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

Project Name:

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

Deliverable D6.1

Demonstrator design, implementation and initial set of experiments

Date of delivery:	31/10/2016	Version:	1.0
Start date of Project:	01/07/2015	Duration:	30 months

Document properties:

Document Number:	H2020-ICT-2014-2 5G NORMA/D6.1
Document Title:	Demonstrator design, implementation and initial set of experiments
Editor(s):	Marco Gramaglia (UC3M)
Authors:	Ignacio Labrador Pavón, José Enrique González, Jorge Rivas Sánchez (ATOS), Alessandro Colazzo, Riccardo Ferrari, Stefano Sartorelli (AZCOM), Farid Sheikh (NOMOR), Ade Ajibulu (RW) & Stan Wong (KCL)
Contractual Date of Delivery:	31/10/2016
Dissemination level:	PU[1]
Status:	Final
Version:	1.0
File Name:	5G NORMA_D6.1.docx

Revision History

Revision	Date	Issued by	Description
1.0	28.10.2016	Ignacio Labrador	Final document

Abstract

This document describes the development of Work Package 6 activities up to date. The rationale behind the demonstrator selection is described: the envisioned plot, the target KPIs and the scenarios as described in D2.1. Detailed information for the 4 demonstrators proposed in 5G NORMA is provided: the Native Multi-service Architecture, the Service-aware QoE/QoS Control and Orchestration, the Secured Multi-Tenancy Virtual Network Resources Provisioning via V-AAA and the Online Interactive 5G NORMA Business Cases Evaluation Tool. Current status and work plan for each of them is also provided. Moreover, two already showcased demonstrators which were presented in the past Mobile World Congress 2016 and other very important international venues are described.

Keywords

Demonstrators, PoC, QoE/QoS Network Control, Network Slicing, Orchestration, Software Defined Network, Security, Economic Analysis, 5G, NFV.

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

Executive Summary

In order to demonstrate the breakthrough innovations we are proposing in the framework of the 5G NORMA project (NOvel Radio Multiservice adaptive network Architecture), we carefully defined a set of four demonstrators that will showcase very heterogeneous Key Performance Indicators (KPI) that can benefit from the introduction of the 5G NORMA concepts.

More specifically, we decided to develop two demonstrators with distinguishing marks of the work carried out by Work Packages (WP) 4 and 5, and two demonstrators exhibiting how the innovative concepts of 5G NORMA impact on the authentication and the business layers.

The first demonstrator, named Native Multi-service Architecture will display how the Software-Defined Mobile network Controller (SDMC) concept can be introduced into the state of the art Radio Access Network (RAN) stack. Tight hardware–software integration, simplified by a well-defined Application Programming Interface (API) capable of abstracting the complexity introduced by vendor-dependent specificity was achieved. The demonstrator will show how service-aware tailoring of Network Functions (NF) will enhance the Quality of Experience (QoE) perceived by users.-This document describes the implementation plan for the Demo 1 set-up by the two partners involved: Azcom and Nomor.

The second demonstrator, entitled Service-aware QoE/QoS Control and Orchestration, shows the feasibility of the introduction of the network slicing concept into the mobile network architecture. This demonstrator will leverage on open-source tools to showcase an innovative solution in the field of network control and orchestration. The service-aware orchestration, in particular, will adapt the network function placement according to the requested KPIs. This demonstrator is going to be integrated by University "Carlos III" of Madrid (UC3M) and Atos.

The third demonstrator will focus on novel mechanisms for the Authentication Authorisation Accounting functionalities. Based on open source software and implemented using commodity hardware, this demonstrator will show the feasibility of reproducible experiments in multi-tenant environment. This demonstrator will be developed by the King's College of London (KCL).

Finally, the fourth demonstrator will focus on the evaluation of business aspect of 5G NORMA through a web-based platform. This demonstrator will be developed by Real Wireless (RW).

This document also remarks the preliminary demonstrator that has already been presented during 2016 in different major events, getting a good success of audience.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Term	Description
3G/4G/5G	3rd/4th/5th Generation of Mobile Networks
3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ADC	Analog to Digital Converter
API	Application Programming Interface
ARM	Advanced RISC Machines
ARPU	Average Revenue Per User
AWS	Amazon Web Services
BBU	Baseband Unit
BSS	Business Support System
CCU	Central Controlling Unit
CDF	Cumulative Distribution Functions
CN	Core Network
CPRI	Common Public Radio Interface
C-RAN	Cloud RAN
DAC	Digital to Analog Converter
DNS	Domain Names Server
DSP	Digital Signal Processor
DTLS	Datagram Transport Layer Security
E2E	End to End
eNB	Evolved Node B
EM	Element Manager
EPC	Enhanced Packet Core
ETSI	European Telecommunications Standards Institute
EU	European Union
EUCNC	European Conference on Networks and Communications
E-UTRAN	Evolved UTRAN
FCAPS	Fault, Configuration, Accounting, Performance and Security
FG	Forwarding Graph
FPGA	Field Programmable Gate Array
GNU	GNU is Not Unix
GPL	GNU's General Public License
GPRS	General Packet Radio Service
GPU	Graphical Processing Unit
GSM	Global System for Mobile communications
GTP-U	GPRS Tunnelling Protocol for User Data
GUI	Graphical User Interface
H2020	Horizon 2020
HARQ	Hybrid Automatic-Repeat-Request
HD-Video	High Definition Video
HSS	Home Subscriber Server
HTML	Hypertext Mark-up Language
HW	Hardware
ICT	Information and Communication Technologies
IETF	Internet Engineering Task Force
IMTA	Interference Mitigation and Traffic Adaptation
IP	Internet Protocol
JSON	JavaScript Object Notation
KCL	King's College London
KPI	Key Performance Indicator
LFF	Large Form Factor
LL	Low Latency
LTE	Long Term Evolution
MAC	Media Access Control

MANO	MANagement & Orchestration
MBB	Mobile Broad Band
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
MNO	Mobile Network Operator
MVNO	Mobile Virtual Network Operator
MWC	Mobile World Congress
NASA	National Aeronautics and Space Administration
NF	Network Function
NFV	Network Functions Virtualisation
NFVO	NFV Orchestrator
NFVI	NFV Infrastructure
NORMA	NOvel Radio Multiservice adaptive network Architecture
NoSQL	Not Only SQL
NS	Network Service
NSO	Network Services Orchestrator
OAI	OpenAirInterface
ODA	Open Document Architecture
OSA	OpenAirInterface Software Alliance
OSM	Open Source MANO
OSS	Operations Support System
PC	Personal Computer
PCB	Printed Circuit Board
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
P-GW	Packet Data Network Gateway
PHP	PHP Hypertext Pre-processor
PoC	Proof of Concept
QCI	QoS Class Identifier
QoE	Quality of Experience
QoS	Quality of Service
R+D+i	Research, Development and Innovation
RAM	Random Access Memory
RAN	Radio Access Network
RANaaS	RAN as a Service
RF	Radio-frequency
RISC	Reduced Instruction Set Computer
RLC	Radio Link Control
RO	Resources Orchestrator
RRC	Radio Resources Control
RRM	Radio Resources Manager
SC	Scheduler Controller
SDMC	Software-Defined Mobile network Control(-ler)
SDMX	Software-Defined Mobile network Coordinator
SDN	Software Defined Networking
SDR	Software Defined Radio
S-GW	Serving Gateway
SISO	Single Input Single Output
SoC	System on a Chip
SQL	Structured Query Language
SW	Software
TDD	Time Division Duplex
TLS	Transport Layer Security
TTI	Transmission Time Interval
UC3M	University “Carlos III” of Madrid
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral

UTRAN	UMTS Terrestrial Radio Access Network
V-AAA	Virtual AAA
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtualized Network Function
VNFM	VNF Manager
WP	Work Package

1 Introduction

1.1 Objectives of the document

Demonstrators and the work done by Work Package 6 are a fundamental part of the 5G NORMA activities. The project aims at introducing novel concepts (i.e., the five pillars) into the design of future 5G mobile network architecture:

- Adaptive (de)composition and allocation of mobile network functions
- Software-Defined Mobile Network Control and Orchestration
- Joint optimization of mobile access and core network functions
- Multi-service and context-aware adaptation of network functions
- Mobile network multi-tenancy

To both prove the feasibility of the introductions of these novel concepts inside the network architecture and showing the advantages they can provide, we carefully designed a set of demonstrators. Their goal is hence to showcase the heterogeneous set of Key Performance Indicators that can be enhanced by the introduction of the 5G NORMA innovations.

The rationale behind the definition of the demonstrators followed two principles: i) the set of demonstrators should prove a large subset of the introduced innovations and, ii) the demonstrators should reflect the work done by the other WPs.

Therefore, we implemented 2 early demonstrators that have been showcased at the Mobile World Congress (MWC) 2016:

- Demo MWC16-1: Hardware (HW) demo developed by AZCOM which demonstrates the advantages of the functional decomposition and relocation of network functions.
- Demo MWC16-2: Software (SW) demo developed by NOMOR, focused on the context aware reconfiguration.

In addition, 5G NORMA designed four main demonstrators, whose development is still ongoing. They are designed to represent a broad part of the 5G NORMA innovations, and their current development is presented in this document.

In these four, the first two demonstrators are directly related to the 5G NORMA pillars, respectively Demo 1 and 2. The last two, instead, are related to other KPIs and other WPs:

- Demo 1: Native Multi-Service Architecture developed by AZCOM and NOMOR. It introduces the concept of Software-Defined Mobile Network Control and Multi-service and context-aware adaptation of network functions.
- Demo 2: Multi-slice service aware orchestration, developed by UC3M and ATOS. It will introduce the concept of Adaptive (de)composition of network functions with Software Defined Mobile Network Orchestrator. It also introduces novel concepts of network sharing and network slicing.
- Demo 3: developed by King's College of London, on the authentication, authorization and accounting aspects of the architecture. This work is the outcome of Task 3.3 from WP3.
- Demo 4: This differs from the above demonstrators since it is not hardware-based or demonstrating functionality. Instead this **demonstration and evaluation tool** developed by RW, will be used to allow assessment and dissemination of the economic impact of the 5G NORMA innovations. In this case, the demonstrator is one of the outcomes of WP2.

Most of the presented demonstrators rely on Open Source software, which is not only a choice driven by their design, but it will also enhance their reproducibility and impact, increasing the footprint of the 5G NORMA project.

Throughout this document, we describe how the demonstrators are structured as well as their envisioned outcome, and we list the steps that will lead to the final definition. We will also discuss the roadmap for the demonstrators' integration (when expected, for Demo 1 and Demo 2) and the preliminary outcomes that have already been obtained.

1.2 Structure of the document

This document is a summary of all the WP6 activities up to the delivery date and it is structured as follows.

Section 2 describes the consistency of the demonstrator within the 5G NORMA architecture. The different demonstrators will tackle several aspects of the 5G NORMA work, ranging from the re-definition of wireless network functions up to security and economic evaluation. We also describe the existing links between the works carried out in the different work packages.

Then, in Section 3, we describe the activities that have characterized the work of the work package (WP) up to now, being the two preliminary demonstrators presented at Mobile World Congress and the European Conference on Networks and Communications (EUCNC) conference the most important contributions.

The core of the document is Section 4, which describes in detail the work planned and already performed by each partner of this WP. Subsections are divided into demo plot, the description of the novelties to be demonstrated, the relevant KPIs (as described in D2.1), and the current status of the work. Additional considerations about the integration of the demonstrators (planned for Demos 1 and 2) are provided in Section 4.5.

Section 5 describes possible improvements and expected outcomes of the partners. Finally, Section 6 concludes the document.

2 Coherence with the 5G NORMA innovations

In the following sections, we describe the relationship between the different demos and the 5G NORMA architecture as it is currently defined, and also, the connection between each demo and other work packages.

