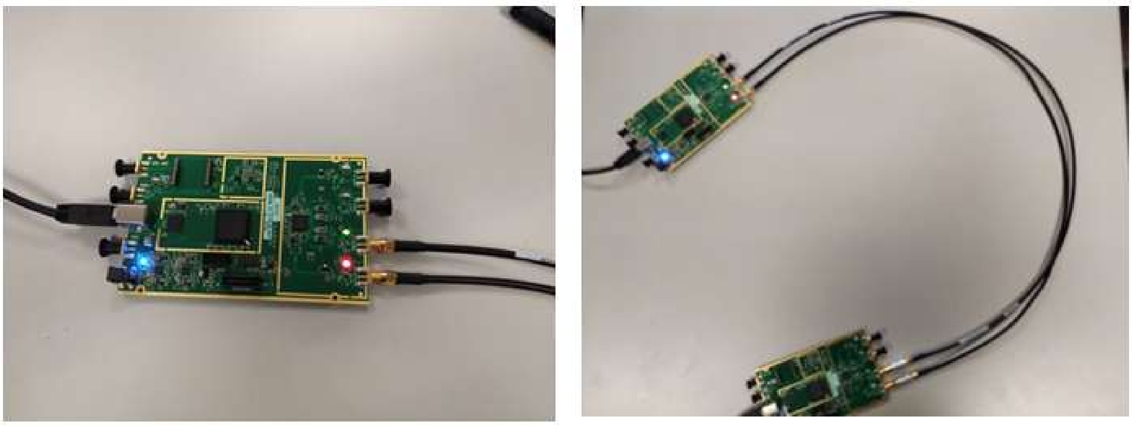


Figure 3-14: Demo 2. Hardware Radio Platform

The USRP is used by the LTE software (on both UE and eNB sides) to modulate and demodulate the in-phase and quadrature (IQ) samples produced and received by the software on one of the available licensed bands. For this demonstrator, we use Band 7 at 2.6 GHz. The full configuration parameters for the radio frontend are listed in Table 3.1.

Table 3-1: Configuration Parameters for the Radio Frontend

Parameter	Value
Frame Type	FDD
Band	7
Downlink Frequency	2.68 Ghz
Uplink Frequency	2.56 Ghz
Number of RB in Downlink	75
Tx Gain	90 dB
Rx Gain	125 dB

On the network infrastructure side, the SDR card is used by SRS eNB to provide connectivity to the UE. The SRS suite implements all the functionality of the RAN while different instances of the core network are provided by the OAI core. More specifically, the software components available are:

- the eNB: This is the main part of the SRS suite. It implements the Physical (PHY), Media Access Control (MAC), Radio Link Control (RLC), Radio Resources Control (RRC) and Radio Resource Management (RRM) layers of the LTE stack;
- the Home Subscriber Server (HSS) and the Mobility Management Entity (MME), from OAI; and
- the Gateway (GW) function, joined into a SP-GW module (from OAI).

The goal described in Deliverable D6.1 [5GN-D61] was to implement network slicing down to the RAN. Unfortunately, that proved to be very challenging to achieve, so we decided to revert to a network slicing demonstrator just for the core network functions.

We created a virtual function for each of the core network functions which is going to be used within a network slice. More specifically, the VNFs are

- HSS: Ubuntu 16.04 LTS – Kernel 3.13.0-112-generic – 4GB Disk – 1GB RAM
- MME: Ubuntu 16.04 – Kernel 3.13.0-112-generic – 4GB Disk – 4GB RAM
- SP-GW: Ubuntu 16.04 – Kernel 4.7.7.-oaiepc – 4GB Disk – 1GB RAM

The mobile network setup is also composed by one PNF, the eNB which is run on a bare metal server for performance reasons. The eNB hardware and software configurations are as follows:

- eNB: Ubuntu 16.04 LTS – Kernel 3.19.0-61-lowlatency – 500GB Disk – 16GB RAM

The overall setup including the different interfaces used by our software is depicted in Figure 3-15. The eNB block in this figure is just an emulator used for testing purposes. This element is replaced by the actual eNB (a physical network function) in the final demo set-up. This represents just one network slice.

As we see, the RAN protocol stack deployment does not provide many innovation per-se, but it is a fundamental enabler for the overall Demo storyline. The two services implemented in the demonstrator (the mobile broadband and the reduced latency) will share the same RAN infrastructure, so the RAN should be able to:

- Provide the minimum bandwidth and latency requirements needed for the two envisioned services: mobile broadband and reduced latency.
- Provide an easy way of classifying the incoming traffic, providing a way of multiplexing and de-multiplexing traffic associated to the two different slices. This entails software modifications to the state of the art software for the open source mobile network stack.

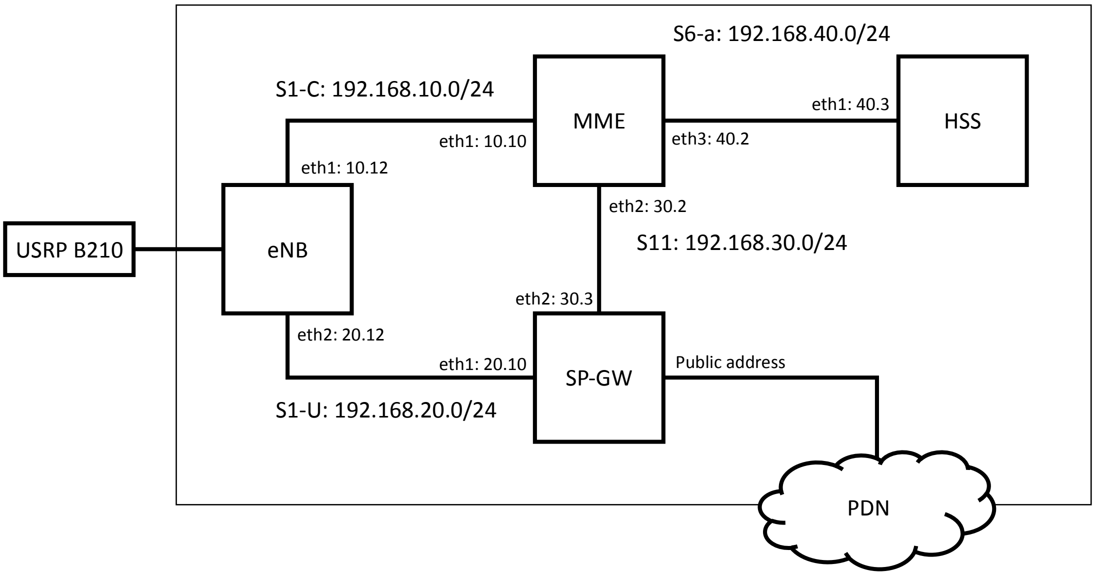


Figure 3-15: Softwarised mobile network architecture for a single network slice

3.2.4 Mobile Broadband Slice

The idea for this slice is to deploy a service to add context-based add-ons on a live video streaming signal depending on the user profile and preferences and certain environmental conditions. These parameters (user profile and environment conditions) are defined here as QoS/QoE influence factors. They could trigger re-orchestration of VNFs when required. The generated add-ons can be subtitles generated depending on the subscriber preferences (language, colour, size, position, etc.) or other support videos with the sign language translation (e.g., to help people with a hearing impairment). These add-ons will be explicitly requested by the user at any time during the video playback or triggered by environmental conditions (e.g., by the ambient noise), so a good synchronization between add-ons and the main video signal is a must.

Figure 3-16: Using the MBB Slice to Add Sign Language Translation

Beyond the demo, such a service could be used to implement other different real-life applications such as:

- generate add-ons in the form of banners, in order to organize banner advertising campaigns with the possibility to segment the audience using different parameters (e.g., the video contents, subscriber profile or subscriber location);
- introduce graphic elements on the screen as part of the audio-visual narrative (this is already a common practice in certain Asian countries); or
- generate real-time captions providing contextual information about what the user is watching (relevant data about actors and characters, locations appearing in a film, the clothes or objects appearing in a scene, etc.).

As example, Figure 3-16 shows a possible case where a main video signal is being enriched with subtitles and a sign language interpreter to assist a hearing-impaired user:

Beside the possible real life applications of this demo, the three VNF orchestration-related main concepts we wish to demonstrate with this demo are:

i. On-boarding of the network service itself, i.e., the automatic deployment of the necessary VNFs driven by network service descriptors where the operational layout and requirements are defined;
ii. Dynamic update of the NS Forwarding Graph (FG) depending on QoS/QoE measurements (QoS/QoE-awareness); and
iii. Placement of VNFs to specific compute nodes, to simulate the placement of NFs in the edge or core cloud depending on the service requirements.

Figure 3-17 shows the main blocks we have considered for the demo to develop this concept.

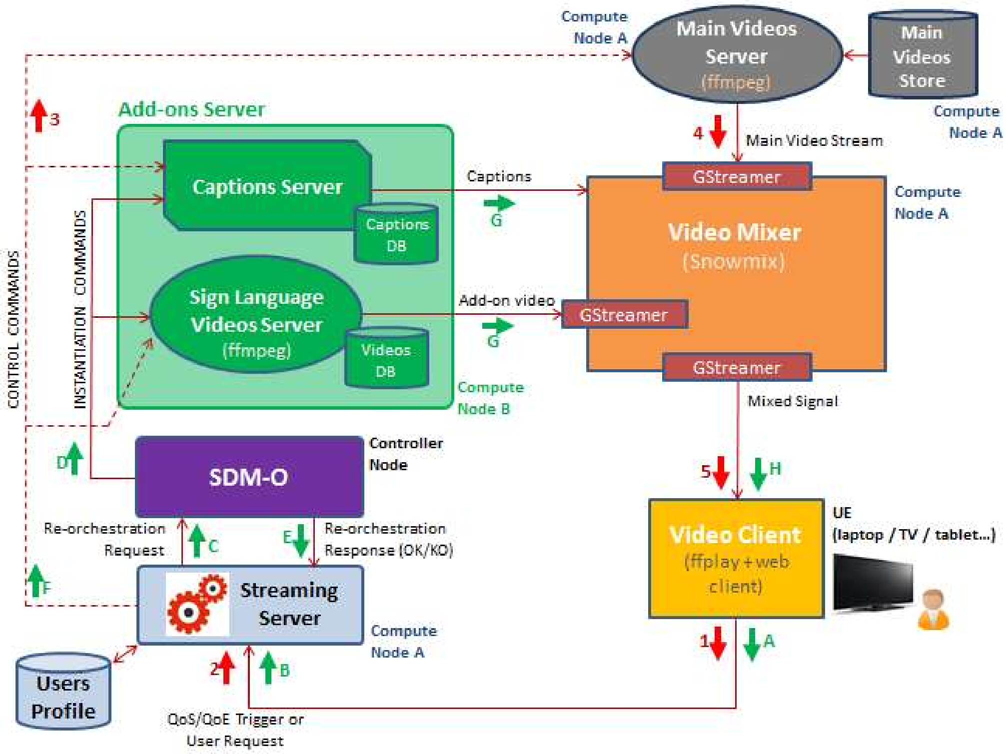


Figure 3-17: Demo 2. Detailed MBB Slice Blocks Diagram

In the figure, the red arrows labelled from 1 to 5 show the initial interaction where the user uses a Video Client (yellow block on the right) to request the playback of a certain video. In this case, this would be just a regular video service providing the requested video to the user where the Streaming Server block (bottom left) is acting just as a proxy for the Main Videos Server (where the video files are actually stored). The video streaming pass through the Video Mixer block (although it is not mixed with anything else) just to avoid disruptions in the service when add-ons were actually inserted.

Subsequently, when add-ons are requested by the user (green arrows labelled from A to H) the Streaming Server will communicate with the SDM-O block, which will instantiate the Add-ons Server (green block). As we see, this Add-ons server is split here into two internal blocks: the Sign Language Videos Server (which is used to provide the sign language videos) and the Captions Server (for providing subtitles). Once the appropriate add-on server is instantiated, the SDM-O will inform the Streaming Server block, which can send control commands towards the Add-ons Server (dashed line). The selected add-on (video and/or subtitle) is injected into the Video Mixer block which delivers the mixed video signal to the final user.

As we see in the figure, the Video Mixer block is implemented using Snowmix, an Open Source[7] very flexible command line tool for dynamically mixing live audio and video feeds which supports overlaying video, images, texts and graphic elements as well as mixing audio [SNMX]. Snowmix is itself based on GStreamer [GSTR], also an open source[8] development framework for creating multimedia applications such as media players, video editors, streaming media broadcasters or video mixers (like in this case). It supports an extensive catalogue of video and audio formats. These components have been embedded in a special-purpose VNF based on the Ubuntu Minimal OS (release 16.04) which is deployed on the Compute Node A as previously described.

On the other hand, the Add-ons Server is actually split in two VNFs (also based on the same Ubuntu Minimal OS) corresponding to the two blocks in the figure, i.e., the Captions Server and the Sign Language Videos Server. Each VNF basically contains a database with the necessary caption and video files and a service running on TomCat [TMCT]. The service exposes a REST API (see Annex B) used by the Streaming Server block to execute the necessary control commands once the VNF instances are up and running. The Caption Server communicates with the Video Mixer block using a Snowmix-specific protocol which makes it possible to specify where and how the add-ons are placed within the video. The Sign Language Videos Server uses *ffmpeg* [FFMPG] to stream the add-on videos into the mixer.

Figure 3-18: Demo 2. MBB Use Case GUI

As we see, the Streaming Server block is a central component here that could be also an independent VNF running an Ubuntu Minimal OS. It performs three primary functions:

a) As its name states, it works as a server for the final user. Specifically, it embeds an HTTP server to what the final user can access (using a general purpose web browser) to request the video playbacks and the available add-ons.
b) Decoding the HTTP requests from the user, it implements the logic to trigger the instantiation of the necessary add-on server (through a request to the SDM-O) and to

[7] GNU General Public License version 3.0 (GPLv3)

[8] GNU LGPL 2.1

send the necessary control commands to these instances, and also, to the Main Videos Server (see Figure 3-17).

c) To provide a good synchronization level between the main video stream and the add-ons. Add-ons can be requested at any time by the final user, so they must appear properly synchronized with the main video and with a minor delay.

The Main Videos Server is also an independent VNF implemented on the same Ubuntu OS and running on the Compute Node A as previously mentioned. It contains a database with all the possible videos the final user could access.

The SDM-O block (purple rectangle in the figure) represents the 5G NORMA specific block defined in WP3. It is basically a process developed using the Java programming language acting as a server attending the commands from the Streaming Server VNF. Of course, although this is not explicitly represented in this figure, it also communicates with the underlying ETSI NFVI MANO components (i.e., NFVO, VNF Manager and VIM) in order to orchestrate and manage the virtualised resources. This process is executed on the Controller Node (cf. Figure 3-12: Demo2-HW-Environment).

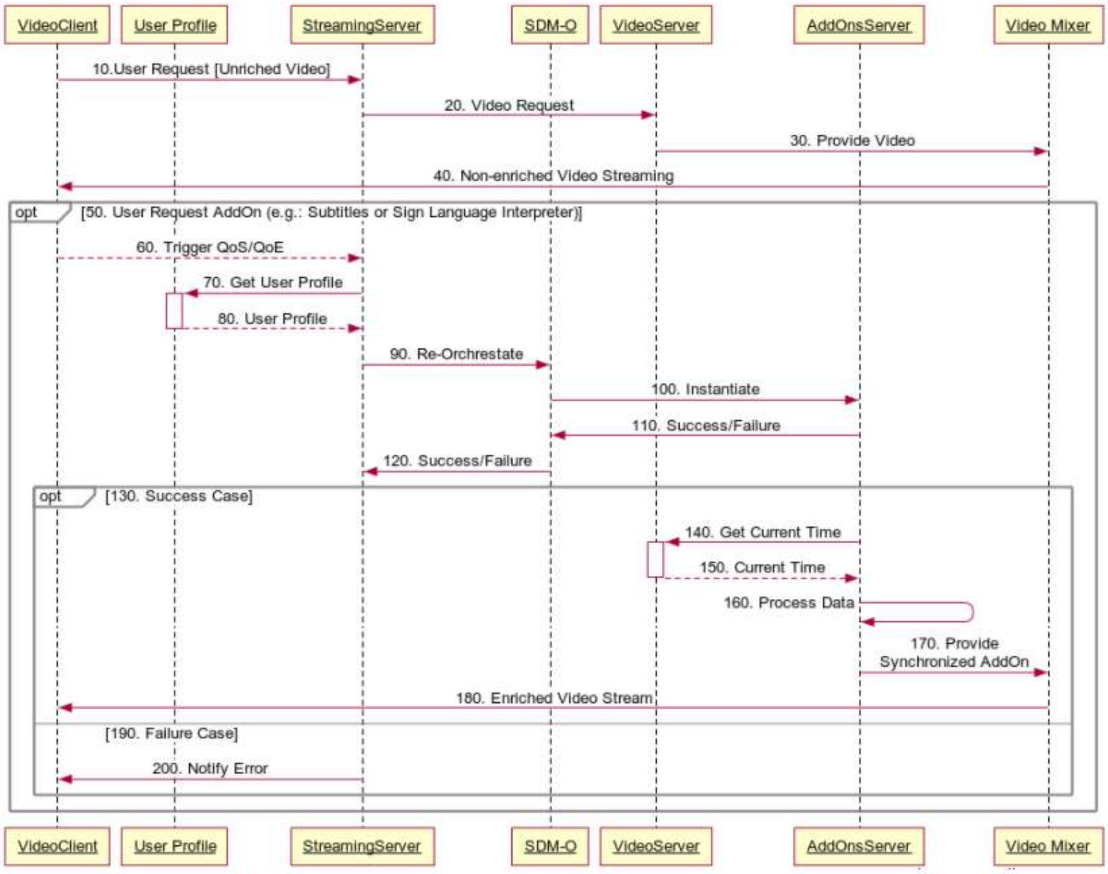


Figure 3-19: Demo 2 MBB Simplified Sequence Diagram

Finally, regarding the End-User equipment (yellow block in the figure), it consist of two different elements, i.e., the video player itself and a web browser with a GUI for the end-user to manage the service. For the former, the *ffplay* video player [FFP] is used. This is a simple and portable media player using the SDL media library [SDL] and *ffmpeg* libraries.

Regarding the GUI, it has been specifically developed for the demo using a web interface based on jQuery. Figure 3-18 shows this GUI. As we can see it provides a basic functionality to cover the demo purposes, i.e., the selection of different users (to show how different user profiles could have different effects on the video player) and the simulation of certain QoS/QoE influence factors (environment noise in this case) that could automatically trigger the re-orchestration process.

As a summary of the whole service provided with this slice, Figure 3-19 shows a simplified sequence diagram which illustrates the use case:

3.2.5 Reduced Latency Slice

This second slice implements a service that needs specific KPIs regarding the end-to-end delay and, therefore, needs an especially tailored orchestration in order to maintain the targeted QoE/QoS for the users. This will be a very common case in the envisioned landscape of 5G services. Ultra-reliable low latency communication (URLLC) is precisely one of the foreseen use-cases that will be enabled by 5G networking. As the goal of this demonstrator is to showcase the innovations of 5G NORMA in the field of orchestration and network slicing, we focused on a service that, although it does not require the sub 5 ms end-to-end delay targeted by URLLC, it has more stringent delay requirements than eMBB.

The service we implement for the reduced latency slice is a version of an augmented reality application that may be useful for future factory applications. Here, aided by a pair of Virtual Reality (VR) glasses, a worker can see real-time information about specific objects in its environment (such as the ones flowing on a conveyer belt).

Our implementation limits the image recognition process needed by such applications with an easier and low-complexity QR code recognition. The service plot is sketched as follows:

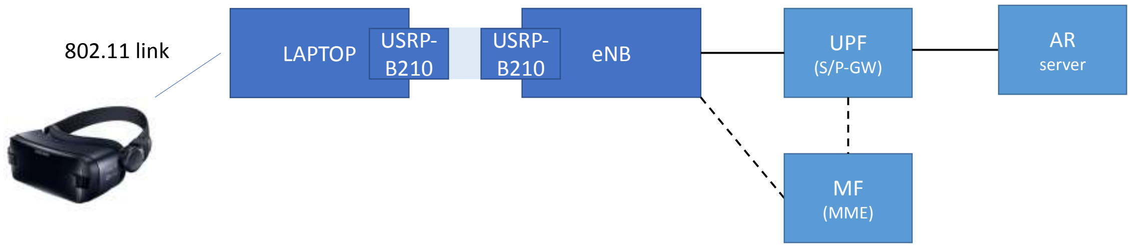


Figure 3-20: Reduced Latency Demo Plot

The user wears a pair of VR glasses and can observe the surrounding environment. There, different QR stickers link to potential sources of real time information. The VR application (see the application layout below) promptly shows this information to the user, updating it with a very high frequency. For our tests, we link stickers with the server running the virtualized network, showing to the user information about, e.g., the CPU load, its temperature or the used bandwidth.