2.1 Architecture consistency

Although the 5G Norma architectural definition is still work in progress while writing this document, some of the most important architectural principles have already been defined. Hence, the following table table shows how the different demos are aligned with the main architectural definitions:

Demo 1: Native Multi-Service Architecture	This joint demo (Hardware & Software) presents a proof of concept for (de)composition of NFs, which is one of the architectural key principles in 5G NORMA. It demonstrates how certain network functions (the routing components) could be decomposed and executed directly on the System on Chip (SoC) base station hardware in order to reduce latency and increase throughput. It also incorporates an SDMC-like software component, which is one of the main building blocks defined in the 5G NORMA architecture (WP3 & WP5). The adjustment in throughput and latency will be computed on that SDMC component, which will send commands towards the Evolved Node B (eNB) using a specific communication protocol to (re-)configure the placement of the network functions (the MWC'16 demos MWC16-1 and MWC16-2 previously mentioned in the introduction are a first approach to this demo, but working separately, i.e., without a Hardware/Software integration).
Demo 2: Service-aware QoE/QoS Control	This demo demonstrates two main architectural principles: Network Slicing and Virtualized Network Functions (VNF) Orchestration. Network Slicing is one of the design principles for the 5G NORMA architecture as stated in WP3. On the other hand, the orchestration modules used for this demo are basically a specific implementation of the European Telecommunications Standards Institute Networks Function Virtualization MANagement and Orchestration (ETSI NFV MANO) framework, which basically corresponds with the architectural blocks described in WP3 for the Management and Orchestration layer.
Demo 3: **Secured Multi-Tenancy Virtual Network Resources Provisioning via V-AAA**	This demo directly connects with the architectural security principles regarding the Unified Authentication, Authorization and Accounting –AAA– System as they are described in WP3.
Demo 4: Online Interactive 5G NORMA Business Cases Evaluation Tool	This demo evaluates the economic impact of the new 5G NORMA architecture as a whole, considering some of its key principles with different metrics, such as multi-service and multi-tenancy use cases.

Table 2-1. Relation between demos and the 5G NORMA architecture.

2.2 Connection with other WPs

There is a tight relationship between the four demos described in this document and the other technical WPs in the project (WP2, 3, 4 & 5). The following table summarizes the different assignments and relationships with other work packages for each demo (a bigger font represents a more direct relationship):

Demo	Partners	Related innovations	Related WPs
Demo 1: Native multi-service architecture	Azcom Nomor	Adaptive (de)composition and allocation of mobile NFs (software and hardware)	Mainly WP4 (WP2, WP3,WP5)
Demo 2: Service-aware QoE/QoS control	UC3M Atos	Network slicing Virtualization Infrastructure	Mainly WP5 (WP2, WP3, WP4)
Demo 3: Secured Multi-Tenancy Virtual Network Resources Provisioning via V-AAA	KCL	Basic implementation of authorization and authentication using 5G NORMA architecture	Mainly WP3 (WP2)
Demo 4: Online interactive business cases evaluation tool	Real Wireless	Demonstration and evaluation tool. This web-based evaluation tool will have an online GUI to showcase the economic benefits of 5G NORMA	Mainly WP2 (WP3, WP4)

Table 2-2. Assignments and relationship with WPs

Demo 1 primarily focuses on WP4 (Flexible RAN design), i.e., we are providing a HW implementation of a base station able to execute VNFs on a chip. In addition, it is also influenced by WP2 (use cases), WP3 (overall architecture – an SDMC component is developed here), and WP5 (QoS/QoE control – based on latency and throughput measurements).

Demo 2 showcases many of the functionalities investigated in the framework of WP5 (Flexible connectivity and QoE/QoS management). Novel concepts regarding the decomposition of network functions and their service-aware orchestration are the fundamental pillars of this demonstrator. Finally, this demonstrator will feature enhanced resource sharing techniques among network slices. It is one of the main outcomes of WP3 that is implemented by some of the QoS/QoE management algorithms proposed in WP5. It further connects with WP2 (use cases) and WP4 (it deploys also an eNodeB and tries to split the radio stack in different VNFs which are placed on different network slices).

Demo 3 directly connects with the architectural security principles from WP3. It is also related with the use cases defined in WP2.

Finally, Demo 4 (this is more correctly termed a demonstration and evaluation tool) is mainly related with concepts in WP2, but it is also influenced by the overall architecture concepts in WP3 (it evaluates the socio-economic impact of the new 5G NORMA architecture as a whole considering some of its key principles with different metrics, such as multi-service and multi-tenancy use cases) and the radio related concepts described in WP4.

Furthermore, besides the information received from the different WPs, WP6 contributes also providing feedback to the other WPs about relevant facts and limits found during the demo's development.

3 Activities up to date

The work already done by Work Package 6 has been focusing on two main items:

- The participation in international events, in which preliminary demonstrators of the 5G NORMA innovations were presented.
- The careful design, planning and initial implementation of the final demonstrators.

We remark that, besides the very short time frame, the impact achieved by the work of WP6 has been very notable. Early stage demonstrators have been presented at two major events held during 2016: MWC'16 (held in February 2016) and EuCNC'16 (held in June 2016). In both events, 5G NORMA demonstrators received positive feedback and were positively accepted by the audience.

The showcased demonstrators were a proof of concept of some of the 5G NORMA pillars such as the adaptive decomposition and allocation of network functions and their multi-service and context-aware adaptation. Although these demonstrators were developed in an early stage of the project, their success demonstrated the feasibility and impact of the 5G NORMA innovations in both large scale and hardware-based scenarios.

Therefore, this section describes the public demonstrators presented at MWC'16 and the EuCNC conferences. They are basically a software demo focused on the context aware reconfiguration principle, and a HW demo to demonstrate the advantages of the functional decomposition and relocation of network functions.

3.1 Context Aware Network Reconfiguration

NOMOR has extended its network simulator to study and exhibit the advantages of the novel network architecture with multi-service and context-aware scheduling. The output of this task was the software demo presented at MWC'16 and EuCNC'16, on behalf of 5G NORMA project. 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 by placing the Radio Resource Management (RRM) functionalities into the central cloud data centre enables implementation of interference mitigation strategies using centralized 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 service.

Design Description and Objectives

The main objective of this demo is to represent the performance of the flexible and service-aware radio access network as one of the key innovations of 5G NORMA. The ability to simulate Centimetre-Wave (cmW) frequency band in addition to supporting asynchronous HARQ (Hybrid Automatic-Repeat-Request) and shorter Transmission Time Interval (TTI) are the 5G-oriented features of the simulator. The showcase consisted of two key parts: the system-level simulator and a 3D GUI (Graphic User Interface) visualizer. The system-level simulator runs on a Linux machine while the 3D GUI visualizer runs on a Windows platform. The demo is basically a software-based showcase with no specific hardware required for running and demonstrating. For the overall demo, two computers are required, i.e. one for the simulator and one for the visualizer. These two elements are coupled to each other and communicate via an Internet Protocol (IP) interface over the network card.

Furthermore, 5G NORMA has proposed that the network functional elements can be located either in edge or central cloud based on the user service requirements (e.g., data rate and delay budget) as well as network characteristics (e.g. backhaul latency). The (de-)centralized RRM algorithm will optimally adapt the user resource utilization on the traffic load fluctuations and inter-cell interference conditions. For the case of central-cloud, the effect of centralized RRM is demonstrated by clustering multiple cells and jointly deciding a TDD (Time Division Duplex) configuration in order to maximize system performance. The clustering will be done based on the service demands on the relative cells. For cells serving users with only time-critical services, clustering may be disabled to avoid increasing delays due to non-ideal backhaul between the cells forming a cluster. In contrast, for the cells with only throughput-critical services, clustering minimizes interference and maximizes the network throughput, consequently.

As proof of concept, the demo demonstrated the effect of the aforementioned algorithms for different services delivery in the 5G RAN. The context- and service-aware algorithms are demonstrated through two typical services, which are HD-Video streaming (i.e., a service with high data rate required per session) and V2X (i.e., a set of services with very low delay budget and low traffic volume). In addition to the network and terminal throughput, the performance of hazard avoidance algorithm (i.e., a time-critical algorithm) for automotive users is used as another important evaluation metric.

Demo Scenario Layout

The scenario contains two main regions formed inside a small part of the New York Manhattan city layout as it is shown in Figure 3-1. In each of these regions different types of service profiles and users are placed. The first region, as shown on the left side of Figure 3-1, contains mainly pedestrians using the HD-video service and closely placed eNodeBs. The antennas are mounted on top of the light posts and highlighted.

The second region of interest is on the right side of figure, where there are some cars using the V2X service. These cars communicate over the 5G network in order to warn each other about possible hazards. It is worth noting that the mobility of users is defined based on the city layout using SUMO mobility models [20], which defines movement of cars and pedestrians with different constraints like streets, lanes, traffic signals, or pedestrian crossings.

Figure 3-1. Overall Scenario Layout.

In the first region with a dense deployment of small-cells and HD-video streaming service, the network function blocks are placed in the central-cloud in order to perform joint dynamic TDD selection with cell-clustering. The increase in throughput achieved by applying network reconfiguration and dynamic TDD selection is shown in Figure 3-2 below. The bars in the bottom left show the session network throughput. The blue bar shows the average session downlink throughput after the network reconfiguration using the 5G NORMA innovations, while the violet bar illustrates the same metric without the 5G NORMA proposal. The increase in throughput is due to the joint coordination among cells resulting from a central-cloud configuration.

Figure 3-2. City region-1 with HD-video streaming service.

The second region in the city considers a V2X application through 5G networks, as shown in Figure 3-3 below. The demo highlights the effect of locating core network functionality closer to the end-user, i.e., in the edge-cloud. In this part of the city, the 5G NORMA proposes to reconfigure the network as an edge-cloud because of the high traffic demand for time-critical services in this area. In the figure, the communication between two cars is highlighted by the green and yellow arrows, for downlink and uplink transmissions respectively. Here we focus mainly on the end-to-end latency KPI, since the V2X service is a time-critical service. According the achieved results, reconfiguration network functional elements into edge-cloud decreases the end-to-end latency for the connected terminals to relative cells.

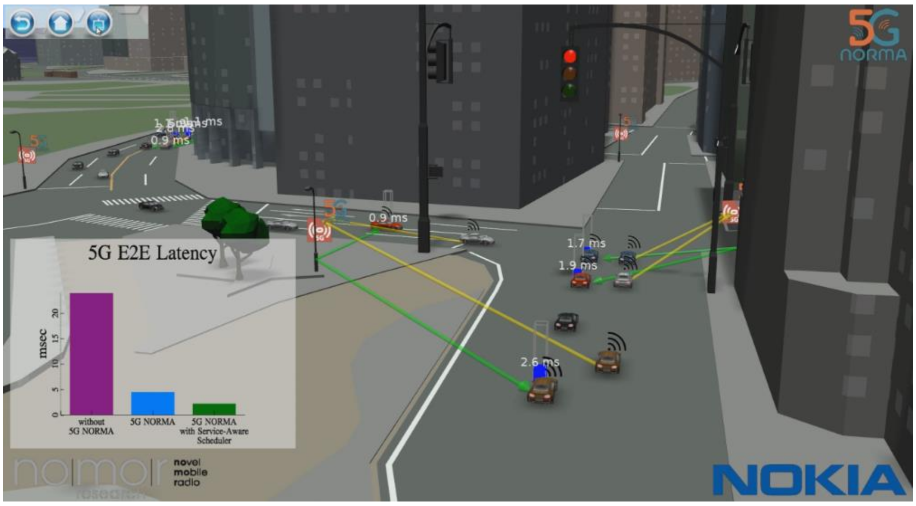


Figure 3-3. City region 2 with V2X communication.

Development Activities

As mentioned, the network simulator has been extended to study the aforementioned scenarios according to the 5G NORMA proposals. The key extension tasks in the network simulator and the 3D-GUI visualizer are:

- **V2X Traffic Model Implementation:** V2X service was implemented inside the simulator. In this model, the terminal in the car sends a small random-size packet in the uplink when receiving a road traffic trigger. As soon as the eNodeB receives this packet at the Media Access Control (MAC) layer, it forwards the packet to the target car.
- **Multicast traffic model:** The traffic model described above is a unicast traffic model. However, during the course of demo development, it was required to support multicast V2X communications, i.e., a user sends a message in the uplink and it is received by multiple User Equipment (UE) in the downlink. For this purpose, the network simulator has been extended to support multicast.
- **Vicinity-based Hazard Trigger Model:** In this task, a vicinity event-based trigger for the generation of data traffic while identifying the traffic safety critical situations is implemented. The objective was to develop a simple application for car-to-car communication for exchanging warning messages among cars. When one car detects a dangerous situation, it warns other cars about it. For the realization of this concept, a logic that identifies the case of the dangerous situation when a car suddenly brakes and could cause possible collision with preceding car is developed. The main logic is to identify the UEs (cars) in danger when their distance to the preceding cars falls below a certain safety margin. This results in triggering hazard messages to be sent in the uplink to the serving eNodeBs. The message is supposed to be sent by the car that is braking and is in front of the other following cars. The logic also identifies the cars, which could potentially be exposed to danger, i.e., the cars which are driving in from behind. These cars receive an alarm message sent by the car braking in front.
- **Backhaul Latency:** The goal is implementing backhaul latency on the Nomor's system-level simulator to be used with enhanced Interference Mitigation and Traffic Adaptation (eIMTA) clustering decisions. Ideally, there should be a delay (or latency) between an eNodeB sending load information to the master eNodeB and then master eNodeB sending decision information to the controlled eNodeBs. However, it is possible to combine these two into one single delay since the controlled eNodeBs will keep their old eIMTA pattern until the new decision is received. This latency was made configurable from a configuration file and also from the GUI. All clusters will have the same delay value. For those special eNodeBs, which are not in any cluster, zero delay will be considered for eIMTA decisions.
- **Service-aware TDD pattern assignment:** In the system-level simulator, the clustering between different eNodeBs is performed based only on the vicinity of different eNodeBs. For demonstrating the effect of service-aware network architecture, an element of service-awareness was added to the clustering logic.
- **Implementation of a new TDD pattern:** In Long Term Evolution (LTE), there are seven different uplink-downlink TDD patterns defined in the standards (source: 3GPP - 3rd Generation Partnership Project- 36.211). These TDD patterns have a *downlink-to-uplink switch periodicity* of about 5ms to 10ms. As part of the work in 5G NORMA, a new TDD pattern has been designed and implemented, which has a periodicity of only 1ms. This TDD pattern shall help reducing latency because of the faster switching periodicity.
- **Mobility mode and modifications of mobility:** Mobility in the system simulator is controlled by external mobility traces. For this purpose, SUMO mobility models are used. In SUMO, first, the street layout needs to be imported from OpenStreetMaps and then fine-tuned according to the actual scenario. Then they can be used to generate mobility traces. Due to the scenario complexity and the mismatch between OpenStreetMaps and the 3D city model, the street and buildings data usually does not

match perfectly each other. In order to match them perfectly, some manual changes were made on the traces generated from SUMO.

- **Extension of the 3D-GUI visualizer:** in addition to network simulator, the following new features have been added to the 3D-GUI visualizer:
  - **Development of KPI plotting tool:** presenting some of KPIs in the visualizer in form of 2D-bars was needed. However, it was not possible to integrate this feature inside the 3D-GUI. Therefore, another application was developed for plotting these 2D bars. This application was developed based on the MathGL tool [17]. MathGL is a set of C-libraries for printing high-quality scientific data. It enables fast handling and plotting of large data arrays. It can export graphics to bitmaps and vectors. These libraries were used to make bar plots and then export them as images. MathGL license is the GNU General Public License (GPL).
    The plotting tool application connects with the system simulator via an IP interface and periodically gets KPIs values. These values are then plotted. It generates an image file for the plot and at each new data refresh it overwrites the existing image file with the new data. This image file is then read and placed on top of the 3D-GUI display as an overlay.
  - **Adding traffic lights:** The mobility files generated for cars in the Manhattan scenario were based on SUMO where cars stop at junctions based on traffic lights. It was required to implement a feature to show traffic lights and to switch them based on the actual traffic lights in SUMO. In a first step, the default switching frequency of traffic lights from SUMO was modified to represent them in the 3D GUI. Then, modifications were made to the models to change the colour of each of the signals from off to red, yellow or green. Unlike other elements, which are controlled from the system simulator, traffic lights are stand-alone entities in the 3D GUI.
- **Implementation of New KPI Features:** The system simulator has some useful KPIs already. But some more KPIs were needed for the MWC'16 demo. The following KPIs were developed:
  - **KPI separation based on groups:** It was required to separate KPIs based on groups. Hence, a group KPI features was first implemented when the KPIs for UEs belonging to one group was aggregated. In the user definition scenario script each user can be assigned to a group. This feature can be extended to be used for almost any existing KPI in the simulator.
  - **End-to-end latency:** For car-to-car communication, a special KPI was needed for analysing the time delay between sending a packet by one UE and when it reaches the other UE after passing through the infrastructure. For this, we implemented a new KPI, which represents the sum of the time the packet takes to travel from the source-UE to the eNodeB, and from the eNodeB to the target-UE at the Packet Data Convergence Protocol (PDCP) layer. This KPI was first implemented per UE but was then extended to have it for a group of UEs to analyse end-to-end delay only for a set of UEs.
  - **Mean session throughput:** A new KPI representing the average of the session throughput of UEs has been implemented.

MWC'16 Demo: Setup and Reception

The demo at Mobile World Congress 2016 was presented 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 monitor screen provided by the MWC organizers. The setup is shown in the following figure:

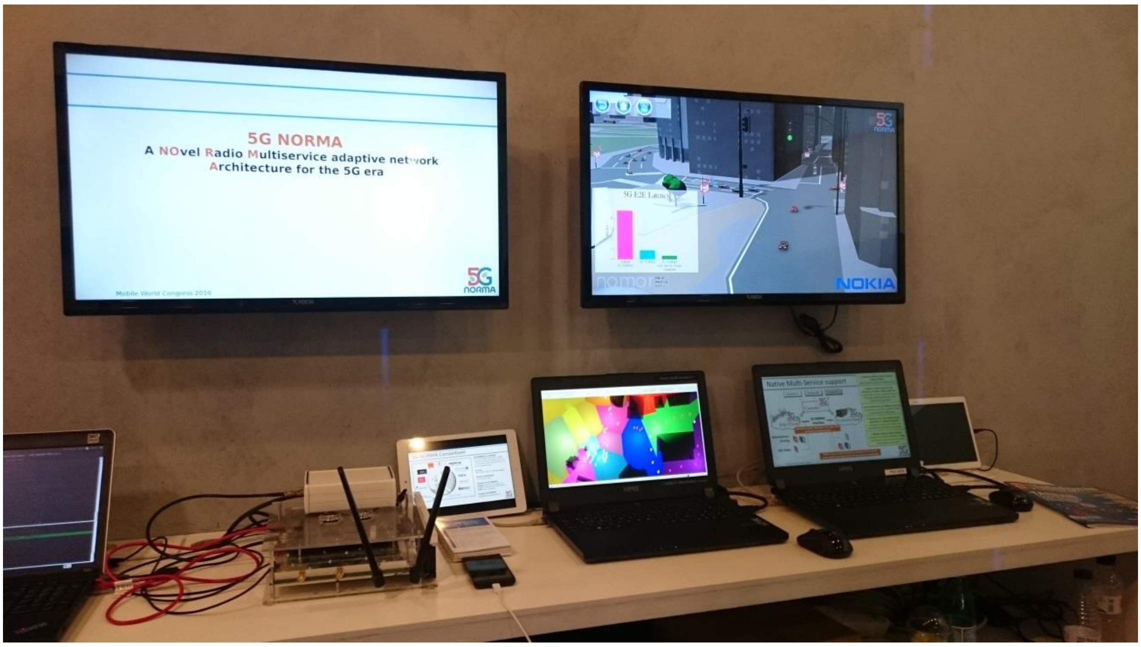


Figure 3-4. MWC'16 Setup: The software demo presented on the monitor (top right), with Simulation and GUI running on the two laptops (bottom right).

The 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 great number of relevant audiences 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 3-5. Demo being explained to the European Union (EU) commission delegates.

Figure 3-6. The media speech by EU commission delegates in front of the demo booth.

3.2 Hardware Demo

An integral part of the validation process of the 5G NORMA project is the realization of Proof of Concept (PoC) demonstrators. PoC demonstrators show that a certain concept or approach is technically feasible and can be implemented with reasonable efforts.

Azcom Technology has extended its base station, see [18], to study and show the advantages of the functional decomposition and relocation of network functions and its impact on the network end-to-end latency figure. It also showed practically to a non-technical attendee the benefit of the low network latency and the kind of applications that can be enabled. On behalf of the 5G NORMA project, the hardware demo was also presented at MWC'16 and EuCNC'16.

Figure 3-7. A picture of the 5G NORMA hardware demo at the EU Commission booth at the Mobile World Congress 2016 in Barcelona.

Design Description and Objectives

Low-latency service is provided by the network functions relocation, one of the baseline concepts of 5G NORMA. A set of network functions, belonging to both RAN and Core Network (CN), critical in terms of latency, are moved from the central cloud to the edge cloud in order to avoid backhaul latency and unnecessary hops in the path to the CN.

The low latency network slice is so instantiated activating the needed network functions in the appropriate clouds. The most critical ones, e.g. routing, are moved from the core network to the edge cloud, close to the user. For even more stringent latency requirements, the network functions can be also placed closer to the user, on an edge cloud located on the base station itself. In Figure 3-8, three different levels of cloud are depicted, depending on the latency required, for three different types of services. The corresponding network slice is kept until the service ends, afterwards its resources can be reallocated for other purposes.

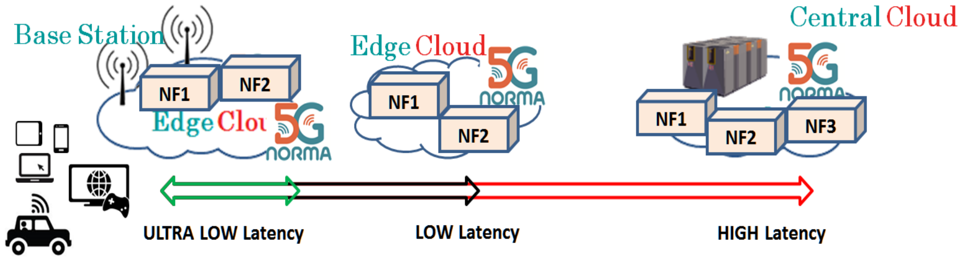


Figure 3-8. Instantiation of network functions in different clouds depending on the latency requirement of the service.

Therefore, according to 5G NORMA proposals, the basic idea of the demo is to show the latency impact in driving a scale model rally car using a commercial tablet as the steer, both connected to the LTE eNB, as depicted in Figure 3-9 below.

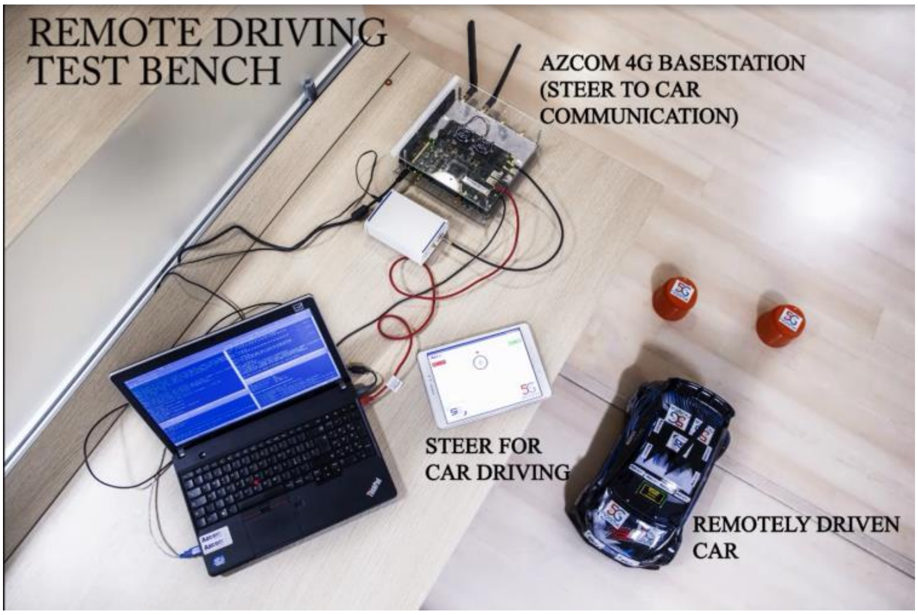


Figure 3-9. Hardware demonstration of the low latency benefits.