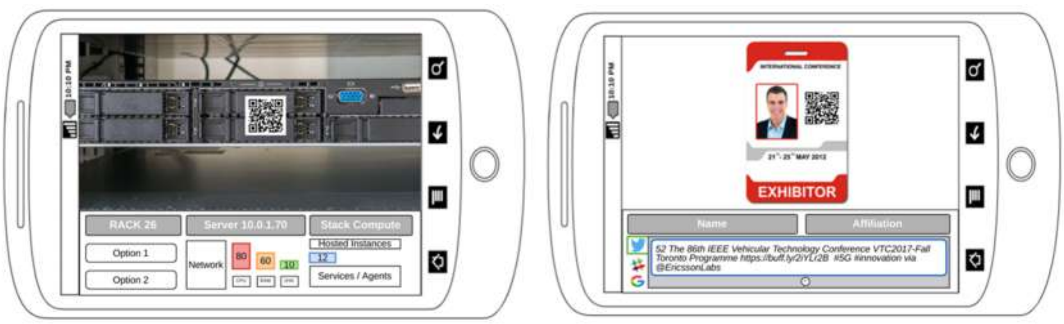


Figure 3-21: Two application layouts for the reduced latency slice

The reduced latency slice runs on the same infrastructure used by the mobile broadband slice. It completely shares spectrum and RAN (we do not classify traffic at MAC level directly), but it goes through two different core networks that run in several VNFs, as explained in the following sections.

The mobile phone is a Samsung Galaxy S6, running Android 6.0 Marshmallow. The VR glasses are Samsung Gear VR. The AR application features code written in both C++ and Java, using the EasyAR [EASYAR] framework for the QR code recognition and the OpenGL (GLE20) library for showing the information on the focused QR code.

The application is continuously requesting information from the AR server, which gathers information from a live source such as the current load of the compute nodes used in the demo. It is based on a REST API developed in Python using the Flask framework [FLASK]. The database where the data is stored is managed by MongoDB [MONGO].

The demo plot is complementary to the mobile broadband slice, providing a way of orchestrating the VNFs differently and according to the service purposes. The NFVO orchestrates resources such that the core of the reduced latency slice and the AR server always has a way to provide the needed QoE/QoS requirements.

3.2.6 Main Demo Innovations

In this section, we describe the main innovations addressed by this demo, which mainly are the following two:

- the network slicing adaptation of the mobile network stack, and
- the VNF mobility.

3.2.6.1 Network Slice Adaptation of the Mobile Network Stack

As discussed above, the slicing of the demo happens in the core. Spectrum and RAN are shared between the two slices, while each tenant can deploy its own core NF for each service. We used the former 3GPP LTE core network functions (i.e., SP-GW, MME and HSS) to provide the functionalities that will be provided by the future 5G NFs. However, in order to enable such functionality in the core network, also the RAN had to be changed to provide this novel functionality. More specifically, we implemented the RAN slicing Option 3 (slice-aware shared RAN), as defined in [5GN-D42], which is represented in the Figure 3-22.

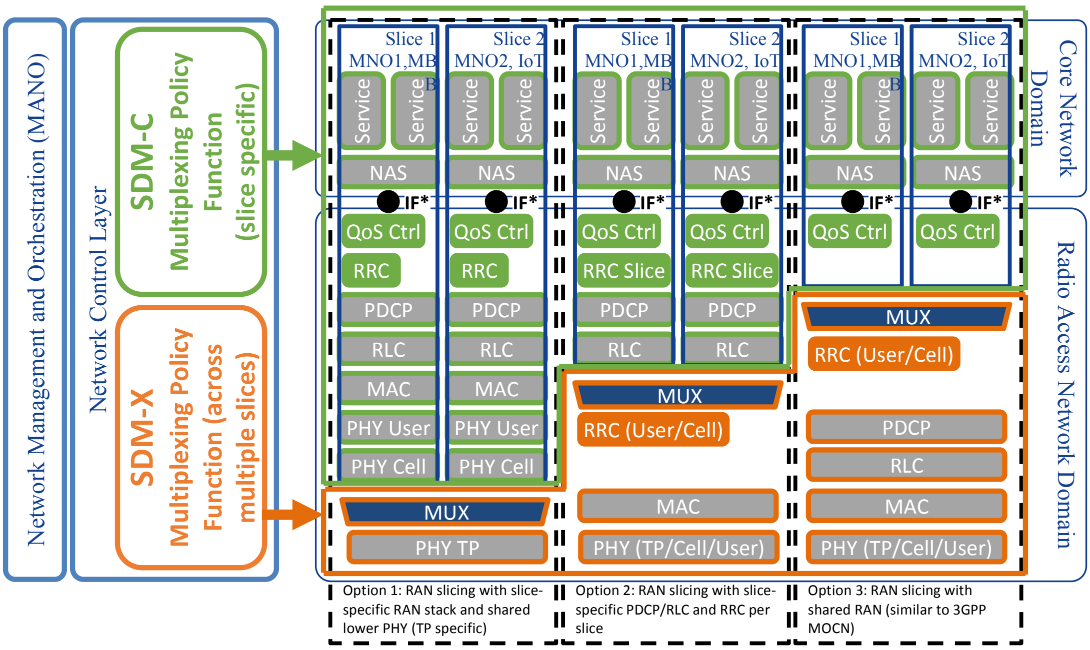


Figure 3-22: RAN Slicing Options as Defined in WP4

For achieving this we modified the two software components available in the SRS suite, the SRS eUE and the srsUE in order to achieve the needed functionality.

The SRS suite defines different classes for the different tasks that have to be fulfilled in the RAN:

- a common library (formerly known SRSLTE) that performs the encoding/decoding operations, up to the MAC;
- the RLC, RRC and the Packet Data Convergence Protocol (PDCP) layers; and
- specific modules for the UE (the UE module for the data plane of the UE) and the eNB (S1AP, that manages the connectivity for the core control plane, and GW for the GW connectivity).

In order to perform two registrations against two different Non-access Stratum (NAS) instances, we duplicated (i.e. an instance for each slice) the UE module and the RRC layer, which has to trigger two NAS connectivity requests. The data is finally multiplexed and de-multiplexed at the PDCP module, according to the traffic's final destination address. That is, by triggering two NAS registration procedures, the UE obtains two valid IP addresses from each slice GW. Therefore, it can create two virtual network TUN interfaces in the UE hosts that finally represent the two slices. The final network setup, not representing the physical deployment of VNFs, is presented in Figure 3-23.

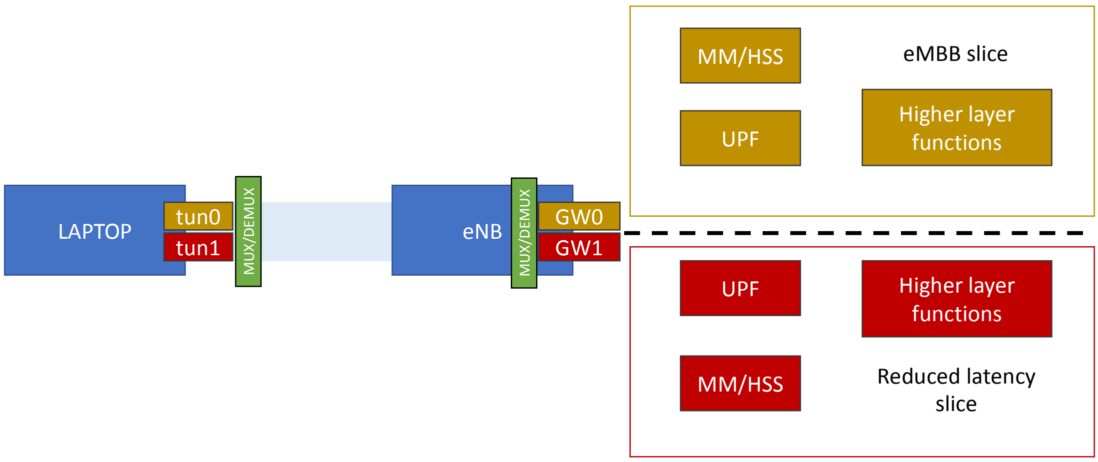


Figure 3-23: Demo 2 Software Architecture

All the source code developed for the software mobile network stack used in this demonstrator has been released as open source and is available on GitHub at https://github.com/GinesGarcia.

3.2.6.2 VNF Mobility

One of the objectives in WP6 is to show some key principles of the 5G NORMA architecture, like a service-aware adaptive allocation of functions to different network nodes using VNF mobility concepts.

Regarding this, the two slices in the demo (MBB & Reduced Latency) have been devised considering the possibility of dynamic relocation of VNFs between different compute nodes, which are acting as hypothetical edge and central cloud nodes in a real 5G network. For our particular case, we've addressed this problem using the so-called 'live-migration' functionality provided by OpenStack. We performed two different experiments:

- The 1st one, based on the so-called 'block live-migration', just needs a network connection from the source node to the destination. Basically, the complete virtual machine is transmitted without service interruption. In our specific case, VNF instances can be quite big (tens of GBs once instantiated), so it would be necessary to have a very high bandwidth and dedicated network connection to achieve fast migration times. With a common network connection like the ones we have in our testbed the process takes times in the range of seconds to minutes[9]. Furthermore, this approach also reduces the VM processing power during the migration time, so we considered this is not an acceptable approach for the demo (and for a future 5G network as a whole).

- The second approach is called ‘shared storage live-migration’. As the name states, it is based on using a shared storage which is accessible from both, source and destination host. In this case, performance has been quite good (in the range of few milliseconds); the following screenshot shows one of our tests while the SP-GW VNF is being migrated from one node to another in our lab environment:

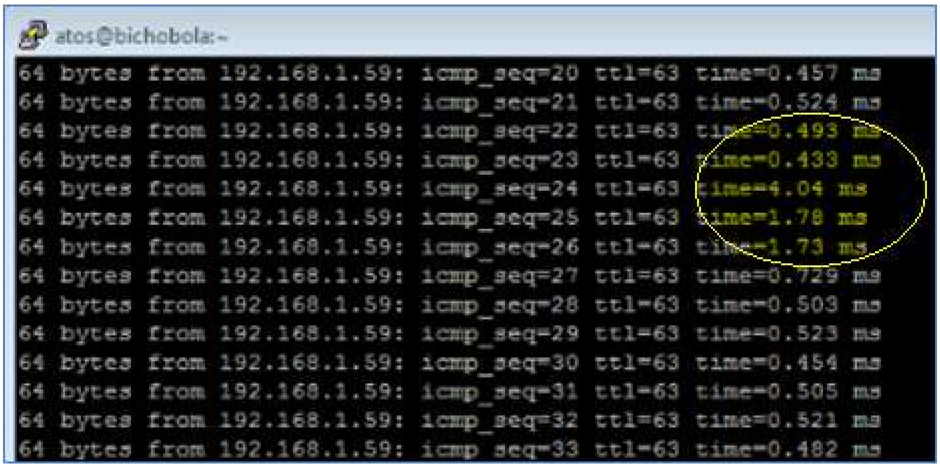


Figure 3-24: Ping During SP-GW Live Migration

In this case, traces are showing the output of a ping command between the controller node and the VNF being migrated (see Figure 3-12: Demo 2 Hardware Environment). The marked values show the moment of the migration where the delay is slightly higher for a few seconds (traces are generated at a rate of one per second).

Of course, although the performance is quite good using this ‘shared storage’ approach, this shared storage node is a drawback in itself. This procedure has well-known inconveniences, among others:

- to add shared storage nodes implies to design the network topology according to this;
- the shared storage node becomes single point of failure;
- the network itself becomes a single point of failure;
- the network security (the shared storage should be placed in a separate secured network); and
- even with the good performance results we’ve seen in our demo case, the network latency could impact performance, especially for certain very high requiring LL scenarios in 5G.

However, beyond the specific demo purposes, we think this problem is not inaccessible. Different solutions could be addressed. The most obvious could be to reduce the size of the VMs to be migrated, so a regular network connection could be used to move the whole VM with good performance (without a shared storage). Also, the usage of Containers [KT-14] or Unikernels [AM-13] could help on that. Another possibility is the improvement of the shared storage option to tackle the known weaknesses, e.g., adding redundancy to the single points of failure and improving the security.

[9] E.g., a 50 GB VNF transmitted at 10 GBps (a very good connection nowadays) would take 5 seconds. This same VNF on a 100 MBps connection (a more common connection) would take more than 8 minutes. See [BISW] for a more complete study about this.

3.3 Demo 3. Secured Multi-Tenant Virtual Network Resources Provisioning via V-AAA

3.3.1 Introduction

The KCL Secured Multi-Tenancy Virtual Network Resources Provisioning via Virtualized Authentication Authorization Accounting demonstration is a complementary demonstration in 5G NORMA aiming to illustrate 5G NORMA tenant data isolation, secured virtual network resources provisioning, and hierarchical and distributed virtualised AAA (V-AAA) solutions proposed in WP3. This demonstration also aligns with the flexible 5G network architecture and V-AAA principles developed in that WP.

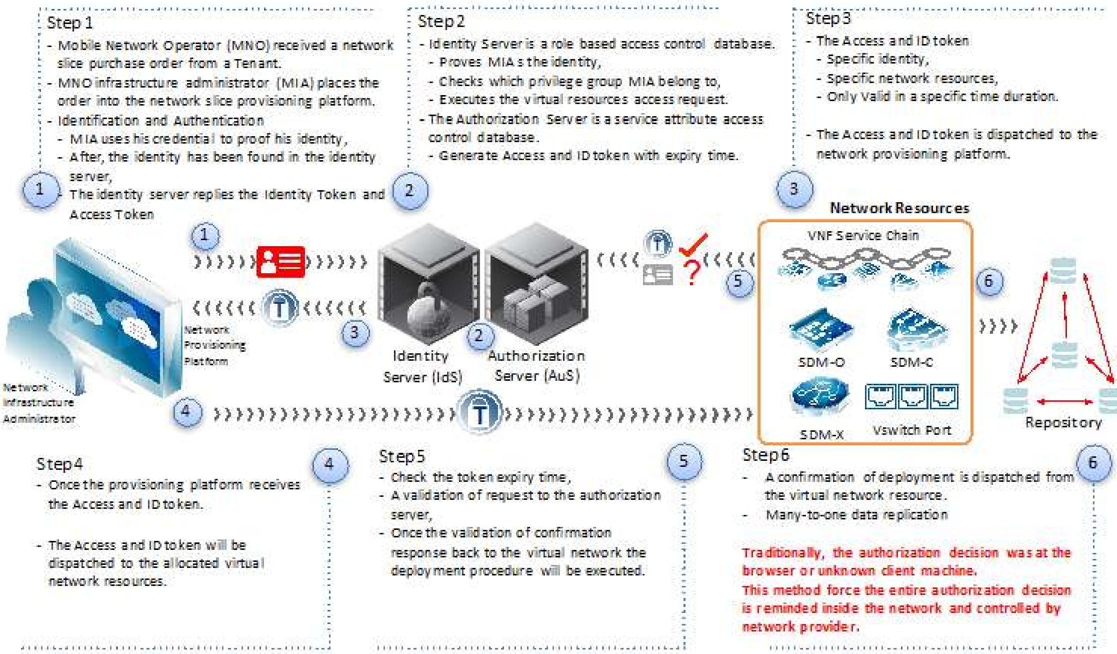


Figure 3-25: V-AAA network resource provisioning storyline

Demo 3 has been designed to execute a series of well-defined steps, as a form of storyline, summarized in Figure 3-25. These steps are the following:

- Step 1 – A Mobile Network Operator (MNO) receives a network slice purchase order from a Tenant. The MNO has number of different types of administrators who have an access right to the network slice provisioning platform. In this demo, the MNO assigns this Tenant's network slice provisioning task to an Infrastructure Administrator (IA). The IA would use the web network slice provisioning platform and places the network slice order into the network slice provisioning platform. Additionally, the web network slice provisioning platform would require a user credential login to the provisioning platform and claim his identity via web network slice provisioning platform.
- Step 2 – The web network slice provisioning platform uses multi-factor authentication to strengthen the authentication process. Then, IA enters his/her login credential and user secret to claim his/her identity from the Identity Server (IdS). The credential and user secret should be hashed and dispatch the hashed value to an Identity Server (IdS). The IdS is a role-based access control database for proving the IA's identity, checking the level of privilege group IA belong to, and executing the network resources (network slice) access requests and verifications invocation to Authorization Server (AuS).
- Step 3 – Once, the AuS confirms the access right of a network resource. In this case, it would be the network slice which the resources blocks are across the core network and the access network. The AuS keeps zero knowledge of the user credential and secret in

the AuS. The Access and Identity token are generated under a specific identity, network resources and time duration. The Access and ID token is dispatched to the web network provisioning platform.

- Step 4 – Once the provisioning platform receives the Access and ID token then it will be dispatched to the allocated virtual network resources when IA ready to use the token.
- Step 5 – This is a validation process in confirming the token expiry time and right network resource to access. This validation is confirmed from the AuS. Once the validation of confirmation response back to the virtual network the deployment procedure will be executed.
- Step 6 – A confirmation of deployment is dispatched form the virtual network resource and the many-to-one synchronisation or data replication will be executed.

3.3.2 Hardware

Demo 3 has been implemented using a testbed deliberately based on cheap commodity hardware, namely, Raspberry Pis (RPIs). Figure 3-26 depicts the commodity hardware and the network topology set-up for the demo. Moreover, Figure 3-27 is a photograph of the set-up, also annotating the components therein.

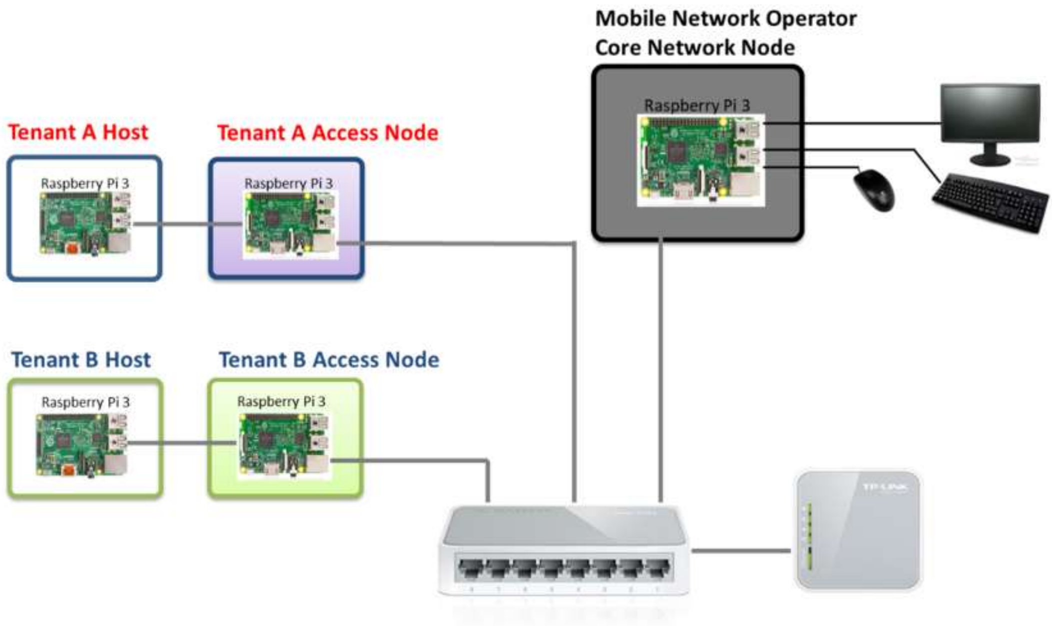


Figure 3-26: Hardware and Demonstrated Network Topology

Only one of the RPIs is connected to user input controls (keyboard, monitor, mouse), to control the demo and associated software, and present a user input means. The interaction with other RPIs is done via remote connection. A TP Link TL-SF1008D network switch is used to network the RPIs, and a generic gateway provisions Internet access. Our implementation is based on 5 RPI 3's: One as the Mobile Network operator, and two RPI 3's for each of two tenant domains. The RPIs are powered using a generic USB hub.

The RPIs and their power source (although not the particular USB hub used in Figure 3-27) have been integrated in a Lego-based housing as a compact, self-contained, and somewhat portable setup.