To enable local routing of the communications between the car and the tablet, the Enhanced Packet Core (EPC) routing components usually deployed in the CN as depicted in Figure 3-10, were moved into the eNB baseband board.

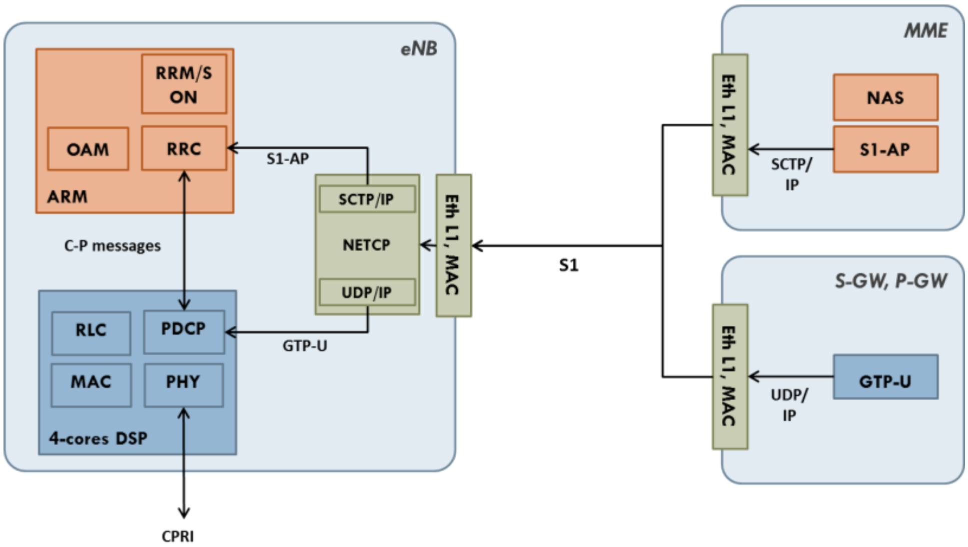


Figure 3-10. EPC routing components deployed in Core Network.

Specifically, the S-GW was hosted by the base band SoC as depicted in Figure 3-11, with the objective to guarantee the lowest latency possible with this HW setup, which in turn offers to the end user a good driving experience.

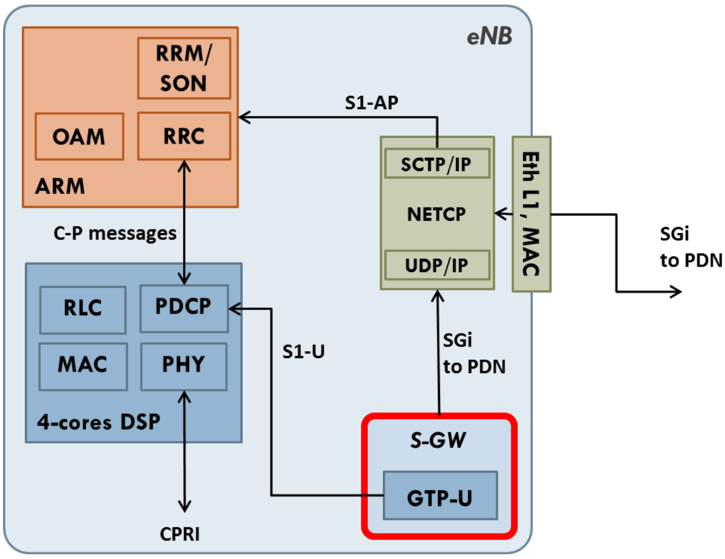


Figure 3-11. Low latency eNB hosting core network functions.

The LTE air interface was used in the demo, but packet sizes and RLC[2]/PDCP parameters were set so that the packets go through the protocol stack as quickly as possible. Moreover, after moving the EPC components into the eNB interface, the S1 interface (now internally managed) continues using the GPRS Tunnelling Protocol for User Data (GTP-U) even if all the components are running on the same SoC, hence no protocol stack optimization was foreseen.

Demo Scenario Layout

Two different situations were demonstrated. The first one mimicked a commercially deployed LTE network in which the EPC routing components were located in the core network and an average End to End (E2E) latency of hundreds of ms was experienced. Latencies in 4G LTE today are up to 200 ms in the worst cases as reported in Table 3-1 [19]:

[2] Radio Link Control

Best 10%	21-43 ms
Median	33-75 ms
Worst 10%	47-200 ms

Table 3-1. 4G LTE Ping results in the world.

In the second scenario, the EPC routing components (S-GW) were moved into the eNB baseband board.

The final application has shown the benefit of the low latency by controlling the model rallye car. In the first scenario the high latency reduced the driving quality, as the command arrived at the model rally car with great delay. In the second scenario the control feeling was very good since the response of the car was immediate. The measured latency in this scenario was from 16 to 18 ms.

Development Activities

At the beginning of the 5G NORMA project, Azcom already owned various products: eNB base station platforms, a fully operational eNB software, and a lightweight EPC application. Briefly, Azcom performed the following key tasks for supporting the above-mentioned scenario:

- **EPC software porting**: the EPC software components have been ported on an Advanced Reduced instruction set computer Machine (ARM) platform, then they run on the SoC of the baseband board, enabling local routing
- **Remote controlling the car via LTE:** Azcom has designed a Printed Circuit Board (PCB) for remote controlling the car via LTE. The PCB integrates an LTE modem (Telit LE910, see Figure 3-12) and a microcontroller (PIC32MZ). In addition, a sort of automotive protocol has been designed and implemented: the car stops if the connection is lost. This board was placed into the car.

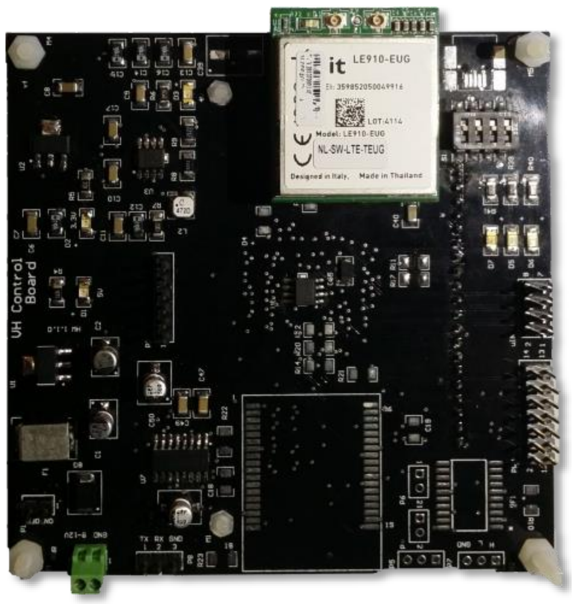


Figure 3-12. Telit LTE Modem.

- **Car Control App**: The control software was developed by Azcom for the Android platform and it runs on a commercial LTE tablet (Samsung Galaxy tab A 9.7" LTE). The application allowed the following key operations: start/stop the car engine, accelerating/braking the car and steering the car (see Figure 3-13):

Figure 3-13. Android Car Control App.

- **Integration and testing:** All hardware and software components were integrated and tested, completing the final demo.

4 Demo Plan

In the following, we describe the details of the four demonstrators planned for the 5G NORMA project.

4.1 Demo 1. Native Multi-Service Architecture

Demo 1 will be based on Azcom's base station with a well-defined programmable interface and a centralized controller, provided by Nomor, able to program the base station through this interface. The centralized controller implementation will follow the 5G NORMA Software-Defined Mobile networking Controller concept, which allows for easily modifying its functionality, thus providing a very high degree of flexibility. Azcom's software blocks can expose a custom interface, through which the external SDMC agents can instruct the blocks themselves on the expected behaviour and network functions to run.

4.1.1. eNodeB

The Native Multi-service Architecture demo aims to show the PoC of flexibility, adaptability, intelligence, and service-aware (de)composition of network functions and services. The basic idea is to show the impact of different deployments of network functions, using one or more end-user services.

Showcase

The demo is composed by one or more eNodeBs, provided by Azcom, and the novel network element known as Software Defined Mobile Network Controller, provided by Nomor. This controlled is supposed to be part of the network architecture, to monitor and control the hardware eNodeB besides taking care of the network (re)configuration.

The demo is intended to show the effect of the network reconfiguration done by the SDMC to improve the user experience in terms of Quality of Service (QoS) in different interference conditions. Two users connected to the same eNodeB are using two different services: a low latency (LL) service and Mobile Broadband (MBB) service. When an emulated interference signal reduces the QoS of users, the SDMC reconfigures the network to guarantee the required quality for those services.

Architecture

The key elements of this demo are the hardware eNodeB provided by Azcom and the SDMC component developed by Nomor.

The eNodeB is provided with an integrated Fast Scheduler Controller (Fast-SC), while the SDMC executes a Slow Scheduler Controller (Slow-SC). These two elements are connected to exchange a set of dedicated messages by means of a communication protocol, implemented on the eNodeB side, as shown in Figure 4-1. The fast scheduler on the eNodeB works in real-time. On the other hand the SDMC defines a scheduling policy based on the feedback from the eNodeB. Thus the eNodeB adapts its scheduler based on the feedback received from the SDMC. This requires implementing some modifications on the legacy 4G eNodeB protocol stack.

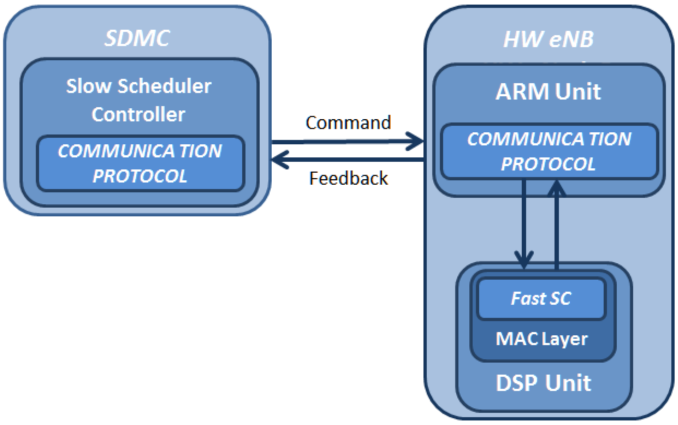


Figure 4-1. Architecture of the Hardware Demo.

Hardware / Software configuration

The demo will include one or more eNodeBs and one SDMC component. The target eNodeB shall serve two users, and it is connected to the SDMC over an IP interface; it shall also communicate configurations and feedback messages, as depicted in Figure 4-2. The interference is emulated, disturbing the services provided to the users. In practice, the SDMC is a software application running on a Personal Computer (PC).

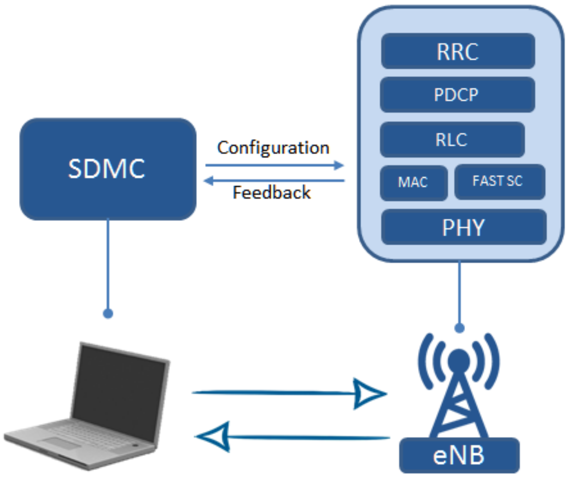


Figure 4-2. Key elements of the showcase.

The eNodeB used for the demo will be the AZCOM AZB-NC0M-4304[3] baseband unit (see Figure 4-3):

Figure 4-3. AZCOM AZB-NC0M-4304 baseband unit.

It consists of high performance processors (Digital Signal Processor –DSP–, Graphical Processing Unit –GPU– and Field Programmable Gate Arrays –FPGA–) with multiple data

[3]Data sheet, Azcom Technology, 2016. *http://www.azcom.it/wp-content/uploads/azb-ncom-4304-1u-digital-enodeb-base-band-unit.pdf#page=1&zoom=auto,-70,473*

movement interfaces that provide a full featured Baseband Unit (BBU) supporting both: centralized and distributed operation. It contains one of the most advanced and powerful multi-core SoC to run and manage the L1, L2 and L3 stacks.

The radio unit will be the AZCOM ngSCBP-RF[4], depicted in Figure 4-4. This module is usable as Remote Radio Head with Common Public Radio Interface (CPRI). Independent configuration of each Radio-frequency (RF) pipe and RF channel allows the ngSBP-RF card to support simultaneous processing of different standard and operating scenarios.

Figure 4-4. AZCOM ngSCBP-RF 4×4 SMALL CELL RF BOARD

In the following table, the key hardware features are reported:

Feature	Value
Operational frequency	from 700 MHz to 2700 MHz
Channel bandwidth	from 1.4 MHz up to 20 MHz
Minimum output power	27 dBm
EVM	<5%

Table 4-1. AZCOM ngSCBP-RF parameters.

The users are two LTE dongles, which will be inserted in a portable test box[5] (see Figure 4-5). The box is shielded from the interference coming from the environment, a key condition for the practical implementation of the demo, since the interference from external sources should be under control.

[4]Azcom Technology, 2016. *http://www.azcom.it/index.php/products/hardware-platforms/small-cell-radio-unit*

[5]Website, Ramsey, *http://www.ramseyelectronics.com/index.php*

Figure 4-5. Portable test box.

This demo will use a customizable eNodeB software, able to handle several commercial 4G UEs, and Azcom's lightweight EPC application (Packet Data Network Gateway –P-GW–, Serving Gateway –S-GW– and Mobility Management Entity –MME–). These software components are proprietary, so the source code won't be released. The additional software components specifically implemented in order to support the external SDMC controller won't be released either, since they are special additions and modifications to the already existing proprietary software.

Objectives and KPIs

The main objectives of the demo proposed by Azcom and Nomor are: i) showing that a novel design of the network architecture with SDMC would produce gains also over legacy network architectures like LTE, and ii) showing that this concept is technically feasible and can be implemented with reasonable efforts.

Hard KPIs: The main KPIs that shall be used to estimate the system performance are:

- The average session throughputs,
- The end-to-end latency.

In this demonstrator, two real users are experiencing two different services: a low latency service and mobile broadband service. When an emulated interference signal reduces the QoS of the users, the SDMC reconfigures the network to guarantee the required quality of both services affecting the above-mentioned KPIs.

Soft KPIs: The soft KPI associated to this demonstrator is mostly related to the new design proposal based on the SDMC concept defined in the 5G NORMA architecture: the network can adapt itself to the demands and requirements of frequently varying network conditions by means of network programmability.

Scenario Design

A preliminary scenario definition has been provided and is depicted in the following figure:

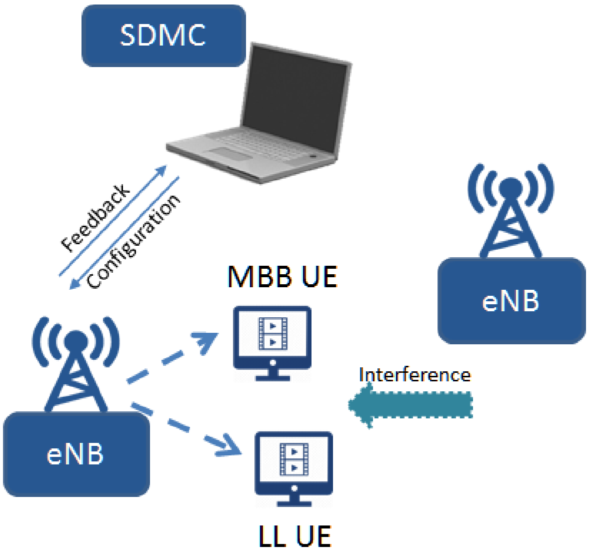


Figure 4-6. Scenario Design.

As mentioned, the eNodeB is serving two different users requiring two different services: mobile broadband and low latency. At the beginning, the eNodeB transmits two video streams to both users. In the interference scenario, the QoS experienced by both users in terms of throughput and video quality will be decreased. The SDMC component should be aware of the increased interference, sending then a feedback message to the eNB Fast Scheduler. The eNB should react by keeping the desired QoS of the two different services. Hence, the MBB user should continue to experience a high video quality, at cost of increased latency. On the other hand, the LL user will experience a reduction of video quality but keeping the latency at lowest level.

Status and Time Plan

The hardware part of Demo1 is organised in subtasks as shown in the following table:

Hardware Specific Tasks (Azcom part)	
Design of hardware prototype (from task T6.1)	D1.2.1
└ Initial design of the demo	D1.2.2
└ Updating of the design of the demo	D1.2.3
To provide one eNodeB sets responsive to the commands received from SDMC	D1.2.4
Provide an Evolved Packet Core	D1.2.5
Description of Azcom eNB (BBU based on Texas Instruments Multicore DSP+ARM SoC)	D1.2.6
Description of Azcom eNB Software deployment	D1.2.7
Implementation of Az side of the interface towards the SDMC	D1.2.8
SDMC support on Az software	D1.2.9
Testbench Az premises	D1.2.10
Integration with Nomor software SMDC	D1.2.11
Final use cases testing	D1.2.12
Get (purchase) all necessary hardware	D1.2.13
Improvement of the demo	D1.2.14

Table 4-2. Subtasks of Demo1, Hardware part only.

The meaning of each task is as follows:

- D1.2.1 – D1.2.2 – D1.2.3:
  These tasks consist of the definition of the demo jointly with Nomor, including also the definition of a common interface between all hardware and software components.
- D1.2.4:
  Azcom will provide a real LTE eNodeB, with an implementation of the communication protocol to communicate with the SDMC.
- D1.2.5:
  Azcom is providing its fully customizable Evolved Packet Core.
- D1.2.6 – D1.2.7:
  Azcom provides technical documents of its eNodeB to Nomor.
- D1.2.8 – D1.2.9:
  In this subtask, the communication protocol and the interface is discussed and implemented on the eNodeB.
- D1.2.10 – D1.2.11 – D1.2.12:
  Here, all the demo components, both hardware and software, are integrated and tested.
- D1.2.13:
  Azcom is procuring all required hardware for the demo.
- D1.2.14:
  Azcom will evaluate any possible improvement of the demo.

A detailed time plan for the project is provided in Annex A at the end of this document.

4.1.2. SDMC Component

Showcase

The aim of this demo is presenting the proof of concept for a flexible, adaptive, intelligent, and service-aware (de)composition of network functions and services. The basic idea is to showcase the impact of different deployments of the network functions, using one or more end-user services. The showcase will be first developed as an Intermediate Demo, which can be presented at conferences and other important 5G events.

The showcase will involve one or more hardware eNodeBs that can serve multiple UEs over the radio access network. The hardware eNodeBs will form a set of cells to which the UEs can connect over the radio interface. In 5G Norma, a novel network element, known as Software Defined Mobile Network Controller, is envisioned as a part of the network architecture. This element will be an integral part of the showcase and will be developed by Nomor as one of their main contributions to this project. The SDMC monitors and controls the hardware eNodeBs beside taking care of network (re)configuration.

The functional deployment will be actually changed or emulated, depending on the software implementation complexity: in the cases in which the network function can be easily re-deployed, it will be done, otherwise the re-deployment effect will be emulated by changing radio or backhaul parameters, which have the same effect on the final service.

In practice, an SDMC algorithm for (de)composition of network functions to meet the diverse Quality of Service (QoS) requirements of different applications in real-time is going to be demonstrated. The showcase aims at demonstrating how flexibly the network architecture can improve the user-experience for different services. The services considered in this demo are also chosen to be 5G-oriented services, in order to keep it closest to actual 5G requirements.

The reconfiguration of network functionality between edge- and network-cloud is to be performed based on a number of parameters such as buffer-status, QoS Class Identifier (QCI) and service-type. Such reconfiguration should lead to an improvement in terms of latency and/or throughput of the end-user. The mapping between QCI and throughput/latency parameters shall be computed at the SDMC and the corresponding reconfiguration commands will be communicated to the eNodeB via dedicated command messages. Corresponding to this command, the throughput or latency shall be changed at the eNodeB to realize the effect of network reconfiguration.

Architecture

The key elements of this demo are the hardware eNodeB provided by AZCOM and a Software Defined Mobile Network Controller developed by NOMOR.

The SDMC connects to the eNB (in particular to L2) and provides two main functionalities:

- Monitoring and controlling of the radio transmission key parameters of the connected users, for example, the received signal strength, the active service-type and load in both downlink and uplink directions.
- Sending commands to (re-)configure the placement of the network functions in the edge-cloud, i.e., deployed at eNodeB with low latency and limited processing resources, and central-cloud, i.e., placed in the core network with relatively higher latency and more processing resources.

The demo will have two main components, the eNodeB with an integrated Fast Scheduler Controller (Fast-SC) and the SDMC with a slow-scheduler. The two components will be connected via a communication protocol for communicating commands and feedback as shown in Figure 4-1 in the previous section.

Hardware / Software configuration

The hardware configuration of the showcase shall comprise two main elements, namely eNodeB and the SDMC. The two elements are connected with each other over IP and shall communicate configurations and feedback messages. This is highlighted in Figure 4-2, in the previous section, where the block on the right side containing the Radio Resources Control (RRC) and MAC layers is shown to describe the internal architectural elements of the eNodeB. On the other hand, the SDMC will be a software application running on a laptop/PC, as highlighted in Figure 4-2.

The SDMC software is implemented using the C++ programming language and is initially deployed on a Windows platform. However, the SDMC is designed in such a way that it can be ported easily to a Linux platform if needed. The communication protocol between the two entities is designed based on the UDP[6]/IP protocol, taking care of decoding and serializing all feedback and configuration packets at SDMC and eNodeB.

Objectives and KPIs

One of the main objectives of this demo is to show how a novel design of the network architecture with an SDMC would be set up in the real-world for future deployment of 5G networks. The key objective is also to demonstrate the gains that such changes in the architecture would bring over legacy network architectures such as LTE. The new design proposed with SDMC is envisioned to produce a flexible network architecture where the network can adapt in order to meet the demands and requirements of continuously changing network conditions.

The main KPIs that shall be used to evaluate the system performance are the following:

- Average session throughput: The session throughput averaged over all UEs in the system. The session throughput is preferred because of the bursty nature of the traffic and longer idle time spans.
- End-to-end latency: This latency measures the time difference between events when the packet is sent by one UE and when the packet is received by the other UE; it is usually calculated at the PDCP layer of the receiver.

[6] User Datagram Protocol.

The main target of the demo is to show an improvement in these KPIs. An improvement in session throughput is expected to be achieved as a result of the introduction of the SDMC controller because SDMC can centrally control the two eNodeBs and therefore jointly assign resources and parameters to reduce interference and to improve throughput. On the other hand, an improvement or reduction in latency is expected to be achieved because of the ability of SDMC to (re)configure network functions in the edge-cloud which results in a lower latency and performance improvements for time-critical services.

Scenario Design

A preliminary scenario design has been defined and is shown in Figure 4-7, in order to explain the concept. This scenario may be later modified for the actual demonstration due to constraints or new findings.