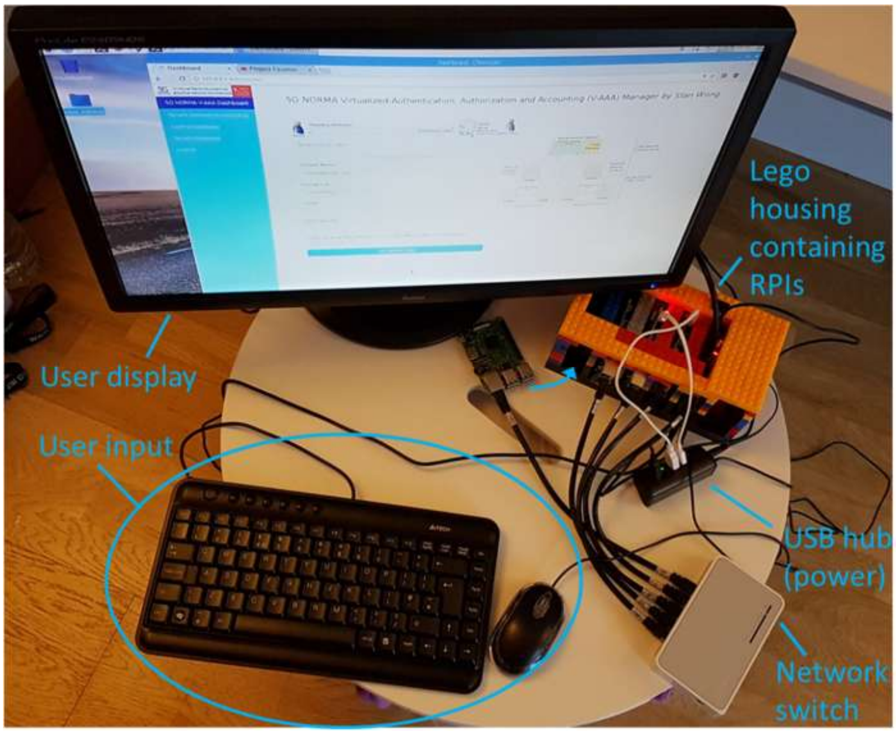


Figure 3-27: Photograph and Annotation of the Hardware Setup

3.3.3 Software

The Software for Demo 3 comprises 4 parts: The web application, the database element, the SDN controller, and the virtual switch. The web application provides the user interface and the ability to input parameters and visualize results. The web application is created and run using the Django web framework [DJANGO] based on Python. The database element provides a registry of users/tenants and their access tokens, and it was implemented using an Apache CouchDB database including an interface through Project Fauxton [FAUXTON]. The SDN Controller represents the managed resources by the tenant, and for this the Ryu SDN Framework is used [RYU]. The virtual switch is controlled by the SDN Controller, which is implemented using Open vSwitch [OVS]. The functionality demonstrated at the virtual switch is the ability to add a port.

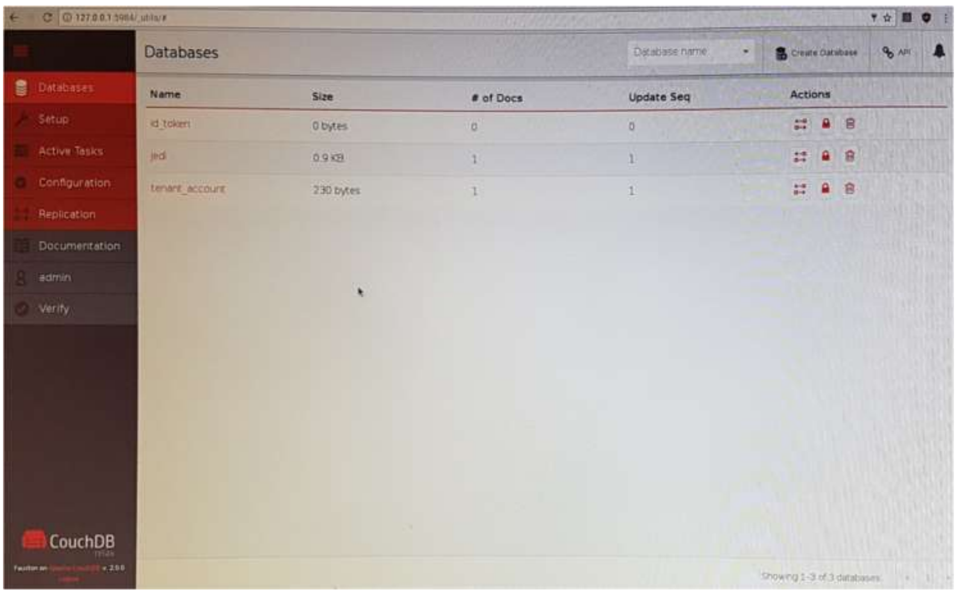


Figure 3-28: Project Fauxton CouchDB Instance Running During the Demo

Figure 3-28 is a screenshot of the Project Fauxton CouchDB showing records created during a demo run for a registered user ("jedi"), a tenant account, and an ID token (the latter not being populated yet). Moreover, Figure 3-28 shows the successful invocation of the management/web interface.

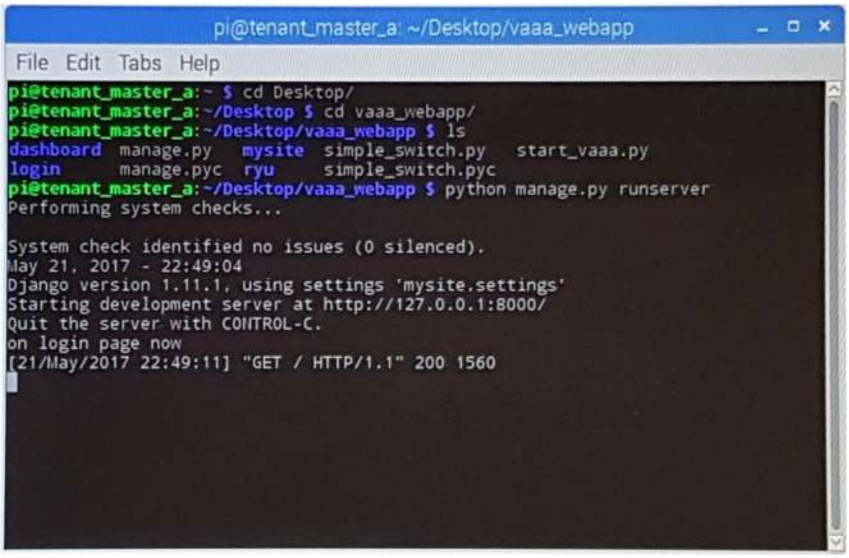


Figure 3-29: Successful Instantiation of the demo (web/management interface)

A key objective of the Demo 3 commodity hardware testbed is to show secured provisioning and deployment based on a tokenization technique. This is achieved using an implementation of OpenID Connect and Open Authentication Protocol version 2 (Auth 2.0). The hierarchical and distributed V-AAA provides this secure approach in provisioning and deployment of network resources at the central cloud and the edge cloud.

Basically, the V-AAA uses the tokenization technique for secure identification, access, termination, provisioning and deployment of network resources and services via a provisioning platform. When a tenant requires the creation, manipulation or termination of network resources, it uses the same sequence to access the network or service resource to request a grant for authorization, and to access a token to a specific network entity and resources in a particular period of time. The sequence of message exchanges is illustrated in Figure 3-30 (the individual entities shown in the figure are implemented using RPIs).

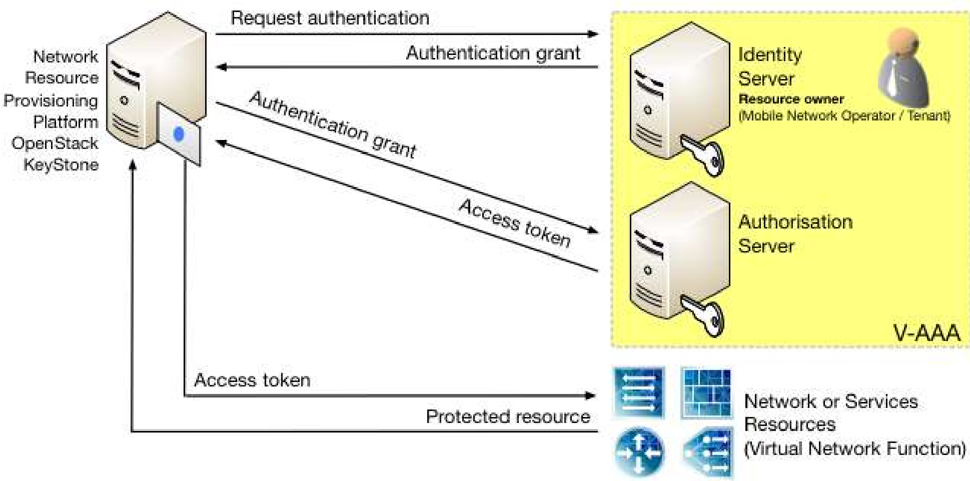
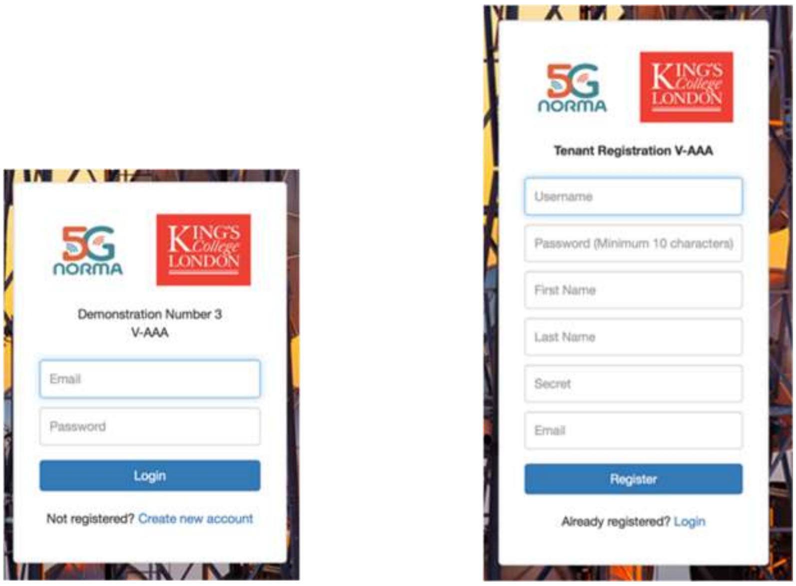


Figure 3-30: Basic Network Resources Authentication Sequences

Careful work has been undertaken to create the web application and integrate it with the running backend framework. The V-AAA web application and design of the user interface are discussed here, including a discussion and verification of the concept through the given images and the information therein. The objective of the V-AAA web application is to visualise the backend V-AAA information in the frontend, and to allow for passing parameters to it. Through this and other access means, key objectives of Demo 3 can be seen to be achieved.



(a) (b)

Figure 3-31: Demo 3 V-AAA web application user interface tenant (a) login page, and (b) user registration page

Figure 3-31 (a) shows the login page. This page provides V-AAA web application access and distinguishes users gaining access to the network resources. A user registration page has also been developed, allowing users with appropriate rights to access resources to be registered and records stored thereof in the CouchDB database. The user registration page is shown in Figure 3-31 (b).

Figure 3-32 shows an identity token and the access token request form page. The aim of this page is to generate the identity token request to identify the server, verify the tenant's identity, to obtain the identity token, and to access the token that will be stored in the CouchDB.

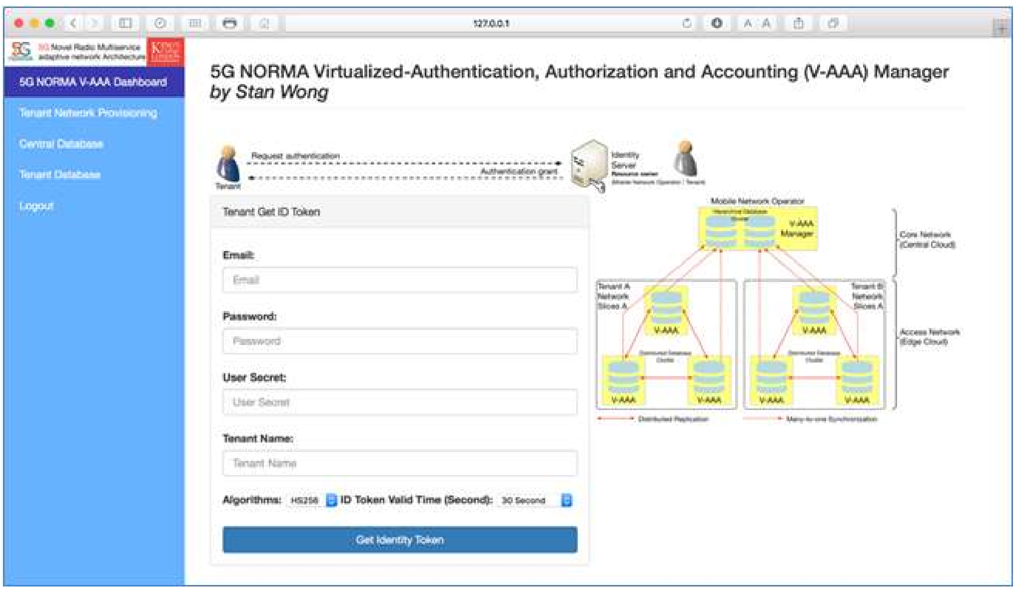


Figure 3-32: Demo 3 V-AAA dashboard identity token and access token request page

Figure 3-33 gives an example of network resources (virtual switch port) provisioning and deployment page that we have developed, which might be tied in to the backend. The aim of this is to provide a simple virtual network resources provisioning and deployment platform, and to dispatch the access token to a specific network entity. The input form indicates the specific network entities, the SDN controller, and the virtual switch. In this page, the intention is to show that the tokenization technique provides a secured technique to gain access on any virtualized network resource. This tokenization technique can also be applied to provision or deploy network slices.

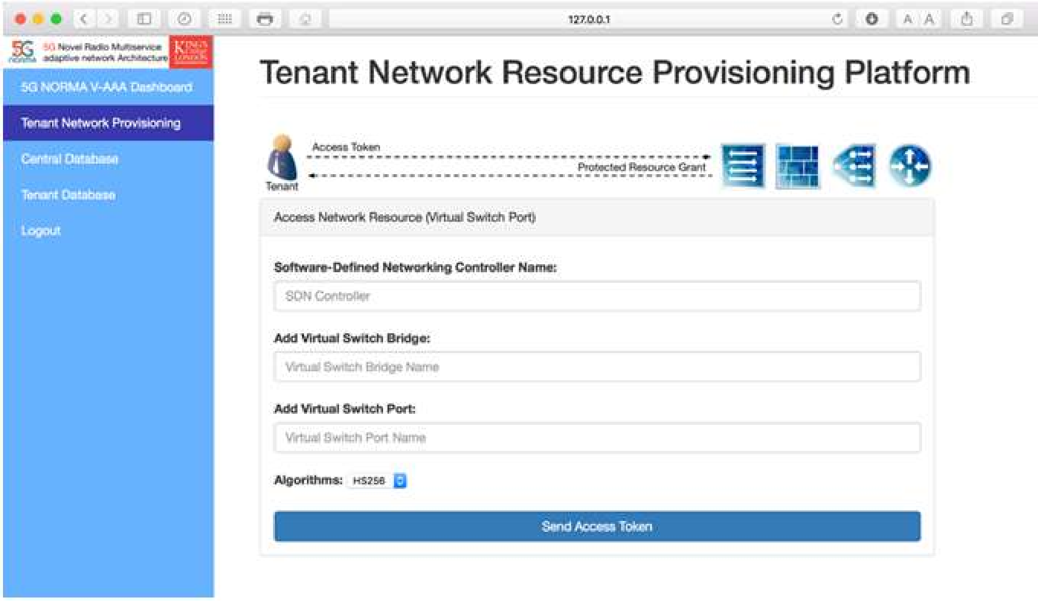


Figure 3-33: Demo 3 Tenant Network Resource Provisioning Platform Page

Figure 3-34 gives an operator's hierarchical database page. The aim of this is to show the tenant data isolation, tenant data many-to-one synchronization/replication, and tenant data consolidation at the operator hierarchical database. The page uses a collapse panel to display different tenants' information. The displayed tenant information is retrieved from the hierarchical database. In this Figure two different tenants' information are shown (*Jedi* and *Sith*), as well as their virtualized network resources and access time.

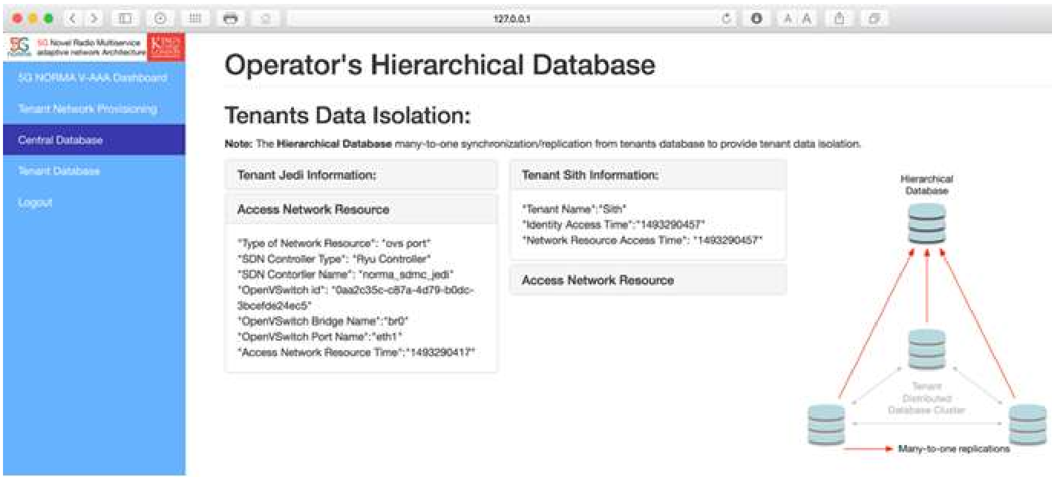


Figure 3-34: Demo 3 Mobile Network Operator's Hierarchical Database

Figure 3-35 likewise gives a tenant's distributed database page, whereby again the information is directly mirroring the backend. The aim of this page is to show the bidirectional replication of tenant data. The page uses the collapse panel to display tenant information (e.g., a list of the virtualized network resources, identity token, and access token). The displayed tenant information is retrieved from the distributed database.

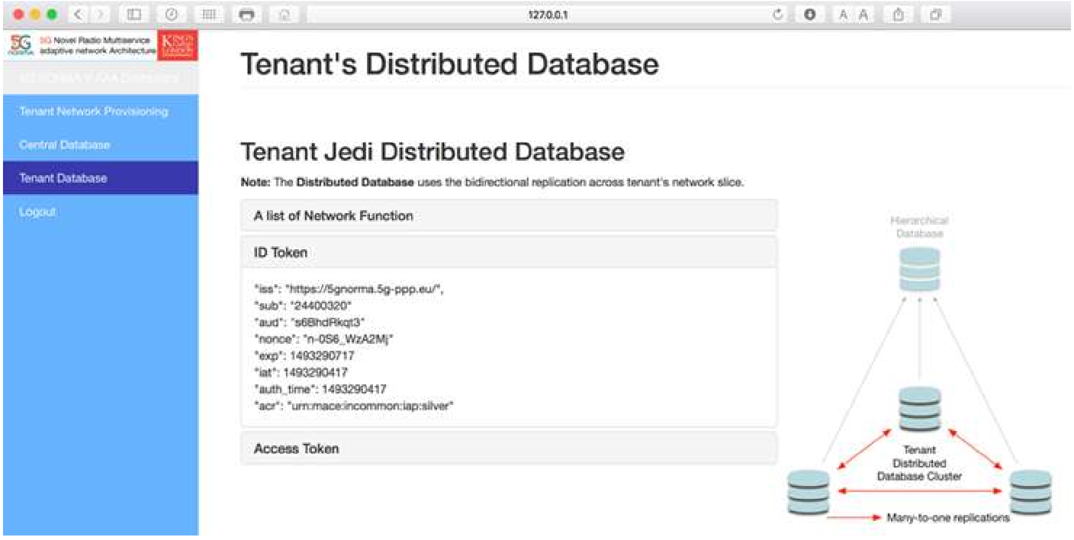


Figure 3-35: Demo 3 Tenant's Network Slice Distributed Database

It is noted that the input to the SDN Controller and feedback of the SDN status are affected by difficulty/unreliability in making the Django web UI and SDN controller threads talk to each other, impacting the data shown in Figure 3-34 and Figure 3-35. Hence, whereas all other fields are OK, the data in the fields related to the SDN controller input and response had to be manually mirrored across the two threads in this instance (e.g., OpenVSwitch port name, OpenVSwitch Bridge name, etc.). While the entire framework is fully developed, including the web UI frontend, the SDN control and all other aspects of the backend (encompassing the database Open vSwitch port instantiation, etc.), this has been necessary merely because of the outstanding challenges of getting the two threads (Django and Ryu) to reliably talk. More work is being undertaken to address this challenge.

3.4 Demo 4. Online Interactive 5G NORMA Business Case Evaluation Tool

3.4.1 Introduction

The online demo presents the economic benefits of a 5G network, which is one of the key principles in the 5G NORMA project. The online interactive tool demonstrates the key outputs and sensitivities of the evaluation cases selected as part of the overall project evaluation framework. The tool shows the economic feasibility of the 5G NORMA architecture by looking at three different evaluation cases which are described in Deliverable D3.1 [5GN-D31].

In brief, the evaluation cases are as follows:

- Evaluation case C.1: Baseline evaluation case – comparison of 5G NORMA with evolved legacy (LTE-A Pro) networks for MBB services;
- Evaluation case C.2: Multi-tenant evaluation case – comparison of single and multi-operator networks for MBB services; and
- Evaluation case C.3: Multi-service evaluation case – comparison of single and multi-service networks.

The demo is developed by using an online web-based interactive dashboard approach. The user can simply use a web browser to check the results.

Note that this demo is not a physical demonstration of certain capabilities such as other demos in this project. However, it is an online tool with certain capabilities where the user can check costs or revenues for the defined evaluation cases. The values provided via the web-based interface are abstracted from the network modelling results developed in WP2 as described in more details in D6.1 [5GN-D61]. The economic validation tool is dependent on the analysis and results of the socio-economic model developed in WP2.

The main criteria are taken from two key 5G NORMA innovations, these are multi-tenancy and multi-service capabilities. These innovations are relevant because they suggest that at least part of the value from 5G will be driven by the potential economies of sharing (e.g. multiplexing gains and economies of scope) in 5G networks and part by the underlying value of the services.

In order to be capable of modelling the 5G NORMA network architectures, the WP2 model has drawn up a list of 5G network elements in collaboration with WP3 of the project. The WP2 model has derived unit cost, capacity and network dimensioning rules for these elements. Some elements will be similar to those already used in current 4G networks such as antennas and base stations. However, the WP2 model was extended to cover other issues such as front-haul, core transmission network transport, edge and cloud servers, software elements, and processing requirements. In addition, revenue side is also derived in WP2 to show the economic benefits of such architecture.

As commented, the results provided in Demo 4 via the web-based interface will be abstracted from the network modelling results developed in WP2. Note that the runtime to obtain the WP2 results is based on an optimised network deployment and will have a long execution time for each case studied. We use key results from the different evaluation cases in WP2 to abstract approximate functional relationships between key attributes that can be used in the real time interactive tool.

3.4.2 Implementation

The web-based demonstrator allows users to explore the economic benefits of the 5G NORMA architecture by modifying the inputs of the tool. This graphical interface gives users the ability to quickly check the benefits of the new architecture, showing the key outputs of a complex cost model through a simple dashboard.

In D6.1 [5GN-D61], we anticipated that the web tool would be developed by using PHP (Hypertext Pre-processor) and HTML (Hypertext Mark-up Language) web-programming languages. However, in the latest release of the demo, this approach was replaced by a simpler one based on Spreadsheetweb.com [SSW], a web service for creating dashboards from spreadsheet files. We consider this new approach is technically simplest and sufficient to implement the demo.

The dashboard is a key prototype with the following functionalities (see Figure 3-36):

- Select input combinations: The user selects the desired inputs through the web-interface (dashboard frontend)
- Interrogate the abstraction data: The model interrogates and selects the suitable data from the database (according to the selected inputs). The data are mainly taken from the outputs of WP2 for the selected scenarios.
- Present results on the web-interface: The results are presented on the web-interface, which is part of the dashboard frontend.

The demo results are initially implemented in form of an excel file where all the functionalities are developed across several spreadsheets. Hence, the model functionality is mainly developed through excel functions and equations. The main data which drives the results in the dashboard are taken from WP2 cost and revenue results, which is part of the database of the model.

Once the user makes the selection, the algorithm in the spreadsheet interprets that and reacts accordingly. There is some intelligence in the dashboard which allows the model to get the user’s selection, interpret it and provide the required outputs.

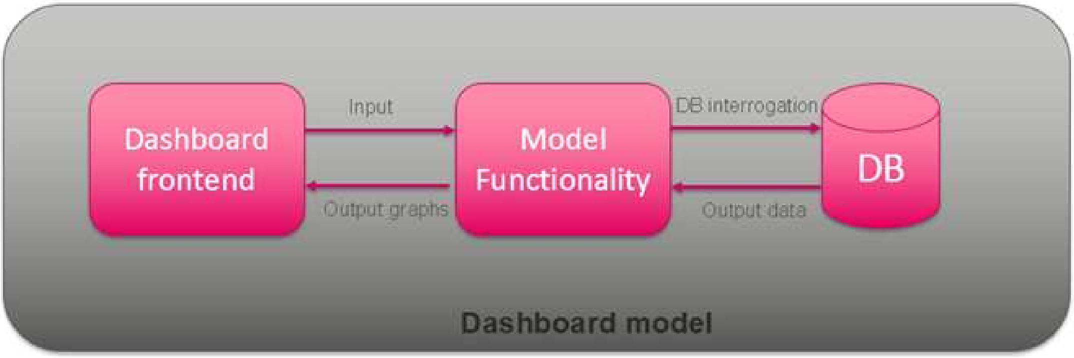


Figure 3-36: Dashboard Model Implementation

Once all the functionalities of the dashboard are implemented and tested in the spreadsheets, the model can be made available online through Spreadsheetweb.com. In brief, this platform converts the model functionality from the local machine to the web, hence making this local based dashboard into an interactive online tool. The process to make such a conversion is shown in Figure 3-37, which includes the following steps:

1. The dashboard model implementation and testing (described in the previous paragraph) are made by the developer.
2. The developer uploads the model to Spreadsheetweb.com. This also involves some manual interaction to adapt the functionality of the specific dashboard.
3. Spreadsheetweb.com converts the model into an interactive tool that is available online.
4. The developer then redirects RW webserver to the online tool.

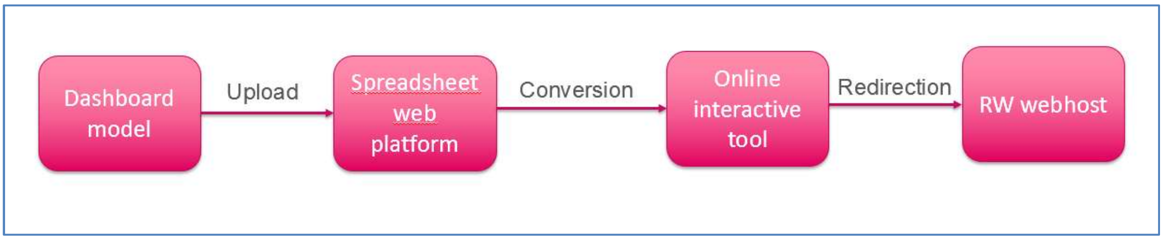


Figure 3-37: Process of the dashboard conversion into an online interactive tool

3.4.3 User Guide

Overall, the online demonstrator allows end users to select a range of inputs such as:

- Traffic Growth;
- Evaluation Case; and
- Service Selection (this is part of evaluation case 3).

Once a user makes a selection of the desired inputs, the outputs are shown in terms of traffic in Mbps/km^2, and indicative figures of the network costs and revenue are shown in GBP. The costs are shown in terms of CAPEX, OPEX and TCO (see outputs section for more details).

End users can access the online dashboard by simply typing in web-browser the following link: http://www.realwireless.eu/main.php, which redirects the user to the main page of the demo.

In the main page, shown in Figure 3-38, the user is able to:

- Get familiar with the simulation area;
- Read the description of the 5G NORMA architecture (CRAN) vs. traditional network (DRAN); and
- Select one of the evaluation cases to browse through (bottom of the page).

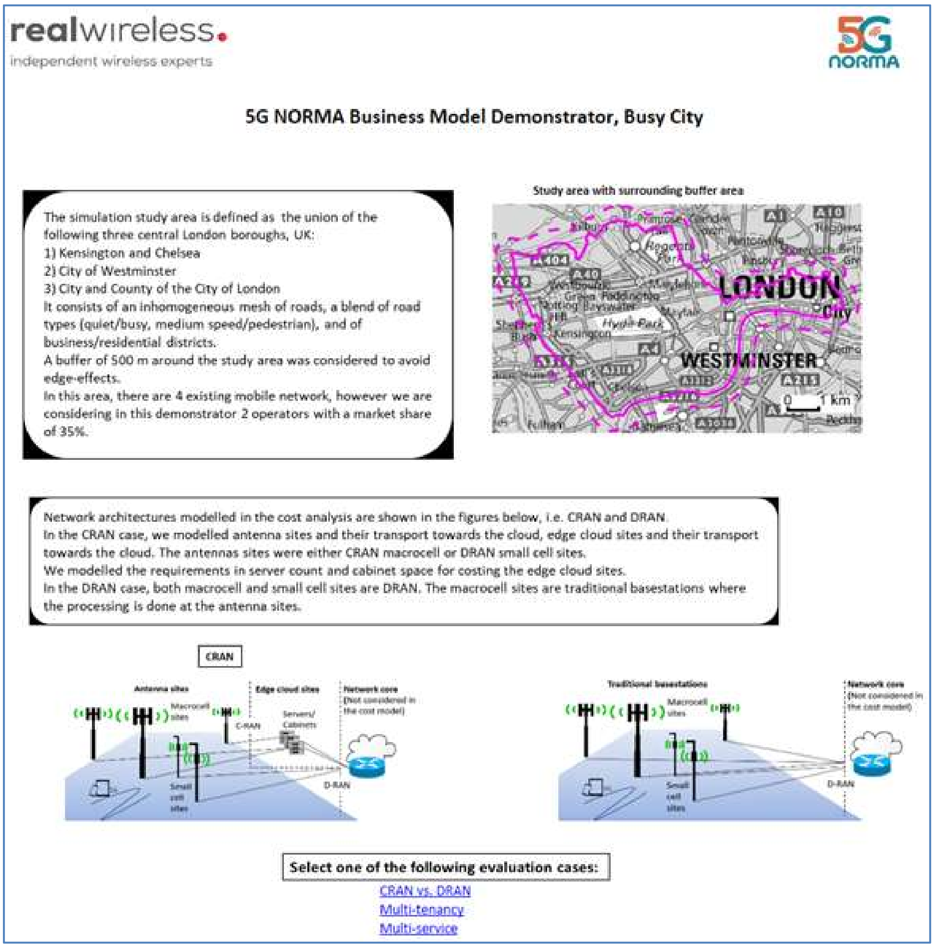


Figure 3-38: Demonstrator Main Page

As we see, from the main page, the user can navigate through the evaluation cases by simply clicking on the desired evaluation case as shown in Figure 3-39.

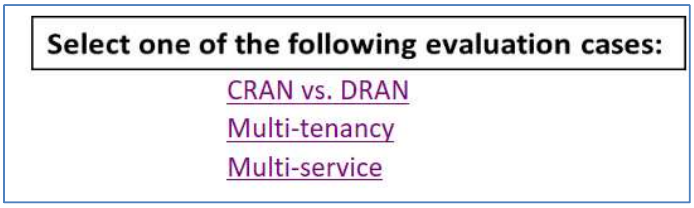


Figure 3-39: Selection of the Desired Evaluation Case

The following outputs are shown in each evaluation case:

- Evaluation case 1: CRAN vs. DRAN cost comparison;
- Evaluation case 2: Multi-tenancy vs. multiple independent networks cost comparison; and
- Evaluation case 3: Multi-service cost and revenue results.

Once the user selects the evaluation case from the main page, he can then select the desired input parameters in the evaluation case webpage. For evaluation cases 1 and 2, the user can select as input the traffic CAGR for the study period as shown in Figure 3-40.

Figure 3-40: Input Selections in Evaluation Cases 1 and 2

However, in evaluation case 3, the user can select the desired services as shown in Figure 3-41.

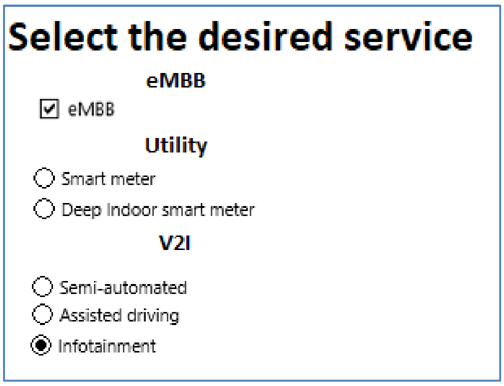


Figure 3-41: Service Selection in Evaluation Case 3

A description of the outputs is provided in Section 4 (Results and Verification) below.

4 Results and Verification

In the following subsections, we analyse the WP6 results reviewing the main KPIs for each demo; afterwards, we summarize the main lessons learned.

4.1 KPIs

The KPIs analysed in this section are structured as follows:

- Performance KPIs, including what we refer to as "Hard KPIs", with numerical objective values for parameters such as latency, throughput, or orchestration times.
- Verification KPIs, analysing the degree of fulfilment of the main objectives defined for the WP6.
- Socio-economic findings, coming mainly from Demo 4.
- Contribution KPIs, analysing how we processed/generated feedback from/to other WPs, and also how our work impacted external entities.

4.1.1 Performance KPIs

Performance KPIs are mainly coming from Demos 1 and 2 as described below. Demo 1 reports values about average session throughputs and E2E latency (measured on the hardware part) as well as average downlink throughput and average delay (measured on the software part). On the other hand, Demo 2 reports bounded latency and bandwidth values in the RAN part as well as relevant figures about orchestration times in the cloud infrastructure (NS deployment times and VNF migration times).

4.1.1.1 Demo 1. Hardware Part.

The hard KPIs that have been used in this demo to estimate the system performance are as defined in [5GN-D61]:

- The average session throughputs,
- The E2E latency.

In the following it is described how these two KPIs are affected for both the services in the different steps of the demonstrator's hardware part.

At the beginning, both services are configured with the SP-GWs in the edge cloud. The latency measured is depicted in Figure 4-1, for the eMBB service (blue line) and URLLC (orange line) respectively.

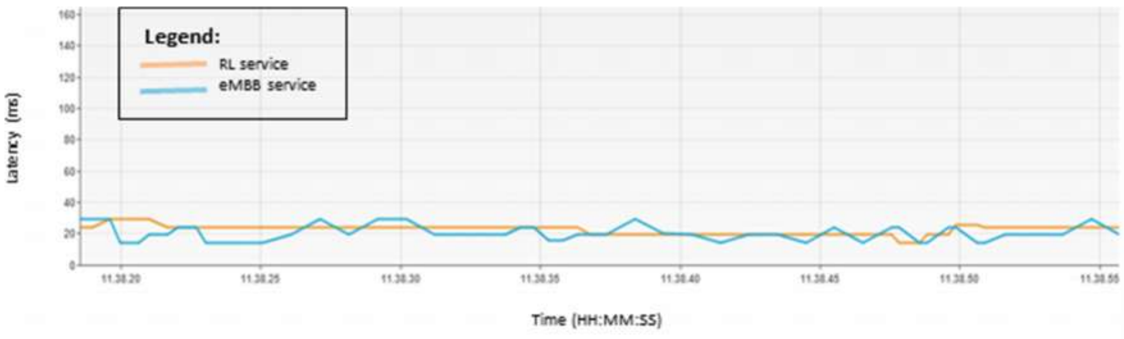


Figure 4-1: Latency of both users at the setup phase

As we see the E2E latency is kept low (below 30 ms) and stable for both services.

After the relocation of the GW of the eMBB user, the latency of this service is increased. On the other hand the RL service is not affected by this relocation and the latency is still the same. In the following Figure 4-2, the latency of both services is depicted.

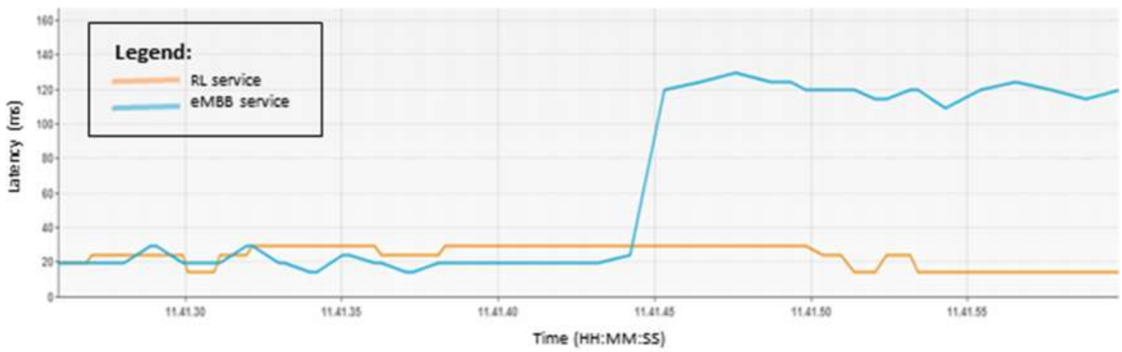


Figure 4-2: Latency after triggering the services

As we see, in this case the latency for the eMBB service increases after the relocation, but then it remains stable.

The demonstration evolves with the addition of inter-cell interference, which affects the quality of the eMBB service. On the other hand, the RL QoS is not reduced. The following steps are performed:

- The SDM-C receives the feedback from the eNB, so it is aware of the QoS reduction in the eMBB service.
- The SDM-C executes a service-aware scheduling algorithm, which is used to reconfigure the whole scheduling policy.
- The corresponding control command is sent to the eNB through the SBI.
- Then the eNB applies the new scheduling policy restoring the quality of the eMBB service.

During the SDM-C policy decision, the latency in the RL service is not affected but is kept low and stable.

4.1.1.2 Demo 1. Software Part.

In the software part of the demo, the two main evaluated KPIs are:

- the average downlink throughput, and
- the average delay.

The demo presents the changes of these KPIs for three cases:

1) when all network functions are in the edge cloud,
2) when they are in the central cloud,
3) when the service-aware algorithm is making the placement decision.

For the first case, the core network functionality is also brought close to the eNBs, and the E2E latency for the users can be reduced considerably. This effect is highlighted in Figure 4-3. Here it can be seen that the throughput of both UE groups are considerably lower because the eNBs in edge cloud configuration cannot efficiently coordinate the eNBs or apply centralised interference mitigation to improve the network throughput. However, the E2E latency is relatively improved in this setting because of the edge cloud configuration. Therefore, this setting only satisfies the service requirements for the MTC users.

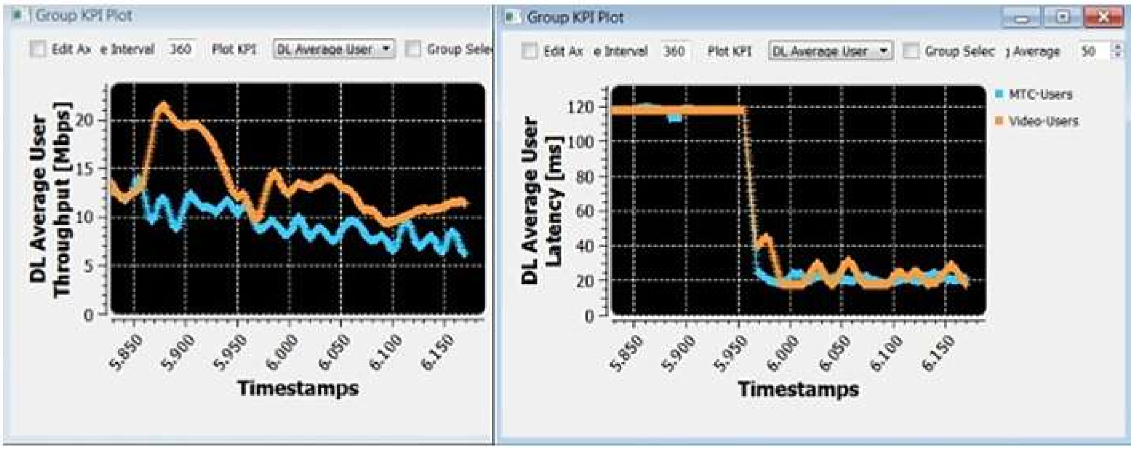


Figure 4-3: Software demo – placement of NFs in the edge cloud

For the case when all network function elements of eNBs are shifted to the central cloud configuration, SDM-C enables the eNBs to efficiently coordinate among each other to reduce interference and hence improve throughput. As a result, it can be seen from Figure 4-4 that the throughput is improved for both groups of users.