In the proposed scenario, there are two eNodeBs and three UEs. The UEs are differentiated based on the respective service running on them. Two UEs have video streaming, i.e. Mobile Broadband service and one UE is an autonomous driving UE, i.e. a Low Latency service. There shall be at least one video streaming UE connected to each eNodeB at all times. The autonomous driving UE may be connected to either one of the two eNodeBs. The eNodeBs are also connected to SDMC and the two entities communicate commands and feedback with each other.

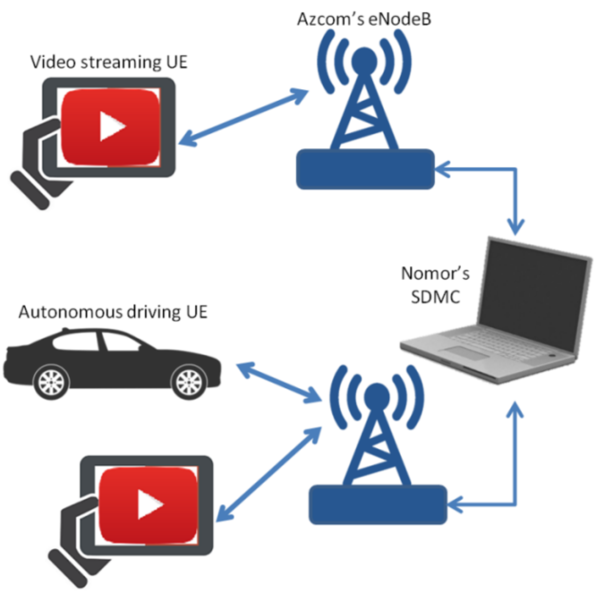


Figure 4-7. Proposed Demo Scenario Layout.

The car UE can freely move in and out of the coverage of these two small-cells. When the car UE moves into the coverage of these two small-cells, the SDMC will recognize that there is a delay-critical service and network reconfiguration should take place to cater for this change. The network will reconfigure itself as edge-cloud, as shown in Figure 4-8. As a result, the connected users observe an end-to-end latency improvement. This will have the effect that the autonomous cars will be able to drive in a more responsive manner, and hence, providing a better experience.

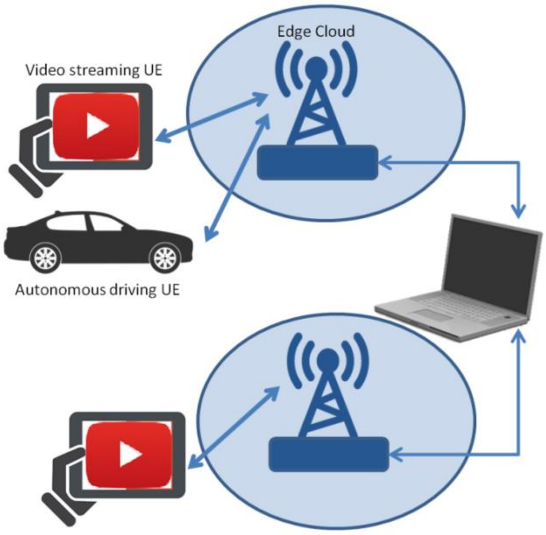


Figure 4-8. Network reconfiguration into edge-cloud.

On the other hand, when autonomous driving service is not active in the system (for example when the car user leaves the coverage area of eNodeBs), the SDMC will recognize this change and will reconfigure network functions in the central-cloud as shown in Figure 4-9. This helps in centralized coordination and hence an improvement in throughput. Consequently, the video streaming users will experience better throughput and they can receive videos faster with higher quality.

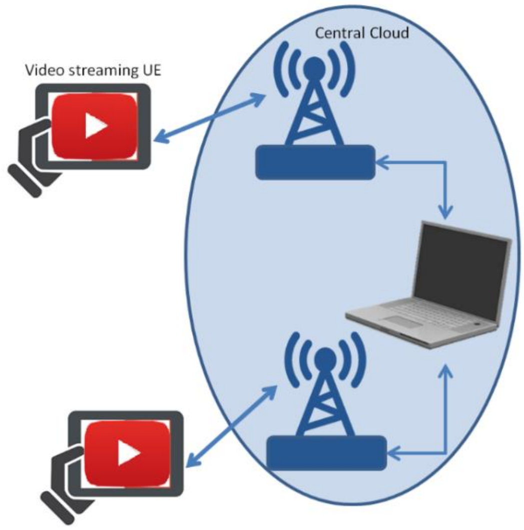


Figure 4-9. Network reconfiguration to central-cloud.

The demo proposal concept presented above consists of two eNodeBs. The main purpose of having multiple eNodeBs is to demonstrate the effect of centralized coordination among multiple eNodeBs when central cloud configuration is activated. However, due to the foreseeable technical challenges of setting up multiple real hardware eNodeBs at the available venue of different conferences and events, the actual demo setup for the Intermediate Demo might be limited to only one real eNodeB. Anyway, in case additional eNodeBs are required, they would be simulated using Nomor's network simulator software.

The SDMC component developed under this project shall be developed as proprietary software of NOMOR and will not be open-source. This is because the system-level simulator used as back-end for developing the SDMC algorithms is proprietary software. Also, the SDMC application does not make use of any open-source components. Beside that, one of the main motivations for making the software open-source would have been to get source-code contributions from the open-source community. However, in this case, the SDMC concept is so novel that no reasonable contribution from the community can be expected at this stage.

Status and Time Plan

The work on designing and implementing the demo has started. The entire design, development and testing work can be divided into smaller tasks that can help in providing a better time plan. The outline about each task and their timeline can be found in the form of Gantt chart in Annex A. Tasks D1.1.1 and D1.1.2 have been already completed as they were mainly design and concept tasks. Remaining tasks are described briefly with their current status as follows:

1. *D1.3.3 – Design a communication protocol:* In this task, a unified protocol for communicating the different entities shall be designed defining the structure of packets to be communicated mainly between SDMC and eNodeB. The protocol design shall be made extensible to support any future needs for newer message structure, for example, for additions or deletions of certain fields in the packet. This task has been completed.
2. *D1.3.4 – Implement the Communication Protocol*: The encapsulation and decoding entities for the communication protocol shall be implemented inside the SDMC controller. This task is completed.
3. *D1.3.5 – Implement a Basic Functional SDMC:* A basic version of the SDMC application shall be developed as the first stage of the demo. This SDMC version shall be capable of communicating with the eNodeBs hardware. However, it will not have the actual decision logic at this stage. The main purpose of developing a basic dummy version is to be able to set up a test bench at an early stage. This task has been completed.
4. *D1.3.6 – Preliminary Testbench Setup and Testing:* The communication between SDMC and hardware eNodeBs shall be tested with the use of the basic SDMC and the real hardware eNodeBs. Setup and testing of this first test bench will not only provide a better understanding of the functionality and key elements of the communication protocol, but it will also give a good idea about the possible real-world limitations of the hardware demo. The knowledge gained at an initial stage through this testing will be helpful for future design, implementation, and planning. This task is ongoing.
5. *D1.3.7 – Decision Logic for Network Reconfiguration:* In this task, the decision logic for the SDMC will be designed. The main task of the decision logic is to determine the optimal network configuration and settings depending on the current state of the key performance indicators for the connected eNodeBs, for example cell-load and service-type of connected UEs. The decision logic shall try to optimize the network performance by adjusting different network parameters. This task is ongoing.
6. *D1.3.8 – Simulations with Network Simulator and SDMC:* Software simulations shall be run together with Nomor’s network simulator and SDMC for estimating simulated gains based on the network reconfiguration algorithms performed by the SDMC. The simulations shall be run for scenarios closer to real world scenarios in order to estimate the actual gains of optimization and reconfiguration algorithms. Multiple simulations shall be carried out for different combinations of parameters that are of importance for the hardware demo setup. The task is ongoing.
7. *D1.3.9 – Mapping between Feedback, Throughput and Latency:* Based on the results of simulations with network simulator and SDMC, a look-up table shall be developed which contains mapping between state of different network parameters and the resulting gains in throughput and latency. The mapping will be based on the simulated results of network reconfiguration. A number of simulations shall be run for all possible sets of settings that are relevant for the network reconfiguration considering the actual

hardware setup. The results of these simulations will be gathered and analyzed for detectable patterns, and will be summarized in the form of a mapping. The task shall begin in M16.

8. *D1.3.10 – Running tests on the test-bench:* Once the actual SDMC has been developed with all mappings, a test bench shall be set up for analysing the performance improvements. This task shall begin in M17.
9. *D1.3.11 – Improving Communication, Decision Logic and Mapping:* Based on the results and findings of the testing, some improvements and modifications might be needed in the communication protocol. In addition to that, in case the results in real-world are not as expected, some improvements should be made to the decision logic and regeneration of mapping might also be needed. The task is expected to start in M17.
10. *D1.3.12 – Preparing pre-release Demo for Internal Review:* Preparing a demo by defining the storyline, i.e., the demo steps and phases. Each step shall be outlined in detail and should demonstrate different steps in the demo. This task will be started in M17.
11. *D1.3.13 – SDMC GUI adaptation to show eNodeBs activity*: This is an optional feature where the main purpose is the demonstration of the effect of network reconfiguration to the audience in a simple manner. This task will be started in M17.
12. *D1.3.14 – Finalizing the Intermediate Demo:* This task will begin in M19.
13. *D1.3.15 – Improvement of SDMC based on demo feedback:* The intermediate demo will gather some feedback from the 5G NORMA partners and also from the audience at different conferences and 5G events where it is presented. Based on this feedback, the SDMC application shall be improved and possibly new features added. This task shall begin in M21.
14. *D1.3.16 – Research work on SDMC in 5G Networks*: After the first prototype development of SDMC, it will be used for research on 5G networks and its role in integration with other 5G network elements shall be analysed. SDMC will be continuously improved and enhanced based on the results and findings of this research work. The results of this research work could then be presented in the form of research papers and at important 5G related events. This task shall begin in M23.

4.2 Demo 2. Service-aware QoE/QoS Control

Network Slicing

The UC3M Demo 2 will demonstrate mainly one of the concepts considered in 5G NORMA, i.e., Network Slicing. Nevertheless, other functionalities such as service-aware orchestration and enhanced mobility management are also in the scope of this demo.

Showcase

Two network slices are deployed to the same eNB: a mobile broadband network slice and the low latency network slice. Both share the same spectrum. The network architecture is as follows (just u-plane is depicted):

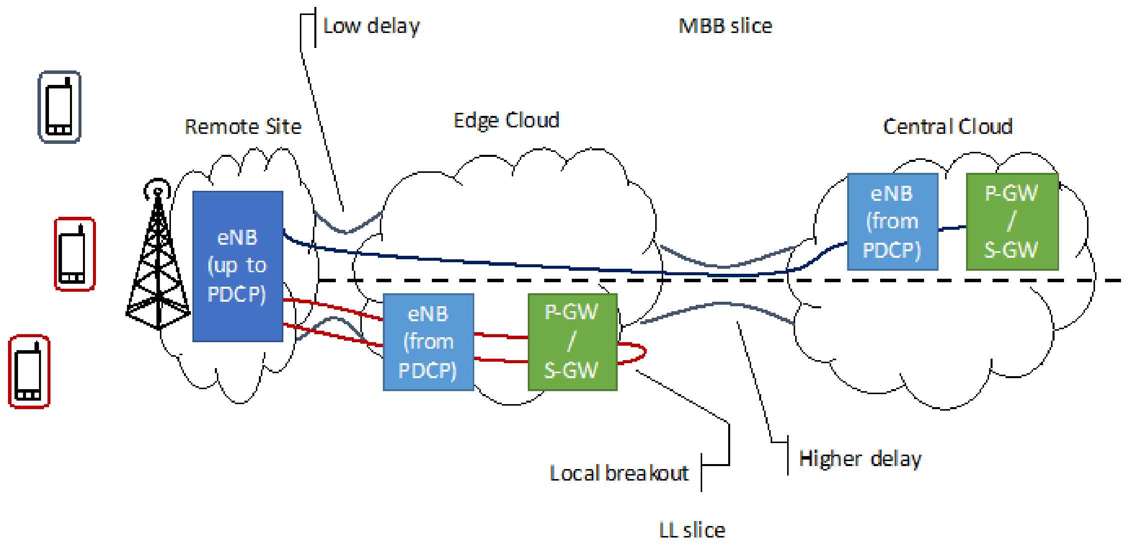


Figure 4-10. The UC3M Demo plot.