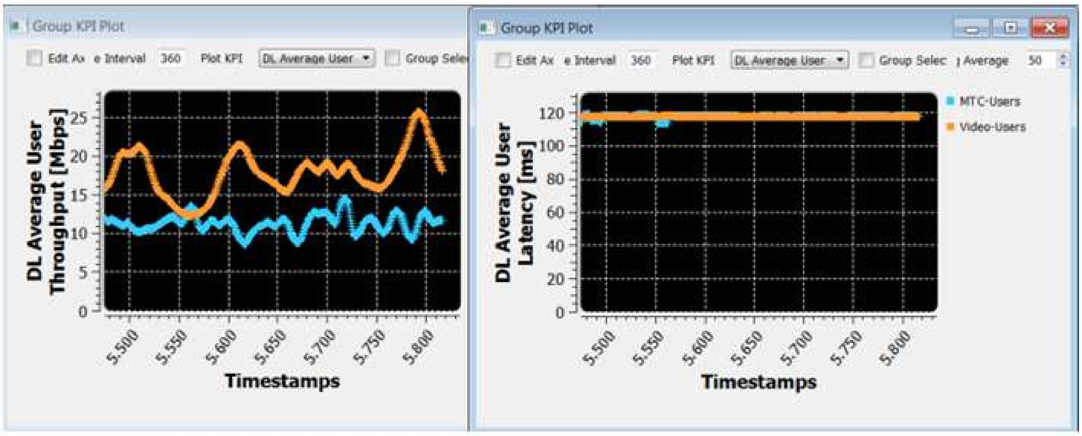


Figure 4-4: Software demo – placement of NFs in the central cloud

In central cloud configuration, on the one hand, an improvement in throughput is observed. However, on the other hand, the latency becomes worse. This high latency, as visible from the plots in Figure 4-4, is not suitable for MTC users since for such users the latency must be lower to guarantee proper functionality of the service. This case shows that if all eNBs are moved to central cloud, the service requirements for MTC users cannot be satisfied. Therefore, an intelligent configuration based on service-aware decision logic shall be considered to satisfy both types of services requirements.

The proposed intelligent service-aware SDM-C logic is highlighted in Figure 4-5. In this scheme, SDM-C can intelligently decide to place network functional elements of each eNBs into central- or edge cloud. The SDM-C is capable of using different system parameters to place functional elements of the eNBs into central- or edge cloud, e.g. it can use the load on eNBs or the type of dominant service running. In the demo presented, SDM-C makes use of the dominant service-type to make a decision on which eNBs' functional elements have to be placed in the central cloud and which eNBs can be moved to the edge cloud.

For the case of the scenario presented in Figure 4-5, the distribution of UEs in the simulation is such that the lower right corner of the simulated area is occupied mostly by video streaming UEs, which are connected to the two eNBs in the lower right segment of the simulated area. These two eNBs have a higher number of video streaming UEs. The remaining eNBs in the simulation have a more significant number of MTC users connected to them.

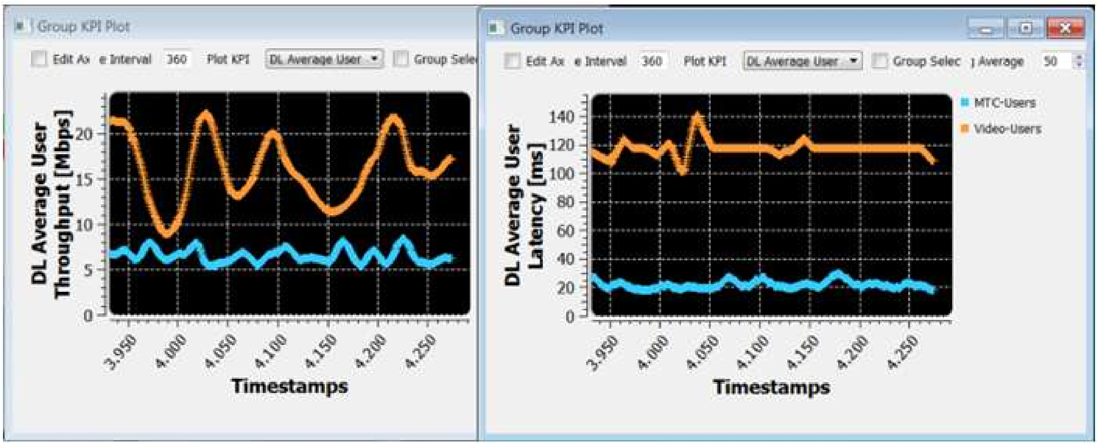


Figure 4-5: Software demo – Algorithm Making the Placement Decision

The SDM-C can detect the demand for different services in different cells based on feedback messages it receives from the connected eNBs. In the simulated scenario, SDM-C detects that the two cells at the bottom right can best serve the requirements of their users if their functional elements are placed in the central cloud. Therefore, it places these network functionalities of these eNBs in the central cloud, highlighted with red in the figure. On the other hand, the remaining cells in the scenario have users with low-latency requirement. Using the service-aware decision-logic, the SDM-C places the remaining eNBs in edge cloud configuration, which is highlighted in Figure 4-5.

As a result of such service-aware placement of functionalities into the central and edge cloud configuration, the video streaming UEs can achieve the desired higher throughput due to better coordination among eNBs. The throughput curve in Figure 4-5 presents the improvement in the throughput while the E2E latency of MTC users can be improved at the same time as it is shown by the latency curve for MTC users in the same figure, Figure 4-5. Therefore, the introduction of an intelligent SDM-C improves the overall system performance and can satisfy a mix of different services in the network.

Virtual Cell Extension

The throughput is also the focus of the virtual cell extension. The demo presents the total network throughput, the uplink and downlink throughput, and their changes as the virtual cells form and how the cell-edge threshold is adopted (i.e., the threshold, which determines the size of virtual cells). The change in the total network throughput (the summation of uplink and downlink) as the result of changing the cell edge threshold manually and the size of the virtual cells is apparent in Figure 4-6.

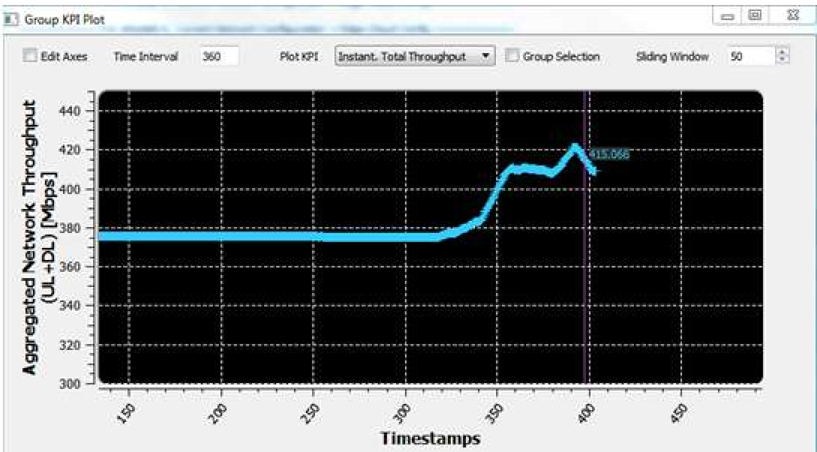


Figure 4-6: The total network throughput increases after changing the cell edge threshold.

4.1.1.3 Demo 2. RAN Deployment KPIs.

Relevant stakeholders such as 5G-PPP and NGMN fix very high standards for the KPIs regarding pure channel performance such as throughput, latency and capacity. The relevant KPIs for Demo 2 are throughput and latency. Clearly, the setup employed by Demo 2 will not provide significant updates in terms of enhanced bandwidth or sub-millisecond latency: the employed software is based on an SDR platform that is not optimized for such requirements and the employed software platform is a modification of the LTE implementation. Therefore, the latency and bandwidth requirements listed here are not to be considered a step forward towards enhanced RAN mechanisms (Demo 1 is showcasing those concepts), but rather a proof of concept of the possibility of a service-specific orchestration capable of reconfiguring itself according to the changing conditions.

Bounded latency

We tested this configuration in a single slice static scenario for the mobile broadband network case. The data plane in this scenario consists of a laptop running the srsUE software, another PC running the eNB and a virtualized core running in the same hardware infrastructure. Here we timestamp the requests sent from the UE to the AR server, and generate two requests per second.

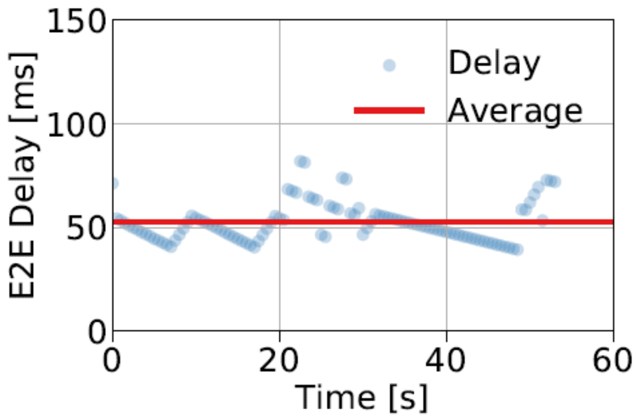


Figure 4-7: Latency for the RAN setup

The results show the instantaneous end-to-end latency of the requests (depicted as blue dots). The red line shows the average value over the entire experiment, instead. Results show that the latency remains almost constant around the average value, with a saw tooth behaviour that we attribute to the scheduler influence on the channel access pattern. Nevertheless, the delay is bounded to values that, although not ideal (there are delays introduced by the SDR card, the software processing of the encoded PRBs), the latency values are sufficiently small to be able to showcase our scenario.

High bandwidth

Enhanced PHY techniques such as MIMO are unfortunately not available with the SDR srsLTE approach. Therefore, the total throughput available for a single user in the system is limited by the availability of a single stream and, most importantly, to the number of Physical Resource Blocks (PRB) used there. As a software implementation, these parameters are easily configurable in both srsUE and srsENB. Of course, a larger number of Resource Blocks (RB) means a larger bandwidth and, in turn, a higher throughput for the users. On the other hand, increasing too much the number of RBs in the system makes it prone to channel errors (larger bands increase the possibility of interference on a general purpose board such as the USRP we use) and to frequent reconnections. After an extensive trial campaign, we finally chose the value of 75 RB that provides enough bandwidth for our purposes, while guaranteeing a stable environment for both the testing and the deployment phase.

In the following, we show the obtained results for the throughput experiment we performed in the enhanced mobile broadband configuration. In order to get full throughput results, we used the Iperf tool running in the same location as the video server used in the demo setup. The sessions are 10 seconds long and the values are averaged over several repetitions of the experiment.

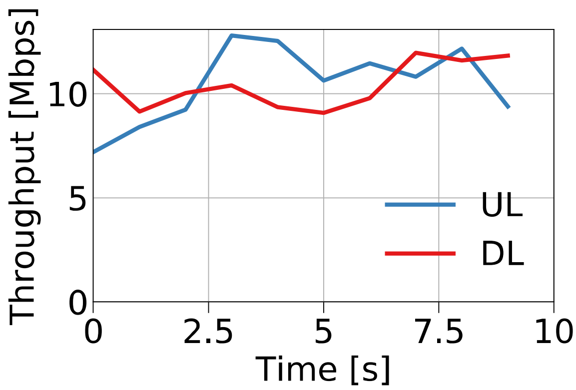


Figure 4-8: Throughput for the Mobile Broadband setup

As already can be seen for the latency results, the throughput is lower than the envisioned new generation 5G Networks. Nevertheless, the combined stability level and the average throughput obtained in several runs is enough for the purposes of the demo, as the video occupies 3Mbps in downlink only.

As stated above, the major KPIs for the RAN deployment are soft KPIs, as they are the enabler for the network sliced architecture defined in WP3 and the orchestration algorithms designed in the framework of WP5.

4.1.1.4 Demo 2. Cloud Infrastructure KPIs

Two KPIs have been considered relevant in this case:

- Network Slice deployment time
- Virtual Network Function migration time

For the NS deployment, we’ve obtained values in the order of tens of seconds, e.g., Figure 4-9 shows a NS creation process containing four VNFs (HSS, MEE, SP-GW and eNB simulator) performed by the Atos VNF-Manager component, which is completed in about 83 seconds:

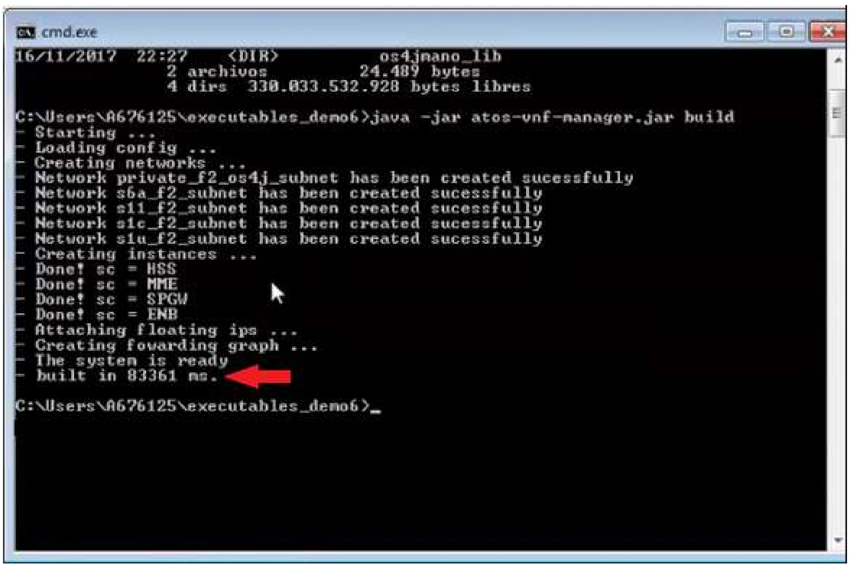


Figure 4-9: RL Slice Creation Time

We consider this is a quite good result if we compare with the time it would be necessary to deploy the same service using legacy physical network functions, which could be in the order of days or even weeks.

On the other hand, regarding the VNFs' migration time, as explained in Section 3.2.6.2, different experiments have been performed moving the SP-GW VNF. This is a stateful VNF, which is used in both slices (MBB and RL). As explained in that section, the migration time is in the range of a few milliseconds using the shared storage live migration option.

4.1.2 Verification KPIs

In this section, we explore the verification KPIs, i.e., those non-quantitative KPIs which are related to some important WP6 objectives. Few of them are common for all four demos, but most of them are relevant for Demos 1, 2 and 3 as we see in the following sections.

4.1.2.1 Common KPIs

They are the following:

- Develop both, a hardware and software based demonstrator.
- Serve as show platform to exhibit the project in public conferences and events.

Regarding the 1st KPI, of course all the demos have an underlying software and hardware infrastructure. This is most relevant in Demos 1, 2 and 3 (see respective Sections 3.1, 3.2 and 3.3 for each case). For Demo 4, although most of the development is based on software, a minimum HW infrastructure is also necessary to access and execute the web service.

Regarding the 2nd KPI, all the demos have been designed with a small form factor and modular construction to be portable and easy to show in experimental reduced scale environments such as those usually available in relevant events and conferences. Anyway, although meeting this requirement, all demos have been designed to faithfully show their main concept. Proofs of this are the different conferences at which the demos have already been presented (see Section 5).

4.1.2.2 Demo 1 KPIs.

The main qualitative KPIs regarding Demo 1 are related to the HW part. They are mostly related to the new design proposal based on the SDM-C concept defined in the 5G NORMA architecture. This part wants to show also the gains produced by the SDM-C over legacy network architectures such as LTE. We consider that the SDM-C concept defined in the 5G NORMA architecture gives a big value to the architecture, also on a legacy network, allowing for self-adaptation to the demands and requirements of frequently varying network conditions by means of network programmability. In the implemented example, the SDM-C and eNB can communicate by the defined protocol, facilitating the control which can be implemented with reasonable efforts.

4.1.2.3 Demo 2 KPIs.

In Demo 2, qualitative KPIs are mainly split in two parts:

- KPIs applicable to the RAN deployment, and
- KPIs applicable to the cloud infrastructure.

4.1.2.3.1 KPIs applicable to the RAN deployment

Current standardization in 3GPP is introducing the concept of network slicing in the future 5G network elements, initially as enhanced MAC algorithms only. The 5G NORMA vision is to bring this new concept into the architecture at all levels: management and orchestration as well as control and data layer.

Of course, current Open Source initiatives for the mobile network elements, such as SRS and OAI Flexible Service Orchestration are necessarily not including these items into their releases, as their goal is to provide an open experimentation platform for LTE. There are indeed initiatives to bring network slicing to OpenAirInterface, such as Flexran [FOU-16], but they are focused to RAN slicing only, while our goal is to provide an end-to-end architecture capable of network slicing at all levels including orchestration. Therefore, we had to modify the current Open Source software in order to meet our requirements for a flexible service-aware orchestration.

Flexible and service-aware orchestration

Network slicing on the data and control layer comes with the introduction of a data multiplexer and de-multiplexer, as already discussed in the context of WP4. Therefore, we implemented a modified version of the SRS RAN protocol stack to enable this feature, as described in Section 3.2.6.1. This is a fundamental enabler for the flexible service orchestration as we envision in 5G NORMA. That is, we seek for a full separation of dedicated resources in a network slice; therefore we needed to provide the core network with "handles" to the RAN to manage the flows belonging to the different services. Without the modifications to the RAN stack performed for Demo 2 (not available in any Open Source framework), the Demo 2 narrative would not have been possible.

End-to-end network slicing

The 5G NORMA innovations for multi-tenancy and network slicing are designed to provide the maximum flexibility to the tenant that can i) instantiate especially tailored versions of each dedicated VNF it controls, and even ii) have its own dedicated MANO stack with its own per slice policies. The software developed for Demo 2 fully supports this spirit by implementing the RAN slicing Option 3 (slice-aware shared RAN) of the ones defined in WP4. Up to that point, the two network slices are completely separated allowing for a full tenant customization. That is, QoE/QoS-aware control and scheduling of the core functions are enabled by the implementation of the network slicing principle in the RAN.

4.1.2.3.2 KPIs applicable to the Cloud Infrastructure

This demo covers by itself most of the key objectives defined in the project proposal for WP6. They are the following:

KPI 1.

Design, implement and validate a real-world system proof of concept to demonstrate the feasibility of the concepts of 5G NORMA and the potential benefits of the architecture.

As previously explained, we've designed and implemented a management and orchestration cloud architecture according to the main principles defined in the 5G NORMA WP3. Hence, we consider this objective has been fulfilled. The architectural principles described in WP3 have been carefully evaluated, especially those regarding the implementation of the MANO layer. In fact, from WP6, and specially derived from the work performed for this demo, we've actively collaborated in the definition of these architectural principles in the scope of WP3. Since the architectural principles described in WP3 are mainly based on the ETSI NFV MANO framework (WP3 defines specific ETSI NFV MANO stacks for each slice in the network), we consider there is a very good alignment between the demo and the architectural principles defined in WP3.

Also, because of the usage of a new Java-based implementation for the NFV management and orchestration blocks, we consider this demo has gone beyond the initial intention of this KPI by exploring some of the 5G NORMA specific architectural elements described in WP3 and implementing an initial version of the SDM-O and the Inter-Slice Resource Broker, which are specific 5G NORMA architectural blocks introduced by WP3.

KPI 2.

Provide the necessary and sufficient tangible evidence of the feasibility, applicability, effectiveness, efficiency and profitability, as well as the limits, of the architecture and some of its key concepts.

As described in Section 2.1.1.1, the key concept on which we focus in this demo is network slicing, especially from the NFV management and orchestration perspective as it is defined in the 5G NORMA architecture. To evaluate this KPI specifically in the context of this demo we've studied the feasibility, applicability, effectiveness, efficiency and limits of the ETSI NFV MANO framework, since it is the basis of the architecture defined in WP3[10]. In the following, we analyse the different aspects:

a) Feasibility

The NFV MANO architectural framework identifies the following main functional blocks: NFV Orchestrator (NFVO), VNF Manager (VNFM) and Virtualized Infrastructure Manager (VIM)[11]. In order to evaluate this KPI, the initial approach was to use different already existing Open Source out-of-the-box solutions implemented according to the ETSI NFV MANO specification. However, different problems were found while following this approach (see Section 3.2.2 and Section 4.2 below), hnce, it was finally decided to address this by implementing of a hybrid solution consisting of a well-known open source software product to perform the VIM function [OPSTK] combined with ad-hoc programmatic implementations of the VNFM and the NFVO blocks.

The final result is not a general purpose and complete ETSI NFV MANO implementation, but a specific one to address the particular management and orchestration requirements in this demo. Although the VNF Manager and NFVO have a limited scope (adapted to the demo requirements) we do not see unattainable that, with enough time and resources, its functionality could be extended to cover all the aspects defined in the ETSI specification.

Finally, we consider that having a good ETSI NFV MANO implementation depends to a large extent on having a good VIM implementation, providing also all its functionalities visible to the VNF Manager and NFVO through the interfaces defined for it (*Vi-Vnfm* and *Or-Vi*). This is because, ultimately, the VIM is the component responsible for controlling and managing the underlying NFVI compute, storage and network resources, while VNF Manager and NFVO are just logical abstractions implemented on top of it in order to provide a more high-level view (in the form of complete network services, forwarding graphs, etc.). For instance, if the VIM is able to implement functions such as the live migration of network functions or scaling of the deployed VNFs, then the VNFM and the NFVO could be developed to make use of those capabilities, otherwise not. This implies that the feasibility of the ETSI NFV MANO architecture (the object of this KPI) is not challenging itself, as long as a truly ETSI-compliant VIM is provided. Since many vendors already offer different VIM implementations (e.g., VMWare vSphere, Openstack or OpenVIM) we don't see an overwhelming difficulty in implementing the complete set of functions defined by ETSI.

b) Applicability

In order to demonstrate the ETSI NFV MANO implementation applicability, we should verify that it can be actually used for the specific purposes of this demo. The MANO platform should

---

[10] We consider *profitability* to be out of scope for this specific Demo 2 (that topic is addressed in Demo 4 instead).

[11] Of course, it also specifies other things related to these functional blocks: reference points, catalogues and other external blocks that could communicate with these three components, but at the end, these are the three main functional elements from which everything else is derived.

be able to manage and orchestrate the VNFs provided by both partners participating in this demo (UC3M and Atos).

We've used the ETSI NFV MANO platform to configure and deploy the different network functions developed by both, UC3M and Atos. Based on plain text descriptors in JSON format, our VNF Manager implementation is able to automatically perform the necessary life-cycle management functions: instantiating, updating, and terminating VNFs[12]. In our tests, the deployment time is in the range of tens of seconds.

c) Effectiveness and Efficiency

According to the English Language Oxford Dictionary, *effectiveness* is "the degree to which something is successful in producing a desired result". On the other hand, the term *efficient* is associated to "achieving maximum productivity with minimum wasted effort or expense". In a less formal language, it is usually told that *effectiveness* means "doing the right thing", while *efficiency* means "doing the thing right".

In our case, applied to our implementation of the cloud infrastructure for the demo, we can consider it is both: effective and efficient. It is effective because it has enabled us to produce the desired results. The remaining KPIs in this section (and the descriptions distributed in the other sections of this document devoted to cloud infrastructure) are proof of that. It is also efficient because the resources used to build up the cloud infrastructure for the demo has been adjusted to the demo requirements themselves, i.e., although, as explained before, the initial approach was to try to use a general purpose out-of-the-box NFV orchestrator, the focus was later replaced by a new approach based on developing a programmatic solution covering just the specific requirements for the demo. In this sense, this new approach is more effective, since the resources being used are better adapted to the actual requirements in this demo. A proof of this is that in a short time, we've been able to develop the basic functionality for the demo.

Although we know that this solution is not a complete general purpose ETSI NFV MANO implementation, we consider this lack of generality does not exclude that, if necessary, this ad-hoc solution could be extended to build the basis for a more complete general purpose orchestrator aligned with the overall ETSI architecture. In fact, one of the general purpose orchestrators we were considering for this demo (OpenBaton) has been developed using the same baseline technology as the one we're using here (i.e., the Java programming language and the OpenStack4J library).

d) Limits

The main limitations we have seen in our ETSI NFV MANO implementation are the following:

i. The booting time of the deployed VNFs. With a bit of fine-tuning such using a minimal version of the Ubuntu 16.04 operating system and starting-up only those processes strictly necessary to execute the VNF in each case, we have achieved booting times in the range of a few seconds (less than 10s typically), which is sufficient for our demo. However, this could be obviously a drawback when a precise orchestration of the VNFs is required (they should be required to actually start-up just in time when they are needed). In this sense, we consider that alternate approaches based in other technologies different to the typical virtual machines could help (e.g., Containers [KT-14] or Unikernels [AM-13]).

ii. Regarding the mobility of VNFs we have seen the following drawbacks[13]:

[12] VNFs scaling has not been implemented, since it is out of scope for this demo; anyway, this functionality could be easily added since it is well supported by the VIM and the OpenStack4J library.

[13] See KPI 4 below for additional details.

- In our implementation, based on the OpenStack live-migration functionality, this can be achieved only when source and destination are compute nodes which are managed by the same VIM. When this is not the case, the live-migration function cannot be performed. This can be important in a typical 5G telco environment, where edge and central clouds will probably be different network clouds with different MANO components. We consider that, beyond this demo, different migration approaches should be investigated.
- We have also seen that there is a strict requirement about the hardware that has to be used: source and destination nodes should be based on the same HW architecture. Although this is not a strong limitation, we consider it is something that should be considered. Perhaps, using different migration strategies could help to overcome this drawback.
- To achieve a good performance for this functionality, it has been necessary to implement it using the so-called ‘shared storage live-migration’ strategy. This requires a common synchronized shared storage accessible by both, source and destination. Hence, although the VNF is logically re-located in a different placement, it still requires access to the common storage node (see Section 3.2.6.2 about VNF mobility).

<u>KPI 3.</u>

Show some key principles of the future 5G architecture, such as the service-aware adaptive allocation of functions to different network nodes as well as some 5G mobility concepts.

Demo 2 is mostly related to orchestration aspects of network slicing and it is envisioned to be the main showcase of WP3 and WP5 novel concepts. However, some of the considered challenges are in the field of WP4, especially in the field of allocating the available NFs to the best cloud infrastructure location. As we’ve seen, in our case this has been addressed by exploring the possibilities of the live-migration functionality provided by the VIM (OpenStack). Details are provided in Section 3.2.6.2 (VNF Mobility).

Also, the integration of PNFs (such as the eNB) into the 5G NORMA orchestration architecture is one of the final outcomes of the demonstrator.

4.1.2.4 Demo 3 KPIs

The main KPIs in Demo 3 are based on the verification of concepts. These concepts are:

- Verification of the correct maintenance and storage of tenant information in a database;
- Verification of tenant data isolation in the edge network, and tenant data consolidation at the operator;
- Access to tenant data at the operator’s database; and
- Use of tokenization to provide security (identity and access rights).

We have observed the realisation of these concepts through the operation of the demo and its outputs covered in Section 3.3. It is covered there because the associated content (e.g., screen shots) is also heavily-related to the development of the demo and particularly the software (GUI).

This demo also addresses prominent objectives, defined in the project proposal for WP6, which are considered in the following.

<u>KPI 1.</u>

Design, implement and validate a real-world system proof of concept to demonstrate the feasibility of the concepts of 5G NORMA and the potential benefits of the architecture.

A real-world implementation has been developed, proving the concept of hierarchical V-AAA and showing the benefits of the idea in terms of, e.g., being able to maintain tenants’ information and perform AAA in general in the local domain. Feasibility has been demonstrated in terms of the proven operation of the concept even on extremely basic commodity hardware.

KPI 2.

Provide the necessary and sufficient tangible evidence of the feasibility, applicability, effectiveness, efficiency and profitability, as well as the limits, of the architecture and some of its key concepts.

a) Feasibility

The concept is extremely light-weight with respect to processing, signalling and other load aspects, shown by its operation on RPIs. The underlying assumptions such as on structure, the entities involved, and other aspects, are compliant with the general understanding of network slicing and how it might be utilised in future networks. The concept is also compatible with the interests of the stakeholders involved (e.g., tenants, operators). The concept is therefore feasible.

b) Applicability

All of the capacities utilised, in hardware and in supporting software, are standard and widely available. The implementation of the concept on light-weight computational elements (the RPIs) illustrates that it is compatible with even the most challenged network slices in terms of available resources, or alternatively is compatible with running of a very large number of network slides or tenants on the operator’s available resources. The baseline assumptions of the demo are intentionally so generic that it can operate in all envisioned deployment scenarios.

c) Effectiveness, Efficiency and Profitability

Again, the implementation on very basic computational elements demonstrates efficiency. The signalling exchanges (e.g., in the storyline of Figure 3-25) for the demo are reduced to the minimum necessary to implement the concept. Moreover, the demo is effective in terms of achieving its core objectives, as reflected in Section 3.3.

Cost of the demo is minimal as it is based on extremely low-cost equipment. The monitor, keyboard and mouse are readily available office equipment—noting that remote operation is also possible. However, the concept is intended to be implemented on pre-existing hardware (computational and networking elements), which by necessity must be present for the operator to have a (virtualised) network, whereby the running of it on RPIs and the other involved hardware is just for demonstration purpose and simplified portability. Hence, the financial outlay for the operator of running the concept is negligible. Further, the profit of being able to viably perform AAA in hierarchical contexts that transpire in network slicing, and maintain appropriate security in running the network and slices thereof in 5G, is immense. Hence, we assess the profitability of the concept as very high. Nevertheless, we acknowledge that limited operational costs might apply, such as human involvement in the administration of the concept (e.g., creation or verification of user accounts, maintaining the database including back-ups, etc.). But these are not unlike human input that is already required in other aspects of running networks, and could be generally shared and implemented (notably, remotely) by already-present human resources.

d) Limits

While having benefits in terms of ease and robustness of development, the modular design and implementation of the demo through the usage of separate elements (web UI engine, database engine, SDN controller software, etc.) itself introduces some challenges. Such an approach, while still being efficient as described above, is nowhere near *as efficient* as a dedicated fully-integrated application would be. However, noting the amount of work that has gone in to the creation of the utilised tools such as Django, CoachDB and Project Fauxton by large teams of people, the creation of such a fully-integrated tool with its own dedicated database, user interface, and SDN control would be an immense effort.

KPI 3.

Show some key principles of the future 5G architecture, such as the service-aware adaptive allocation of functions to different network nodes as well as some 5G mobility concepts.

The demo has achieved this KPI in terms of showing the key concept of distributed hierarchical security necessary to realise operation in the local domain of a tenant or network slice, while maintaining the separation of tenants, and the ability to reflect information to the higher-level mobile network operator with responsibility for maintaining the infrastructure. It has generally shown that security is viable in 5G through the developed approach.

4.1.3 Socio-Economic Findings

The main results from this demo are those generated from the demo itself, i.e., the key outputs that are shown in the demo including costs and revenues. A complete analysis of these results will be provided in the context of the WP2 [5GN-D23]. Anyway, we include a brief summary here.

Regarding the cost, it can be summarised as capital expenditure (CAPEX), operating expenditure (OPEX) and total cost of ownership (TCO) for the study period 2020-2030. In addition, detailed cost elements of CAPEX and OPEX are also provided. Regarding service revenues, they are illustrated in the web-interface in evaluation case 3.

The demonstrator includes a main page where users can read more details about the scenario analysed and the study area. Users can also select the desired evaluation case to browse through (see section 3.4 for more details).

The results for each evaluation case are shown in a separate web page.

- Evaluation case 1: CRAN vs DRAN cost results;
- Evaluation case 2: Multi-tenancy analysis; and
- Evaluation case 3: Multi-service analysis.

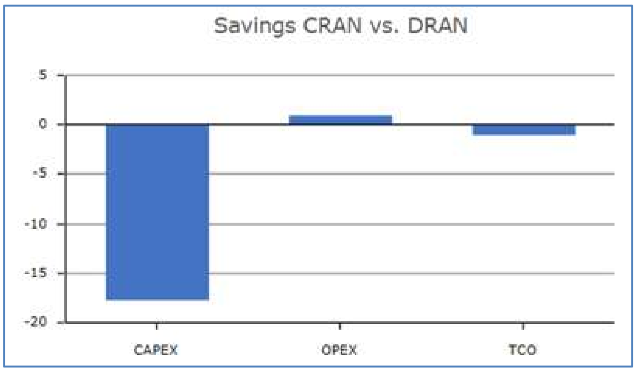


Figure 4-10: Cost saving outputs

In particular the key outputs from demo 4 which show the economic benefits of the 5G NORMA innovations can be summarized as follows:

Evaluation case 1: Results of DRAN vs. flexible CRAN are shown in terms of cost benefits. This mainly shows the comparable costs of CRAN vs. DRAN architecture in terms of CAPEX, OPEX and TCO as shown in the Figure 4-10. In addition detailed CAPEX and OPEX cost elements for each architecture (CRAN and DRAN) are also shown. Cost elements include: site civil works and acquisition, RF front end, labour, transport costs and others (see Figure 4-11).

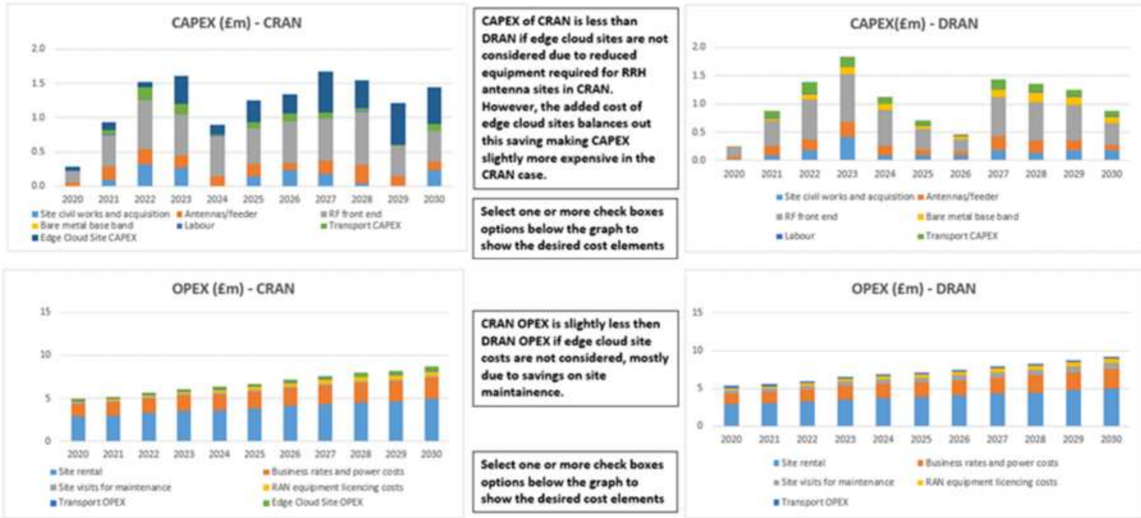


Figure 4-11: Example of Details Cost Element in Evaluation Case 1

Evaluation case 2: Benefits in terms of cost saving when multi-tenancy is applied. This mainly shows the cost saving in terms of CAPEX, OPEX and TCO. In addition, CAPEX and OPEX cost elements are also shown.

Evaluation case 3: Benefits of multi-service support via network slicing in 5G NORMA. which are two-fold. First, limited cost penalty to the network per service (added due to the 5G NORMA architecture providing multiple services from one shared platform) vs. multiple single service legacy networks; second, additional revenue per service added.

In this Evaluation Case 3, the following services are available for consideration under the multi-service case:

- eMBB: Expected to provide substantially higher throughput than today at lower latency
- Utility: Includes smart meters and deep smart meters
- V2I: Includes semi-automated driving, assisted driving and infotainment

In Figure 4-12 the traffic density (Mbps/km^2) in the study area is shown when eMBB and infotainment are taken into consideration.

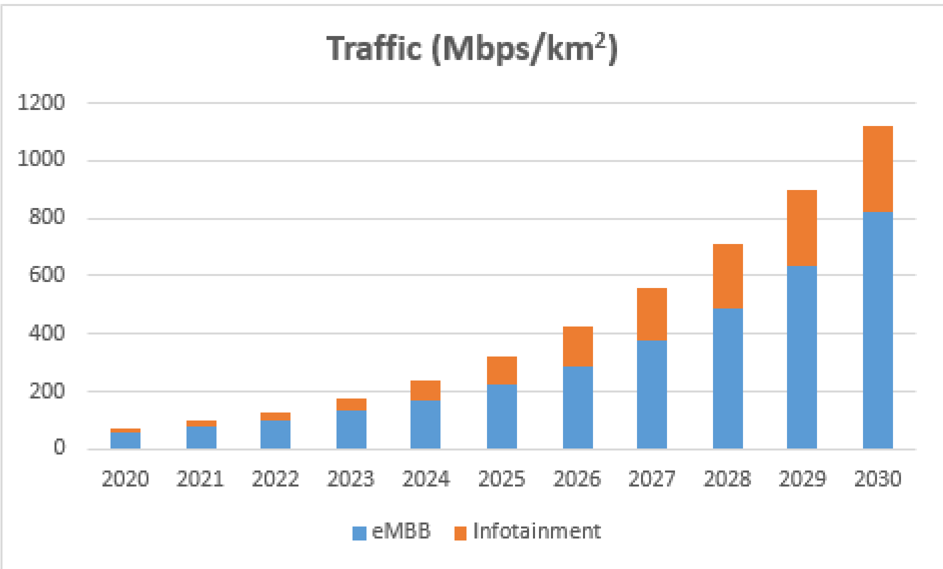


Figure 4-12: Example of traffic density in the study area

4.1.4 Contribution KPIs

In this section, we describe how we've integrated the relevant information coming from other WPs, and also, how we've provided relevant feedback to these other WPs from WP6. We also provide information about how we've contributed from demos to the Open Source community and other 5G-PPP program activities, as well as other dissemination activities.

4.1.4.1 Validation of Requirements and KPIs from WP2

One of the main objectives for WP6 is to provide the architecture's proof-of-concept, validating the technical requirements and KPIs coming from WP2.

In WP2, a set of twelve relevant use cases have been identified, as well as the functional and performance requirements that the network architecture solutions proposed should address, and the scenarios that should be used for the validation of the solutions [5GN-D22]. We consider there is a good alignment of the different demos with these 12 use cases as it is shown in the following table:

Table 4-1: Alignment with WP2 Use Cases

Use Case	Demo 1	Demo 2	Demo 3	Demo 4
Industry Control	●	●	●	●
Enhanced Mobile Broadband	●	●	●	●
Emergency Communications			●	●
Vehicle Communications	●		●	●
Sensor Networks Monitoring		●	●	●
Traffic Jam			●	●
Real-time Remote Computing			●	●
mMTC	●		●	●
Quality-aware Communications	●	●	●	●
Fixed-Mobile Convergence			●	●
Blind Spots			●	●
Open Air Festival			●	●

As we see, Demos 3 & 4 can be considered for all these use cases, since the concepts of security and the economic analysis are common features that should be considered in all these cases.

WP2 also derived a set of functional requirements from these twelve use cases. The following table also shows the alignment of the demos with these functional requirements:

Table 4-2: Alignment with WP2 Functional Requirements

Use Case	Demo 1	Demo 2	Demo 3	Demo 4
Fast network reconfiguration within a network slice	●	●		●
Fast network reconfiguration between network slices		●		●
Separation and prioritization of resources on a common infrastructure	●	●		●
Multi-connectivity in access and non-access part of the 5G system				●
Massive scalability of protocol network functions				●
Highly efficient transmission & processing				●
QoE/QoS awareness		●		●
Adaptability to transport network capabilities				●

Low latency support	●	●		●
Security			●	●

Finally, WP2 groups these requirements along three axes, i.e., very low latency and reliability for critical machine-type communications (mMTC); high throughput (compared to legacy networks) for massive Mobile BroadBand (MBB) communication; and the ability to support high volumes of devices for massive MTC. In Section 2.2, we have seen that our four demos have been designed according to this, hence, we consider this validates well the requirements defined in the WP2.

4.1.4.2 Integrate Information from other WPs

One of the initial general scope objectives for WP6 was to "*implement, test and validate the partial designs coming from the other WPs, allowing the refinement of the different modules' design*".

We consider this objective has been extensively fulfilled as well. In Demo 1, the most evident contribution has been the design and implementation of the SDM-C component defined in WPs 3, 4 and 5, including the communication protocol developed in Demo 1.

Regarding Demo 2, as previously explained, we've mainly addressed the design of the MANO layer of the 5G NORMA architecture defined by WP3. Also, the VNF mobility concepts introduced in WP4 have been addressed here, as well as some of the concepts related to the QoS/QoE-awareness processes defined in WP5.

In Demo 3, this has been satisfied to some extent as well, particularly in terms of the V-AAA concept in WP3 and the tokenisation aspect with which that has been realised.

4.1.4.3 Feedback to other WPs

We consider this KPI has been extensively fulfilled. For instance, in relation to Demo 2, we've actively collaborated in the definition of the MANO architectural principles in WP3. Specifically, we collaborated to define the WP3 multi-tenant and multi-slicing orchestration architecture, which includes not only the control and data layer, but also the orchestration and management layer (as implemented in this demo). This contribution is described in the WP3 Deliverable D3.2 [5GN-D32]. Also, beside the specific topics covered by the demo itself, we also collaborated defining the multi-domain orchestration principles, as well as evaluating the ways to integrate PNFs in the architecture. These topics are described in Deliverable D3.3 [5GN-D33].

Regarding Demo 2, extensive and continuous feedback has been provided to WP3, especially the Security task therein. The design of the concept is intended to reflect directly the developments on AAA made in WP3, and to verify and assess the refinements necessary to that WP3 concept in developing a viable system.

Regarding WP5, we collaborated to define the QoS/QoE framework, and also, providing information about different strategies for the VNF migration processes (this is addressed in the scope of the D5.2 document [5GN-D52]).

Regarding dissemination (WP7), we have collaborated in different papers and presentations where the general principles addressed by this demo are described. We have also contributed to include content (in relation to demos) in the official 5G NORMA website.