The demo setup is composed by three sites as depicted in Figure 4-10: The remote site (that contains the antenna site and the lower layer of the 3GPP RAN) is connected through a low delay high bandwidth link to the edge cloud, where the higher RAN stack and the Gateways for the LL slice are located. The central cloud is connected through a higher delay link. Here the higher RAN stack and the Gateways for the MBB slice are located.

UEs interact among them using latency constrained network services through the core network (Device-To-Device is out of the scope of this demonstrator), i.e., when they are connected using the LL slice they experience the right KPIs, otherwise their QoE is broken.

On the other hand, the users of the MBB slice can use it for non-latency constrained applications, such as video on demand.

Architecture

The main innovation demonstrated by this demo is network slicing up to the RAN functions. Lower RAN functions (up to PDCP) are considered as common network functions for the two slices. Higher RAN and EPC are deployed as dedicated NFs, with different instances per network slice. This is aligned with the 5G NORMA architecture.

Other functionalities demonstrated are service aware orchestrator and enhanced mobility. Orchestration of dedicated network functions provides different outcomes according to the targeted network slices. For the LL slice, the data-plane is located in the edge cloud, while MBB gets the data-plane located in the central cloud. Although not achieved through an on-line algorithm, the outcome is still a valuable result for the 5G NORMA innovations. Enhanced mobility mechanisms are applied to the LL slice such that the local breakout routing optimization avoids unnecessary usage of the Packet Data Network (PDN) [24].

Hardware / Software configuration

The demo HW/SW platform is based on the Universal Software Radio Peripheral Software Defined Radio (USRP SDR) and the OpenAirInterface software implementation of the RAN and EPC stack.

OpenAirInterface

It is a software platform developed by the non-profit consortium OSA (OpenAirInterface^{TM} Software Alliance) and promoted by EURECOM, a French education institution who created the environment in 2009. Its main goal is the development of an open source eco-system for the core network and the Evolved UTRAN (E-UTRAN) network access defined by 3GPP. They provide tools for both software and hardware, which could be used for technology researching and experimentation of future networks such as 5G, C-RAN (cloud-RAN) or massive Multiple Input Multiple Output (MIMO).

The OpenAirInterface (OAI) platform provides a software-based implementation of the LTE system, including the entire protocol stack of the 3GPP standard for E-UTRAN and EPC. Through this platform, it is possible to emulate a LTE base station (OAI eNB) and a backbone network (OAI CN), all on a computer or distributed over multiple. In addition, it allows connecting commercial mobile devices in order to test distinct configurations and network settings, while monitoring them in real time.

The software is written in C for low-latency kernel Linux operating system and optimized for Intel x86 and ARM processors. At present, the core network code is being updated and it is recommended for non-low-latency kernels.

USRP

Universal Software Radio Peripheral. This is a product family designed and sold by Ettus Research [21] since 2010. Their SDR devices such as GNU Radio are intended to be a low-cost hardware platform that performs most of the signal processing and includes an RF front end. Moreover, it is mainly set aside for universities, laboratories and even amateur radio operators.

Among their products, we selected the USRP Ettus B210 model released in 2013, which could be used as a prototype of the Global System for Mobile communications (GSM) base stations with OpenBTS software. It is characterized by including an Spartan6 FPGA, an AD9361 RFIC direct-conversion transceiver and an Universal Serial Bus (USB) 3.0 interface with convenient bus power in order to connect the host PC for data transference.

FPGA is a reprogrammable silicon chip developed by National Instrument, Ettus’ parent company. Unlike processors in PCs, programming an FPGA consists of re-wiring the chip, i.e., reconfiguring the interconnection of logical blocks, in order to implement its functionality rather than run a software application.

Direct transceivers, also known as zero-IF or synchrodyne, are single-chips capable of streaming up to a given RF carrier. Unlike superheterodyne architecture, it does not use an intermediate frequency stage, so that the detection of the input signal is performed by a local oscillator which operates close to the carrier frequency.

Regarding the family cards, we could have also considered Network Series family which have Gigabit Ethernet connection. Nevertheless, Bus Series provides an USB interface that is sufficient for the bandwidth under study. Within Bus Series cards, there are several models such as B200 or B100 which allow for operating in 1 TX 1 RX Single Input Single Output (SISO), but not supported by OAI and B210 is Ettus's first board that will be fully capable of 2x2 MIMO (Multiple Input Multiple Output) by itself. Moreover, it is possible to use an USRP X310 model since it is supported by OAI, but is much more expensive in comparison with B210.

However, USRP cards have low-cost competitors: bladeRF [22] and HackRF [23] devices (see further details in Table 4-3 below). Since OAI platform is not exclusive for USRP devices, it provides software for both bladeRF and USRP devices.

	HackRF	bladeRF		USRP		
		x40	x115	B100 Starter	B200	B210
Radio Spectrum	30 MHz – 6 GHz	300 MHz – 3.8 GHz		50 MHz – 2.2 GHz	50MHz – 6 GHz	
Bandwidth	20 MHz	28 MHz		16 MHz	61.44 MHz	
Duplex	Half	Full		Full	Full	2x2 MIMO
Sample Size (ADC/DAC)	8 bit	12 bit		12 bit / 14 bit	12 bit	
Sample Rate (ADC/DAC)	20 Msps	40 Msps		64 Msps / 128 Msps	61.44 Msps	
Interface (Speed)	USB 2 HS (480 megabit)	USB 3 (5 gigabit)		USB 2 HS (480 megabit)	USB 3 (5 gigabit)	
FPGA Logic Elements		40k	115k	25k	75k	150k
Microcontroller	LPC43XX	Cypress FX3		Cypress FX2	Cypress FX3	
Open Source	Everything	HDL + Code Schematics		HDL + Code Schematics	Host Code	
Availability	2014	2013		2013	2013	
Cost	$300	$420	$650	$675	$675	$1100

Table 4-3. Programmable hardware comparison.

Therefore, taking into account these models and the compatibility restrictions of OAI[7], we have chosen USRP model B210 because it covers a wide bandwidth range that enables full duplex communications and MIMO scenarios, combined with the usage of USB 3.0. In addition, its open source characteristics and FPGA number of logic elements were interesting for further studies.

Objectives and KPIs

The main objectives of the demo proposed are: i) demonstrate the feasibility of end to end network slicing up to the RAN and ii) demonstrate the flexible connectivity properties achieved by introducing selected 5G NORMA innovations into the network architecture. Hence the evaluated KPIs for this demo may be classified into two categories of hard and soft KPIs.

Hard KPIs

In the proposed demonstrator, two different network slices coexist on the same infrastructure. The required KPIs are heterogeneous, namely:

- The low latency network slice does not impose stringent requirements on the available bandwidth, but it does for the latency. The target application has not been detailed yet, but it will require a low latency network service with an end to end delay bounded within tens of milliseconds.
- The Mobile Broadband Network slice, conversely aims to achieve the highest amount of bandwidth available while having the least resource occupation footprint on the operator network.

Although the real KPIs will be targeted to the used application, it is clear that their extent is certainly heterogeneous. Especially, it is expected that the low latency application cannot work properly while using the mobile broadband slice.

Soft KPIs

The soft KPIs associated to this demonstrator are mostly related to the feasibility of some of the concepts proposed in the framework of 5G NORMA WP5.

- Flexible and service aware orchestration: the two envisioned network slices need very heterogeneous KPIs, especially in terms of maximum allowed end to end latency. This calls for an especially tailored orchestration based on the service. The Low Latency network slice will need a different orchestration, unbalanced to the edge cloud, compared to the one used by the Mobile Broadband that aims for achieving high multiplexing gains in terms of resource usage at the cost of a higher delay given by the transit to the central cloud.
- End to end RAN slicing. The 5G NORMA architecture is designed to include the Network Slicing concept into the architecture. This is a significant change compared to the current, rather monolithic, LTE architecture. Although far from being complete, this demonstrator will prove the feasibility of network slicing within the 5G NORMA architecture and some initial lessons learned by its real-world implementation. This includes, for example, the algorithm needed for the network slice selection.
- Network Function Sharing. As envisioned by the Software-Defined Mobile network Coordinator (SDMX) concept from WP3, some of the network functions, especially the ones in the RAN, are finally going to be shared. The feedback obtained by the development of this demo will be used to feed the design of SDMX related algorithms.

Other objectives

The UC3M demonstrator will rely on Open Source software, namely OpenAirInterface. During the development of this demonstrator, several changes to the mainline code need to be done in order to support the innovative features of the demonstrator. We are currently active in the OpenAirInterface community in terms of bug reporting and testing, but we will evaluate during the development phase if the modifications for this demo may be included in the OpenAirInterface repository and kick off the necessary actions.

Status and Time Plan

In the framework of WP6 tasks, we organized the work into nine subtasks that will be fulfilled by the end of the projects. They are detailed next, while the detailed time plan is described in Annex A.

UCM3 Specific Tasks (Network Slicing)	
Define innovations provided by the UC3M demo	D2.2.1
Get (purchase) all necessary HW and SW platforms	D2.2.2
Enable Remote Site (antenna site and the lower layer of the 3GPP RAN)	D2.2.3
Study the feasibility of different functional splits	D2.2.4
Implement the selected functional split	D2.2.5
Study sharing techniques for NF	D2.2.6
Implement the selected NF sharing technique	D2.2.7
Provide UE terminals for the demo	D2.2.8
Implementation of proposed solution in a small scale testbed & evaluation in conjunction with WP6	D2.2.9

Table 4-4. Subtasks of Demo2, Network slicing part only

D2.2.1

The creation of the demo plot was the most extensive task performed so far. We gathered feedback from WP3, WP4 and especially WP5. The outcome of this decision is the demo plot described previously in this section, and the consequent definition of the other subtasks detailed next.

D2.2.2

The choice of the hardware and the software platform has been of paramount importance for a satisfying outcome of this demo. We thoroughly evaluated several HW/SW platforms. We finally decided for the OpenAirInterface and USRP HW/SW platform for the following reasons.

- OpenAirInterface is an Open Source project: it has a large number of users and a very active community of users
- The most used combination of HW and SW is the one with the USRP platform. For this reason, we decided for this HW platform given that it fulfils the requirements in terms of bandwidth.

D2.2.3

We are currently working on setting up the hardware and software, being ready for the next subtask. The OAI platform enables different configurations involving different commercial elements:

- UE commercial:
  - eNB commercial + OAI EPC
  - OAI eNB + EPC commercial
  - OAI eNB + OAI EPC
- OAI UE:
  - OAI eNB + OAI EPC
  - OAI eNB + EPC commercial
  - eNB commercial + OAI EPC
  - eNB commercial + EPC commercial

Among these possibilities, we have chosen OAI UE with OAI eNB + OAI EPC. This scenario is recommended to use two different machines: one for the eNB (where the UE is already connected) and therefore, it constitutes the Radio Access Network and another one for the EPC (which includes the Home Subscriber Server –HSS–, MME, S-GW and P-GW, i.e. the core). Since the kernel of the Operating System is different for these two elements, we created two different virtual machines with the recommended kernel:

- eNB: Ubuntu 14.04 kernel 3.19.0-28 low latency
- EPC: Ubuntu 14.04 kernel 4.7.x

Once we have set-up the machines, it is required to configure the database from where the HSS checks if the user is allowed to communicate, the IP addresses for the control plane and data plane interfaces, the Domain Names Server (DNS) address communicated to UE’s, and other NAS functionality.