4.1.4.4 Contribution to the Open Source Community

The main contribution to the Open Source community comes from Demo 2 and specifically from the RAN part. The modifications made in the Mobile Network Stack software implementation (i.e., srsLTE) are now available in the most important Open Source Code repository (Github) at this url: https://github.com/GinesGarcia/srsLTE_oaiCore (GNU Affero General Public License). Also, we provided a thorough guide for the configuration of an Open

Source mobile network stack with virtualized core, available at https://github.com/GinesGarcia/OpenAirInterface5G_CloudCore.

4.1.4.5 Contribution to the 5G-PPP Program Activities

Another general objective for the WP6 was to contribute to 5G-PPP program activities related to demos and prototypes to align them across 5G-PPP projects also developing demos and prototypes.

From the work performed in Demo 3, extensive input has been provided to the 5G PPP Security WG through participation in all meetings. This is manifested, for example, in the input to the 5G PPP Phase1 Security Landscape White Paper ([SEC-WP], p.21 for example). The discussion in the white paper matches with what we have done and achieved in Demo 3, and lessons learned in the development of the concept enriches the observations drawn in the white paper.

Regarding the collaboration with other 5G-PPP projects, as we can see in Section 5 below, some of our demos were jointly presented in relevant events together with the FANTASTIC-5G project.

4.1.4.6 Other Dissemination Activities

We've presented early and intermediate versions of some of our demos in conferences and relevant events. This is treated with more detail in the Section 5 (Impact). Besides, we've also published technical papers related to our demos, which are listed in Table 4-3.

Table 4-3: Published Papers

Authors	Title	Event	Date
A. Colazzo, R. Ferrari, R. Lambiase (AZCOM)	Achieving low-latency communication in future wireless networks: the 5G NORMA approach	FANTASTIC-5G WS on "Ultra-Reliable and Mission Critical Communication", EuCNC; Athens, Greece	2016/06/27-30
S. Khatibi, F. Sheikh	Service-Aware Network Reconfiguration for 5G Networks	NOMOR Whitepaper on NOMOR website	Nov. 2017
S. Wong et al. (KCL).	Virtualized Authentication, Authorization and Accounting (V-AAA) in 5G Networks	2017 IEEE Conference on Standards for Communications and Networking (CSCN),	18-20 Sept. 2017

4.2 Lessons Learned

In this section we explain the main lessons learned from the four demos previously described.

Demo 1 is focused on the SDM-C component, which is one of the core components of the 5G NORMA architecture. Both parts of Demo 1 provide different conclusions. The first part is focused on a real behavior on the SDM-C in a legacy network and shows how the SDM-C could behave. It is shown how it can manage a physical eNB with a communication protocol, which required a reasonable effort to be defined. Several commands and feedback can be sent by means of this protocol, easing the management of the network, reducing the number of interfaces needed. On the other hand, we have also seen how a PNF like component can be insert in the new 5G NORMA architecture with reasonable effort, which has allowed the connection and consequently the communication between the SDM-C and the eNB. This led to a good transition from the legacy to the new 5G architecture, exploiting the novelties proposed by 5G NORMA.

Regarding the software part in Demo 1, one of the key learnings has been about the presentation of the demo to the audience. We have seen that using two separated network slices with totally different QoS requirements was very useful for communicating the concept of the network function decomposition. In this sense, the software part of the demo helped the audience to understand the problem in a large scale and from different stakeholders' point of view. Also, regarding the implementation of the Virtual Cells model using a Q-Learning agent, we appreciated that, although it is an interesting idea, it is not always efficient. The introduction of Q-Learning as one the many machine learning approaches to solve the problem was an interesting approach, but although it solved the problem very well, using it in the demonstrator resulted in a non-trivial task.

Regarding Demo 2, a positive lesson regarding the implemented NFV MANO platform is that we have indeed verified that, by applying automated on-boarding techniques, it is possible to achieve reduced deployment times. This is an important KPI for telco operators, since classical network service implementations suffer from long deployment times.

Another important lesson from Demo 2 is regarding the usage of general purpose out-of-the-box orchestration solutions. We consider they are not yet mature enough (at least while we were developing our demo). Our perception after evaluating different possibilities (as we've seen in Section 3.2.2) is that the state of the art regarding this type of general-purpose platforms is still not sufficient yet, e.g., we have seen:

- VNFs are not fully interoperable with different platforms (the VNFs need to be slightly adapted to be deployed on different orchestrators). Although ETSI defines how the descriptors and procedures should be, it is probably necessary to introduce a multi-vendor certification process to ensure compliance and interoperability.
- For each platform, own NSD/VNFD templates, descriptors, and blueprints are necessary, e.g., formats such as Tosca, ETSI NFV, Yaml, and others are used. This is not technically complicated, but a barrier.
- Usually, additional development is needed, especially in Open Source projects. We've seen this in TeNOR, but also in OpenBaton, for which we discovered that for deploying our own VNFs it was necessary to modify them by developing what they call a 'VNF Manager adapter'.
- Key advanced features are still not available at large, e.g., network slicing, auto VNF scalability, or VNF-specific host placement.

We have seen a recent external reference supporting this lesson which is the 1st ETSI NFV Plugtests Report [ETSI.NPR-17] released by the ETSI Centre for Testing and Interoperability, which evaluates different MANO solutions and NFV platforms. The report actually shows high rates of test execution and interoperability for simple features such as Network Service on-boarding, instantiation and termination, but results are still limited for more complex operations, e.g., scaling or NS updates. In addition, no results are reported beyond basic functional testing, e.g., test cases about performance or migration of NFs are not reported.

Another reference supporting this perception is the well-known SDN Central 2017 Annual Survey [SDXC-17], where about half of the responders declare they will consider Open Source MANO platforms only "once they are more mature" (see Figure 4-13).

Considering the results in these reports and our own experiences during the demo development our conclusion is that these types of out-of-the-box platforms are perhaps suitable to deploy static VNFs sets (for instance, to deploy in a virtualized environment the different NFs the telco operators already have in their networks), but the advanced features (such as dynamic scaling, intelligent placement or migration of network functions) are still work in progress.

Hence, a lesson learned regarding this is that if advanced or very specific NFV orchestration functions are required, it would be better to base the development on a good VIM and programmatically implement those advanced management and orchestration functions. Probably this situation will change in the future if the development on these general purpose platforms continues, but we consider the state of the art is still limited right now.

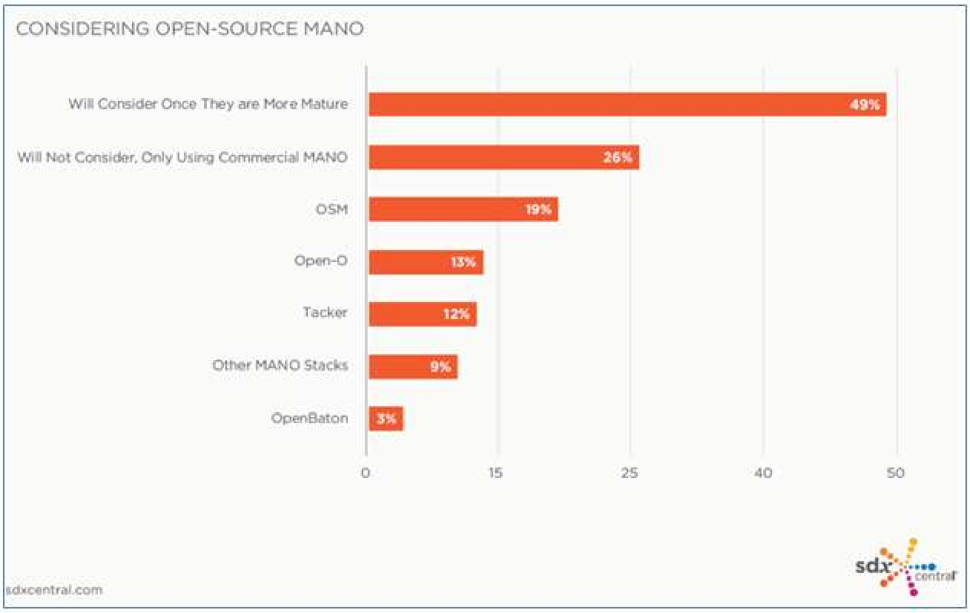


Figure 4-13: Survey considering Open Source MANO platforms

However, as a positive side effect, we've seen this solution of mixing already available software with ad-hoc developed code provides more degrees of freedom to implement some specific 5G NORMA functionalities which are not considered in the already existing out-of-the-box orchestration platforms (e.g., TeNOR, OpenBaton). An example could be the implementation of components specifically designed in this 5G NORMA project, such as the SDM-O or the Inter Slice Resource Broker (IS-RB) defined in WP3 (see Figure 2-1).

Finally, although OpenStack already is a very successful Open Source project with important supporting organizations and a wide community of users and developers, from our experience in this demo, it still needs further development compared to other proprietary solutions. For instance, if we compare with previous experiences with the equivalent VMware solutions our opinion is that some features are still missing or could be improved in OpenStack[14]. Beyond the fact that VMware is proprietary and OpenStack is Open Source, we consider VMware solutions to be much more mature, free of critical bugs and with a good support and documentation, which speeds up the learning period. It offers multiple features and has been already used in very large ecosystems. With OpenStack we have experienced very complex setup processes and troubleshooting; a steeper learning curve is necessary and the documentation is not always current. As a whole, we think that OpenStack has a big potential, but we consider it to be much less mature and flexible (probably due the relatively lesser time in the market).

Similar considerations can be also drawn for the RAN stack development. Current OpenSource alternatives for the RAN stack are still in their early days when dealing with network slicing. There are indeed research efforts in this area [FOU-16/17], but their applicability is limited and they do not consider orchestration since the very beginning. Moreover, they are efforts parallel to ours, i.e., there is a growing attention in this field but we could not rely on these solutions.

As a consequence, we opted for srsLTE as it is a smaller project that allowed us to gain the knowledge needed for the modifications we have developed. The core VNFs are the ones provided by the OAI platform as, currently, it is the only Open Source implementation of the

[14] Although the general approach and philosophy of these two ecosystems are different, there is a clear overlap between functionalities of certain OpenStack projects and VMware products.

core. Another good metric for measuring the growing attention on these topics is the positive feedback that we received on our public code repository.

Regarding Demo 3, key lessons revolve around realising the V-AAA concept in a computationally feasible way even in the most challenging scenarios. It is noted that the implementation has deliberately been done on a network of Raspberry Pis, demonstrating that the idea is feasible on low computational capability equipment. A further key lesson is that separation of AAA functionality in a hierarchy can be challenging for realistic scenarios, thereby limiting the usefulness of the concept unless careful forward-planning is done regarding the required AAA operations and preparation of them with appropriate information in the local domain.

Finally, the main lessons learned from Demo 4 are the following:

- We have gone through the experience of showing results of a complex model through a simple interactive webpage. Although detailed results can be accessed in WP2 documents, key messages from each evaluation case are illustrated through a simple dashboard.
- Although in Deliverable D6.1 [5GN-D61] we anticipated that the web tool would be developed by using the PHP (Hypertext Pre-processor) and HTML (Hypertext Mark-up Language) web-programming languages, we finally encounter a more convenient approach based on Spreadsheetweb [SSW]. We consider this approach is simpler and sufficient to show the results of the 5G NORMA model. This also provided us the opportunity to learn using this new platform how to convert our complex dashboard model from a local machine into a web-browser accessible by the public.

5 Impact

During the development of the project, initial and intermediate versions of the demos that we are describing in this document have already been shown at relevant events. In addition, WP6 published papers on the demonstrators and source code developed for the demos has been released to the Open Source community. These dissemination activities are explained in the following sections.

5.1 Conferences

5.1.1 MWC 2016

An initial version of Demo 1 was presented during the Mobile World Congress 2016 at the European Commission booth. The setup of the demo was performed with two laptops: one of them running the system-level simulator and the other one running the Graphical User Interface. The GUI was connected to the screen provided by the MWC organizers. The setup is shown in the following Figure 5-1.

Figure 5-1: MWC'16 Setup

AZCOM showed a video of its demonstrator where a scale model rally car is driven using a commercial tablet as the steer (see Figure 5-2) to show the latency impact, considering two different situations. The first one mimics a commercially deployed LTE network, in which an average E2E latency of hundreds of ms is experienced. In the second scenario, the EPC routing components (S-GW) are moved into the eNB baseband board. Specifically, the S-GW is hosted by the base band System on a Chip (SoC). In the first scenario, the high latency reduces the driving quality, as the command arrives at the model rally car with great delay. In the second scenario the control feeling is very good since the response of the car is immediate.

Figure 5-2: Running Demonstration at MWC16

NOMOR also presented its network simulator and exhibited the advantages of the novel network architecture with multi-service and context-aware scheduling as shown in Figure 5-3. The demo considered two different services/user profiles, which are:

- High Definition Video (HD-Video) streaming, i.e., a service with high data rate required per session, for pedestrians,
- Vehicle-to-everything (V2X), containing Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) set of services with very low delay budget and low traffic volume.

It is shown that placing the Radio Resource Management (RRM) functionalities into the central cloud data centre enables implementation of interference mitigation strategies using centralised coordinated scheduling in order to improve the throughput for HD-video users. In contrast, shifting the mobile network functions to the edge cloud improves the end-to-end latency and improves Quality of Service (QoS) for time-critical services such as autonomous driving.

Figure 5-3: NOMOR's demonstration in MWC16

This demo at MWC'16 received a lot of positive feedback and attention not only from the general audience and representatives of telecommunication companies, but also from delegates of the European Commission. The demo was presented to a large number of attendees and was highly appreciated by them as shown in the following figures. The most appraised features of the demo were its ease-of-understanding despite presenting complex concepts and its potential for impacting the future 5G networks research.

Figure 5-4: Demo being explained to the European Commission delegates

5.1.2 EuCNC 2016

The same demo was presented also at the EuCNC 2016, where again very positive feedback was received. The demo was shown also to delegates of the European Commission (see Figure 5-5), who appreciated the work very much.

Figure 5-5: European commission visiting the booth

5.1.3 MWC 2017

A 5G NORMA demonstration was presented in the AZCOM technology booth at the Mobile World Congress 2017 (MWC'17). The demonstration showed the Native Multi-Service Architecture which is described in Section 3.1. The demonstration, depicted in Figure 5-6, was performed with two laptops where one of them was used by the hardware demo to show the impact of the reconfiguration of the network on the latency, and the second one was used to show the system-level simulator of the software demo.

Figure 5-6: MWC'17 Set-up

The 5G NORMA demonstration at MWC'17 received a lot of positive feedbacks and attentions from the audience. The demonstration was presented to a great number of industrial and academic attendees and was highly appreciated by them as shown in Figure 5-7.

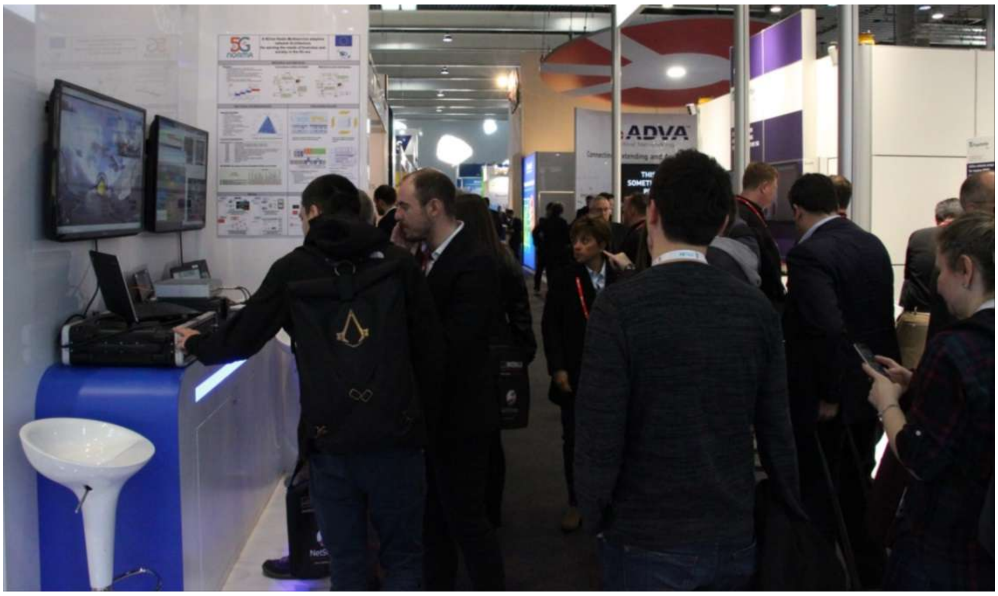


Figure 5-7: MWC'17. Visitors Watching Demo 1

5.1.4 ICC 2017

Early versions of Demos 2, 3 and 4 were presented during the IEEE International Conference on Communications workshops session (see Figure 5-8).

Figure 5-8: Presentation of Demos 2, 3 & 4 during the ICC'17

UC3M and Atos presented an initial static version of Demo 2 consisting of the VNFs provided by UC3M and running on the Atos virtualization environment. They also included some of the UC3M application developments for this demo, i.e., an initial version of the mobile application to read QR codes. It showed real-time measurements (physical measurements from one of the laptops in the demo) and a visualizer (GUI) to illustrate the different virtual functions in the deployment.

KCL also presented an initial version of their security demo with a subset of the Raspberry-PI cards, the SDN controller and the related V-AAA software and an initial version of the GUI.

Finally, RW presented its online interactive GUI showing the cost benefits resulting from the evaluation cases 1 and 2. Although still an intermediate version of the demo, the layout and functionality of the GUI were already very similar to its final version.

5.1.5 EuCNC 2017

At EuCNC 2017, AZCOM and NOMOR presented their joint demo, called Native Multi-Service Architecture, which presents the proof-of-concept for a flexible, adaptive, intelligent, and service-aware (de)composition and adaptation of NFs and services. The demo was composed of two parts (software and hardware), developed by NOMOR and AZCOM, respectively. The key elements shown during this event were the following:

- A Software Defined Mobile network Controller (SDM-C) and software eNodeBs (eNBs) provided by NOMOR,
- A hardware eNB provided by AZCOM, which together with the SDM-C comprises the hardware demo.

RW also presented an preliminary version of Demo 4 with an almost final version of the online interactive GUI showing the cost benefits of 5G NORMA architecture, including 5G NORMA features' selections such as flexible C-RAN and multi-tenancy.

The booth was shared with the FANTASTIC-5G project and was visited by many attendees, who were impressed by both demos. The following figures show the booth and some of the audience during this event.

Figure 5-9: Native Multi-Service Architecture demo at the FANTASTIC-5G and 5G NORMA booth.

Figure 5-10: Visitors watching Native Multi-Service Architecture demo

6 Conclusions

This document has described the demonstrators developed for the 5G NORMA project in the context of Work Package 6. Initially, the three main 5G NORMA concepts demonstrated were explained: Security, Economic Feasibility and Network Slicing (in this last case covering both Network Control and NFV Management and Orchestration). After this, the services implemented by the different demonstrators were explained, as well as their relation with the main three concepts previously introduced. After that, a detailed description of the implementation of each demo has been provided, as well as the main results and the lessons learned for each one.

Regarding Demo 1 (Native Multi-Service Architecture), the introduction of the 5G NORMA specific SDM-C element in the architecture is described as well as the advantages derived from it compared to legacy network architectures. This demo shows how the SDM-C component can be used to coordinate multiple eNBs based on the traffic demands and the reports received from the eNBs. We have seen also how the SDM-C component can be used to command the network function elements to be placed in the edge cloud in order to reduce the latency or in the central cloud to implement centralised algorithms. The demo uses the software eNBs to elaborate more on the proof of concept, which cannot easily be demonstrated using the physical eNBs.

In addition, this demo also presents the formation of virtual cells, i.e., a centralised radio resource management algorithm from WP4. The demo allows the audience to have an in-depth experience with the concept of virtual cells and understand the trade-off of the throughput increase for the UEs near the edge of the cells versus the increase of interference and reduction of the throughput of the UEs in the centre of the cells.

In Demo 2, we showcase mostly the orchestration aspects of network slicing, showing some of the 5G NORMA concepts developed in WP3, 4 and 5. By showcasing two different network slices (mobile broadband and reduced latency), we provide a PoC for this novel concept that has been considered in the 5G NORMA architecture and innovations since the very beginning and will be a fundamental characteristic of future 5G networks. We included innovations from both WP3 (the MANO architecture), WP4 (the RAN slicing options) and WP5 (the orchestration algorithms). The final demo setup shows the feasibility of 5G NORMA concepts in an end-to-end (including orchestration) sliced network.

Demo 3 has shown the operation of the V-AAA concept to bring security management into the local domains of network slices and closer to the edge, e.g., to reduce latency in some operations, therefore, on both counts, being conducive to envisioned future 5G networks and applications. The operation of the demo according to performance expectations such as tenant isolation and data consolidation, GUI operation to provide the operator control and linking of that with a database, using tokenisation to provide security, among other aspects, has been shown. While the entire framework is fully developed certain issues were detected regarding the communication between the threads of the SDN controller and GUI. In any case, we consider that the key implementation of the entire concept on commodity hardware has been demonstrated even with the challenge of using components of extremely low computational capability.

The online Demo 4 presents the economic benefits of a 5G network, which is one of the key outcomes of the 5G NORMA project. The results provided in this demo via the web-based interface are abstracted from the network modelling results developed in WP2. In the latest demo version, the cost benefits of 5G NORMA innovations were shown in terms of CAPEX, OPEX and TCO. The cost results included are for all evaluation cases, i.e. evaluation case 1, 2 and 3. In particular the comparable costs of flexible C-RAN to DRAN architecture (evaluation case 1), multi-tenancy cost benefits (evaluation case 2) and limited costs penalty of multiservices (evaluation case 3) are shown. In addition, the revenue benefits of 5G NORMA are shown for the multi-service case.

The document also describes the main obtained results, which are classified in four different types:

- Performance KPIs, which are mainly coming from Demos 1 and 2 reporting the most significant objective values in the radio part (throughput, latency, delay and bandwidth) and the orchestration part (NS deployment and VNF migration times).
- Verification, where we explore those non-quantitative KPIs related to the main WP6 objectives.
- Socio-Economic findings, where the main facts found in Demo 4 are summarized.
- Contribution-related results, where we analyse the way we have implemented concepts defined in other WPs as well as how we provide feedback to them and other entities.

Beside these results, the document also describes the main lessons learned from the implementation of the different demos. The main ones are related to the implementation of the SDM-C concept and the SDM-C/eNB communication protocol. In addition, valuable learnings were obtained regarding the orchestration platform and the RAN stack development. Regarding the security concepts, the main lesson learned comes from the implementation of the V-AAA concept itself. Demo 4 also provides valuable information through its different evaluation cases, i.e., costs comparisons for CRAN vs. DRAN, multi-tenancy vs. multiple independent networks, multi-service cost, and revenue results.

Finally, the high impact of showcasing the different demos at MWC, EuCNC and ICC during the whole project runtime has been highlighted.

References

[36.214] 3rd Generation Partnership Project Technical Specification Group Radio Access Network Evolved Universal Terrestrial Radio Access (E-UTRA) Physical layer Measurements, September 2017

[5GN-D22] EU H2020 5G NORMA, “D2.2: Evaluation methodology for architecture validation, use case business models and services, initial socio-economic results”, Oct. 2016.

[5GN-D23] EU H2020 5G NORMA, “Evaluation architecture design and socio-economic analysis – final report”, Scheduled for Dec. 2017 (in preparation while this document is written).

[5GN-D31] EU H2020 5G NORMA, “D3.1: Functional Network Architecture and Security Requirements”, Dec. 2015.

[5GN-D32] EU H2020 5G NORMA, “D 3.2: 5G NORMA network architecture – Intermediate report”, Jan. 2017

[5GN-D33] EU H2020 5G NORMA, “Deliverable D3.3: 5G NORMA network architecture – Final report”, Oct. 2017.

[5GN-D42] EU H2020 5G NORMA “RAN architecture components – final report”, June 2017.

[5GN-D51] EU H2020 5G NORMA, “D5.1: Definition of connectivity and QoE/QoS management mechanisms – intermediate report”, Nov. 2016.

[5GN-D52] EU H2020 5G NORMA “Definition and specification of connectivity and QoE/QoS management mechanisms – final report”, June 2017.

[5GN-D61] EU H2020 5G NORMA, “D6.1: Demonstrator design, implementation and initial set of experiments”, Oct. 2016.

[AM-13] A. Madhavapeddy et. al, “Unikernels: Library Operating Systems for the Cloud”, ASPLOS’13, March 16–20, 2013, Houston, Texas, USA.

[BISW] I. Biswas et. al, “A Practical Evaluation in Openstack Live Migration of VMs using 10Gb/s Interfaces”, School of Computing and Information Engineering, Ulster University, Coleraine, Northern Ireland.

[CAFG-17] M. R. Crippa, P. Arnold, V. Friderikos, B. Gajic, C. Guerrero, O. Holland, I. Labrador, V. Sciancalepore, D. von Hugo, S. Wong, F. Z. Yousaf, B. Sayadi, “Resource Sharing for a 5G Multi-tenant and Multi-service Architecture”, European Wireless 2017, The Way Towards Holistic 5G, 2017

[CSSX-16] S. Costanzo, R. Shrivastava, K. Sarndanis, D. Xenakis, X. Costa-Prez, and D. Grace, “Service-oriented resource virtualization for evolving tdd networks towards 5g,” in 2016 IEEE Wireless Communications and Networking Conference, April 2016, pp. 1–6

[DJANGO] The Django Web Framework, accessible at https://www.djangoproject.com, accessed November 2017

[DPK-16] S. Dräxler, M. Peuster, H. Karl, M. Bredely, J. Lessmanny, T. Soenenz, W. Tavernierz, S. Mendel-Brinx and G. Xilouris, “SONATA: Service Programming and Orchestration for Virtualized Software Networks”, Paper submitted and accepted in the IEEE International Conference on Communications 21-25 May 2017 Paris, France.

[EASYAR] EasyAR, https://www.easyar.com

[ENM-14] ETSI, “Network Functions Virtualisation (NFV); Management and Orchestration”, ETSI TR, GS NFV-MAN 001 [PDF], v1.1.1, Dec. 2014.

[ESC-98] Richard B. Ertel, Paulo Cardieri, Kevin W. Sowerby, Theodore S. Rapport, and Jeffery H Reed. Overview of spatial channel models for antenna array communication systems. IEEE Personal Communications, Feburary 1998.

[ETSI.NPR-17] 1st ETSI NFV Plugtests Report, V1.0.0, https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/1st_ETSI_NFV_Plugtests_Report_v1.0.0.pdf, Mar 2017

[FAUXTON] Project Fauxton CouchDB plus User Interface, accessible at http://couchdb.apache.org/fauxton-visual-guide, accessed November 2017

[FFMPG] Ffmpeg, https://www.ffmpeg.org

[FFP] Ffplay, https://ffmpeg.org/ffplay.html

[FLASK] Flask, http://flask.pocoo.org

[FOU-16] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina and K. Kontovasilis. “FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks”. ATC 2016, All Things Cellular Workshop, in conjunction with ACM MobiCom 2016, 7 October 2016, New York, USA

[GOD-97] LAL C. GODARA. Application of antenna arrays to mobile communications, part ii: Beam-forming and direction-of-arrival considerations. Proceeding of the IEEE, 85(8), August 1997.

[GSTR] Gstreamer, https://gstreamer.freedesktop.org

[ITU-REC] Recommendation ITU-R M.2083-0 (09/2015), “IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond”, M Series, Mobile, radio determination, amateur and related satellite services. https://www.itu.int/ dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf

[KT-14] Kyoung-Taek Seo et. al, “Performance Comparison Analysis of Linux Container and Virtual Machine for Building Cloud”, Advanced Science and Technology Letters Vol.66 (Networking and Communication 2014), pp.105-111 http://dx.doi.org/10.14257/astl.2014.66.25

[MONGO] MongoDB, https://www.mongodb.com

[OPB] OpenBaton, http://openbaton.github.io

[OPSTK] OpenStack, http://www.openstack.org

[OS4J] OpenStack for Java, http://www.openstack4j.com

[OSM] Open Source Mano, https://osm.etsi.org

[OVS] Open vSwitch Virtual Switch, accessible at http://openvswitch.org, accessed November 2017

[RYU] Ryu SDN Framework, accessible at https://osrg.github.io/ryu, accessed November 2017

[SDL] SDL Media library, https://www.libsdl.org

[SDXC-17] SDXcentral - 2017 NFV Report Series Part 2

[SEC-WP] 5G PPP Phase1 Security Landscape White Paper, accessible at https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP_White-Paper_Phase-1-Security-Landscape_June-2017.pdf, accessed November 2017

[SNMX] P. Maersk-Moller, “Snowmix”, http://snowmix.sourceforge.net

[SSPR-14] Samdanis, K., Shrivastava, R., Prasad, A., Rost, P., & Grace, D. (2014, September). Virtual cells: Enhancing the resource allocation efficiency for TD-LTE. In Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th (pp. 1-5). IEEE.

[SSW] Spreadsheetweb, www.spreadsheetweb.com

[TEN] TeNOR, https://github.com/T-NOVA/TeNOR.

[TMCT] TomCat, http://tomcat.apache.org

[WAT-92] C. J. C. H. Watkins y P. Dayan. "Q-Learning". Machine Learning, 8: 279-292, 1992.

[XIL-14] G. Xilouris et. al, “T-NOVA: A Marketplace for Virtualized Network Functions”, European Conference on Networks and Communications, Bologna, Italy, June 2014.

Annex A Communication Protocol

The messages are exchanged using UDP/IP protocol because the periodic feedback messages are sent every TTI (Transmission Time Interval), and retransmission for lost packets is not desirable, i.e., UDP has been preferred over TCP.

In the following Table A-1, all the messages exchanged between SDM-C and eNB and their structures are summarised.

Table A-1: List of messages and their payload content

MSG#	Type	Payload						
1	Feedback	Number of available PRBs.						
2	Feedback	Number of connected UEs.						
3	Feedback	UE RNTI	Bearer ID	DL WCQI	DL PRBs	DL MCS	DL QCI	DL Load
4	Feedback	UE RNTI	Bearer ID	UL QCI	UL Load			
5	Feedback	UE RNTI	DL RSRP	DL RSRQ				
6	Command	Scheduler enable/disable						
7	Command	UE RNTI	Throughput gain	Latency gain				
8	Command	UE RNTI	UE Priority factor					

In the following we describe the different message types.

Messages 1 & 2: They contain the number of available resource blocks at the eNB and the number of UEs connected to the eNB respectively. The first message is sent only during the system start-up, providing information for SDM-C about the available bandwidth, hence it is an indication of the cell capacity. The second one is sent only when a new UE attaches (or connects) to the eNB or when a connected UE releases (or disconnects) from the eNB.

Message 3: This is a message sent by the eNB at the start of every reporting interval, i.e. once per TTI. It is a Downlink (DL) transmission feedback message which is an indication of the throughput generated by a specific user for a specific service. The eNB sends one message per connected UE (and bearer). The message contains information such as UE's bearer identifier (ID), its channel quality indicator and the number of resource blocks assigned for scheduling this UE at this TTI. It also contains the QCI which is an important parameter for the SDM-C, because it helps it to distinguish between different services.

Message 4: It contains similar information as Message 3 but for the uplink part. This message contains QCI and throughput for uplink traffic. These two parameters are used by the SDM-C's optimisation and network (re)configuration algorithms.

Message 5: This message contains two critical parameters for each UE, namely the RSRP, which helps the SDM-C in determining the received signal power, and the Reference Signal Received Quality (RSRQ), which also contains information about the signal received from other eNBs and the noise [36.214]. It is collected and sent at every reporting interval TTI in the form of downlink power parameters message.

Message 6: This is a command message. It is used by the SDM-C to enable/disable the scheduler at the eNB. When enabled, the eNB operates the scheduling policy by itself, i.e., not considering messages from the SDM-C. On the other hand, when disabled, the eNB gives priorities to commands from the SDM-C over the scheduler's decisions. The disable message is sent only when there is a change necessary and not periodically as it was the case with most feedback messages. The eNB then replies with ACK/NACK messages.

Message 7: This message contains the gain regarding throughput and latency achieved by the network (re)configuration. It is estimated by the SDM-C and is transmitted to eNBs in a separate command message.

Message 8: It contains the serving weight for a specific user based on its ID. This value is used to differentiate the service requested from each user. Using this information, the SDM-C controls and prioritise different services based on different network (re)configurations.

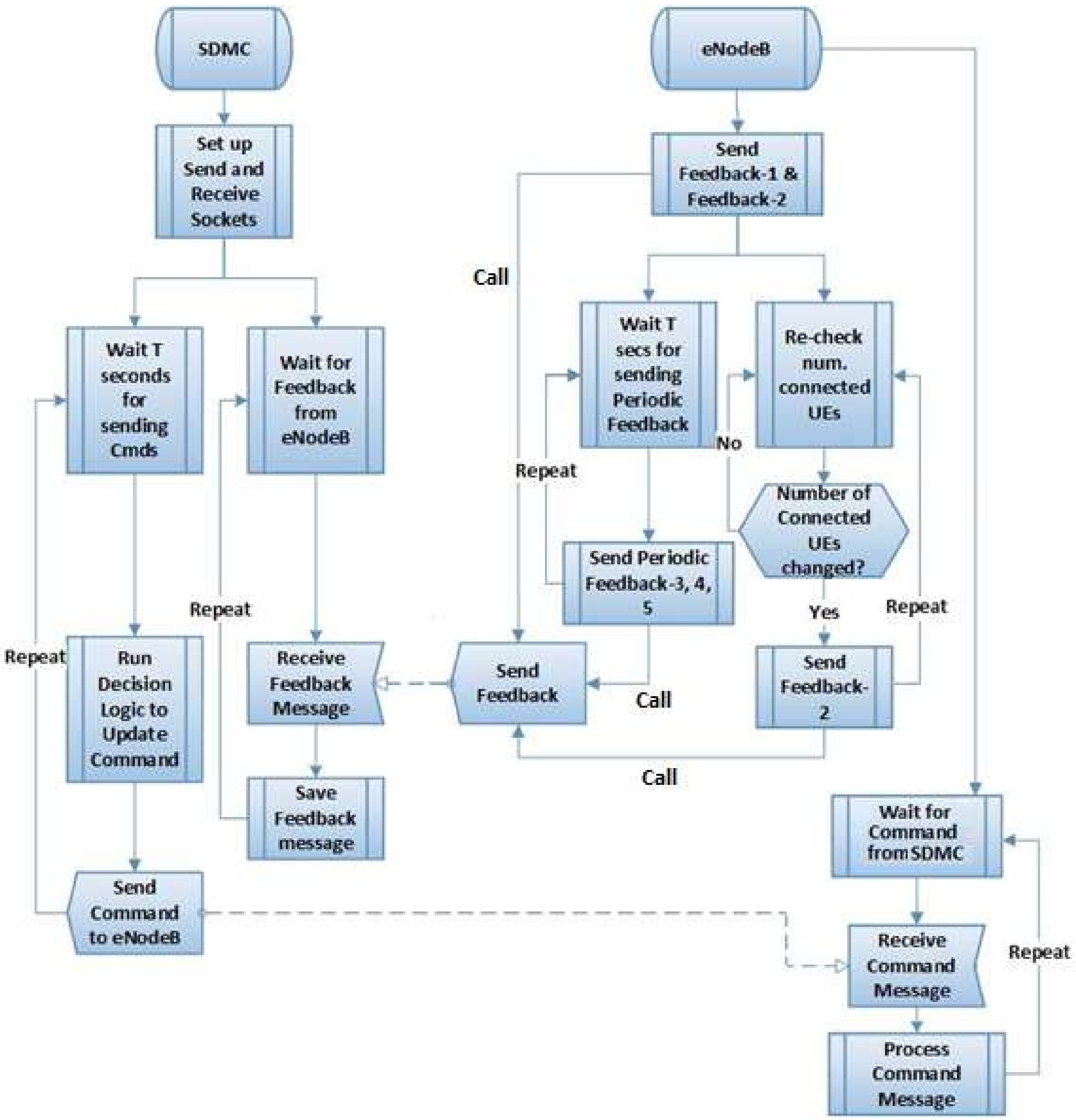


Figure A-1: Flowchart for Communication between SDM-C and eNBs

Message Flow between SDM-C and eNBs

The send and receive functionalities of the SDM-C work in parallel, as shown in the flowchart in Figure A-1, and are handled by separate threads.

For the communication between SDM-C and eNBs, the SDM-C application first opens a receiving IP socket. A receiving thread in the SDM-C continuously listens and receives data from different eNBs at the receiving port and saves it internally in the application's shared memory. The data is then read by the main thread of the SDM-C application and processed. The received feedback data is fed to the decision logic at the SDM-C. In addition to the continuous receiving functionality of SDM-C, some command messages also need to be sent by SDM-C to the eNB(s).

The hardware eNB, after start-up, first sends the non-periodic feedback messages to SDM-C, i.e., feedback messages 1 and 2 in the previous table. The remaining feedback messages 3, 4 and 5 are sent by the eNBs periodically at predefined intervals. The order in which eNBs may send these periodic messages is irrelevant for the communication logic, e.g., message 5 may be sent before message 3.

The SDM-C also sends periodic command messages to eNBs in parallel to the reception of feedback messages. The frequency of these command messages is configurable, being usually set in the order of tens of milliseconds. SDM-C waits a predefined time-interval, i.e., in the order of seconds and configurable from the demo's input files, and sends new command messages to all connected eNBs based on the decision logic, as highlighted in the flowchart.

The communication protocol has been designed in such a way that it is easily extensible to be used for connecting multiple eNBs at one time. The protocol has been successfully tested for communication with up to 10 simulated eNBs, although theoretically, it could work with an indeterminate number of eNBs. The same communication protocol has been implemented in the SDM-C, the hardware eNB and also in the NOMOR's software network simulator for communicating with simulated eNBs.

Annex B Demo 2 eMBB Use Case REST API

This Annex describes the different REST APIs used to communicate the Streaming Server block with the different internal servers in the MBB application (see Figure 3-17). We have three different APIs:

- NormaVideoOutput, to communicate to the Main Videos Server block
- NormaVideoLS, to communicate to the Sign Language Videos server
- NormaVideoSubtitle, to communicate to the Captions Server

These REST APIs are used by the Streaming Server to request the execution of the necessary control commands once these servers are up and running in form of VNF instances. The communication to the main video server is performed in order to: i) start the playback of the main video itself, and ii) to get the timestamp of a video already being reproduced. The latter is done in order to produce synchronized add-ons when they are requested by the user which could happen at any time while the main video is being reproduced. The timestamp information, when obtained, is propagated to the Captions Server or the Sign Language Video Server. Additionally, each API is used to specify the personalized features of each add-on (e.g., subtitles position, size or colour).

NormaVideoOutput API

Url base: http://domain/NormaVideoLS/service/norma

Path	/test
Method	GET
Parameter	None
Header	Content-Type: application/json
Payload	None
Return HTTP code	200 OK
Comments	Test service
Returns	It is alive

Path	/time
Method	GET
Parameter	None
Header	Content-Type: application/json
Payload	None
Return HTTP code	200 OK 404 Not Found
Comments	Place the sign language video in the screen taking in consideration the values provided by the JSON user profile received by parameter and starts ffmpeg output taking in consideration the time of the video received also by parameter

Returns	{
	"statusCode": <HTTP Status Code>,
	"description": "<success or error description>",
	"data": "<value>"
	}

NormaVideoOutput API

Url base: http://domain/NormaVideoOutput/service/norma

Path	/test
Method	GET
Parameter	None
Header	Content-Type: application/json
Payload	None
Return HTTP code	200 OK
Comments	Test service
Returns	It is Alive

Path	/ls
Method	POST
Parameter	None
Header	Content-Type: application/json

Payload	{
	"userid":"<value>”,
	"username":"<value>",
	"service":"sign",
	"timestamp":"<value>",
	"sign_options":{"location":"<value>",
	"size":"0.8 | 1.2"}
	}

Return HTTP code	200 OK 404 Not Found
Comments	Place the sign language video in the screen taking in consideration the values provided by the JSON user profile received by parameter and starts ffmpeg output taking in consideration the time of the video received also by parameter

Returns	{
	"statusCode": <HTTP Status Code>,
	"description": "<success or error description>"
	}

JSON example:


```
{
  "userid":"1", "username":"diego", "service":"sign", "timestamp":"00:05:23",
  "sign_options":{"location":"40 40", "size":"1.2"}
}
```

NormaVideoSubtitle API

Url base: http://domain/NormaVideoSubtitle/service/norma

Path	/test
Method	GET
Parameter	None
Header	Content-Type: application/json
Payload	None
Return HTTP code	200 OK
Comments	Test service
Returns	It is Alive

Path	/subtitle
Method	POST
Parameter	None
Header	Content-Type: application/json
Payload	{ "userid":"<value>", "username":"<value>", "service":"text", "timestamp":"<value>", "text_options":{"font_color":"<value>", "font_type":"<value", "font_size":"<value>", "font_alignment":"<center | left | right>", "transparence_level":"1.0"} }
Return http code	200 OK 404 Not Found
Comments	Locate in the subtitle file (specified in the properties file) the position provided by the time parameter and continue reading the next subtitle entries formatting the snowmix messages according to the JSOn profile parameters received as a parameter
Returns	{ "statusCode": <HTTP Status Code>, "description": "<success or error description>", "data": "<value>" }

Payload example:

```
{
  "userid":"1", "username":"diego", "service":"text", "timestamp":"00:05:32",
  "text_options":{"font_color":"0 1 0", "font_type":"Sans bold",
  "font_size":"20", "font_alignment":"center", "transparence_level":"1.0"}
}
```