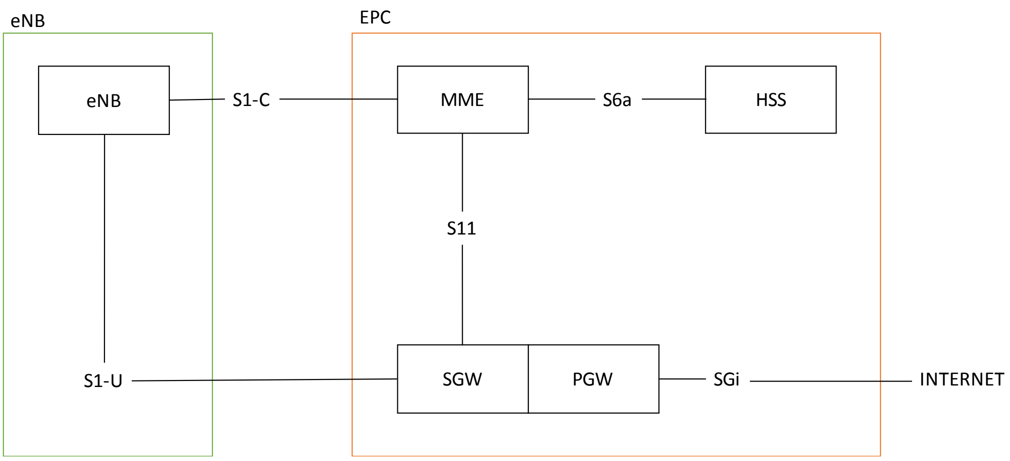


Figure 4-11 The network slicing demo current SW architecture.

Building on the scheme depicted in Figure 4-11, we will develop the next subtasks (colour boxes represent different machines):

D2.2.4 and D2.2.5

After having built a stable testbed, we will seek for the best RAN functional split we can achieve. We target the PDCP – RLC split, but if for reasons of technical limitation of the OpenAirInterface software, this would not be possible, we plan to demonstrate the innovations with other functional splits. Once the best functional split is selected, we will proceed to implement it in the OpenAirInterface stack.

D2.2.6 and D2.2.7

Network function sharing techniques have to be implemented in the OpenAirInterface stack in order to demonstrate the innovations we proposed within 5G NORMA described above. How to achieve optimal resource sharing at the RAN level and resource separation at the CN level are topics currently being investigated by WP4 and WP5. We plan to study the introduction of the proposed techniques in the framework of this demonstrator, beside the one already envisioned by the demo layout described above.

D2.2.8

In order to demonstrate the feasibility of Low Latency services (or the unfeasibility of Mobile Broadband Network slices for Low Latency services) we will run a low latency application. We will evaluate several alternatives including video-gaming, augmented reality, or scientific computing. The choice of the UE will also be considered, being a mobile terminal or a 4G LTE modem the candidates for this role.

D2.2.9

Once all the previous tasks were completed, we will test and debug the demonstrator. We plan to exhibit a reduced set of the demo functionalities at an earlier event, while the fully fledged demonstrator will be available by the final audit. Possible integration will be investigated in this timeframe, although preliminary work will also be done earlier.

NFV MANO Implementation

Showcase

If network functions described in the previous section 4.2.1 were typical bare-metal based network functions (i.e., Physical Network Functions), it would be very difficult to move them between the different slices. Probably, required time and resources necessary to deactivate the network function on the first slice to re-instantiate it on the second one would be too complicated to be smoothly implemented in practice. However, by implementing these functions as Virtual Network Functions the operation of moving them among different slices becomes much more accessible.

Network Functions Virtualization offers a new way to design, deploy and manage networking services by decoupling the network functions from specific hardware appliances [1]. That way, network functions become elementary software building block, which can be managed and orchestrated in a very agile way as software components.

Once the network functions are available as VNFs, there could be, of course, different ways to manage and operate them. For the scope of this demo, we are going to use the ETSI NFV MANO reference architecture [2]. This is the ETSI-defined framework for the management and orchestration of all resources in a cloud data centre, including computing, networking, storage, and Virtual Machine (VM) resources. It addresses various aspects of management and orchestration, which are specific to NFV such as lifecycle management, operations management (e.g., provisioning, scaling), information elements, architecture and the interaction with legacy operational and management systems (e.g., Operations and Business Support Systems -OSS/BSS-), but above all, it offers the decoupling of network functions software from the hardware. As it is well known, the ETSI is a key in developing standards for information and communications technologies in Europe; in this case, a specific group in charge of developing the requirements and architecture for network functions virtualization within telecoms networks has been created, i.e., the ETSI Industry Specification Group for Network Functions Virtualization (ETSI ISG NFV). The ETSI NFV MANO specification comes from this group. We think that this specification, beside providing an adequate framework for the demo itself, is also well aligned with the specific 5G NORMA architecture as it is defined in WP3 and WP5.

The demo showcase itself will, based on this ETSI NFV MANO architecture, deploy and orchestrate the network functions defined in the previous sections as virtual network functions, i.e., the eNB top layers stack on one hand and the routing network functions on the other hand (P-GW and S-GW components). These functions will be designed and developed to fulfil the ETSI NFV MANO specification. Also, they will be deployed and orchestrated on a specific implementation of this ETSI NFV MANO architecture provided by ATOS. Additionally, both network functions will be orchestrated and placed in the different network slices (LL or MBB slices) depending on QoS/QoE measurements.

Architecture

Figure 4-12 represents the ETSI NFV MANO architecture, which is the same architecture that we are going to deploy for this demo:

As we can see the NFV MANO block comprises the blue tone blocks on the right. The other four blocks on the left (OSS/BSS, EM, VNF and NFVI) are outside the scope of the NFV MANO itself, although high level interfaces (or “Reference Points” as they are defined in the ETSI specification) are defined.

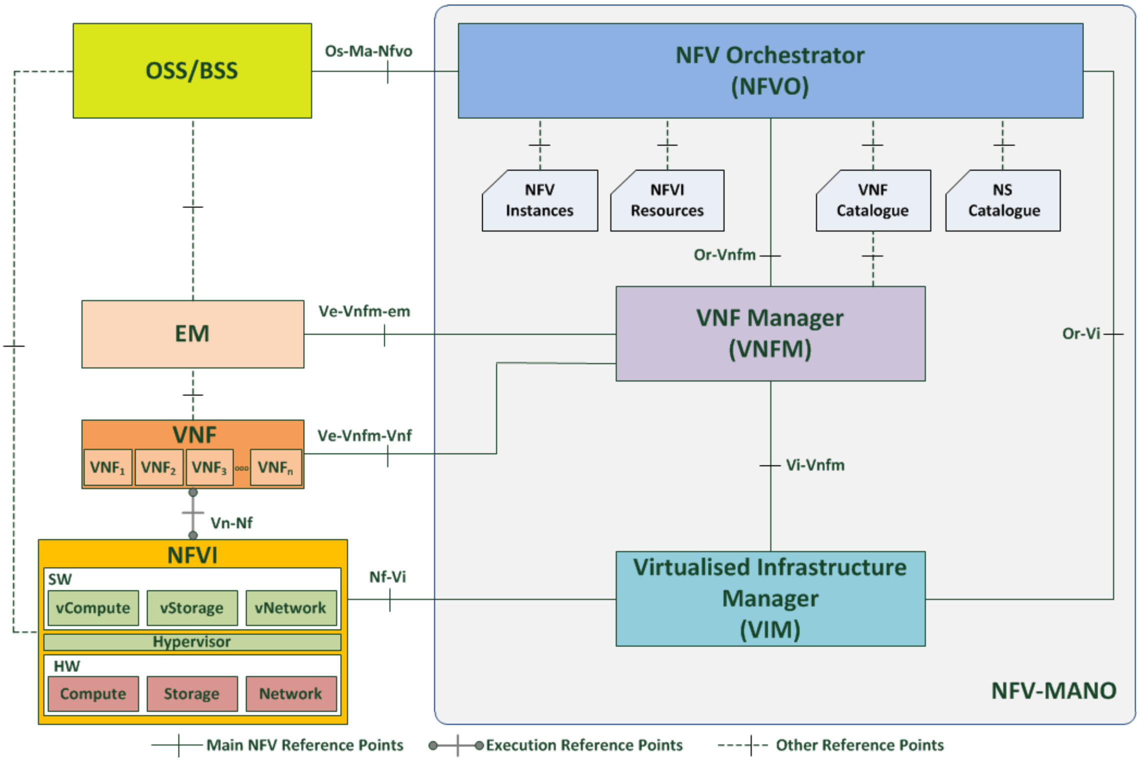


Figure 4-12. ETSI NFV MANO Architecture.

This is a brief description of the functional blocks in the figure:

- Network Functions Virtualization Infrastructure (NFVI). This block represents all the hardware components (e.g. compute, storage, and networking) and software components (e.g. hypervisors and virtualized compute, storage and networking resources) that together provide the infrastructure resources where VNFs are deployed. An important component inside this block is the Network Controller (e.g.: Software Defined Networking –SDN– controller if they exist in the deployment).
- Virtual Network Function (VNF). This block represents the set of Virtualized Network Functions. For the demo purposes, these are the network functions described in the previous section, i.e., the radio stack and the routing components functions. These functions will be executed on the virtualized NFVI resources.
- Element Management (EM). It is responsible for monitoring the fault states, configuration, accounting, performance and security (FCAPS) for a VNF. It collaborates with the VNF Manager to perform those functions that require exchange of information regarding the NFVI Resources associated with the VNF. Its main functions are:
  - Configuration of network functions provided by the VNF.
  - Fault management of network functions provided by the VNF.
  - Accounting for the usage of VNF functions.
  - Collect performance measurement results for the functions provided by the VNF.
  - Security management for the VNF functions.
- Virtual Infrastructure Manager (VIM). This is the module responsible for managing the virtualized infrastructure. Main functions are:
  - Control and manage the NFVI compute, storage and network resources.
  - Collect performance measurements and relevant events from the VI resources.
  - Keep an inventory of the allocation of virtual resources to physical resources.
  - Organize virtual links, networks, subnets, and ports.

  - Manage a repository of NFVI hardware resources (compute, storage and networking) and software resources (hypervisors), and discover the capabilities and features to optimize the use of such resources.
- VNF Manager (VNFM). The VNFM is responsible for the lifecycle management of VNFs under the control of the Network Functions Virtualisation Orchestrator (NFVO), which it performed by instructing the VIM. VNFM operations include:
  - Instantiation & termination of VNFs (lifecycle management).
  - VNFs scaling.
  - Updating or upgrading VNFs.
  - Overall coordination and adaptation role for configuration and event reporting between NFVI and EM.
- Network Functions Virtualisation Orchestrator (NFVO). Performs top level resources orchestration and network service orchestration, as well as other functions. It connects different functions to create an end-to-end, resource-coordinated service. The orchestrator provides a high level logical abstraction view of the Network Services (NS) deployed on the infrastructure (e.g., from here, it is possible to define complex services specifying different forwarding graphs among the existing network functions). As you can see, the orchestrator can access different catalogues (databases) in the NFV MANO scope. The NFVO main functions are:
  - On-boarding new Network Service, VNF Forwarding Graphs (FG) and VNF Packages.
  - NS lifecycle management, including instantiation, scaling, performance measurements, event correlation and termination.
  - Policy management for NS instances.
  - Global resources management, validation and authorization of NFVI resource requests.
- Operations and Business Support Systems OSS/BSS. These are the combination of the operator's other operations and business support functions, which are not captured in the ETSI MANO framework but are expected to have information exchange with the ETSI MANO functional blocks. OSS/BSS functions may provide management and orchestration of legacy systems and may have full E2E visibility of services provided by legacy network functions in an operator's network.

Although all the main blocks and their interfaces are well defined, the ETSI NFV MANO specification does not mandate any specific realization of the NFV MANO architectural framework. So, for the practical realization of this demo, we have selected a set of hardware and software resources to enable the implementation. In the following section, we describe those HW and SW elements.

Hardware / Software configuration

Hardware

The demo HW platform is based on:

a) An HP ProLiant DL380e Generation 8 Server[7]. This is equipped with 12 Large Form Factor (LFF) drive bays, 2 Intel Hexa-Core Xeon E5-2420 processors (6 Cores), 96 GB of Random Access Memory (RAM) and 4x1TB Hard Disk drives (see Figure 4-13 below).

[7] http://www.hgcode.com.br/downloadDoc.php?d=arqConteudo/arqProdutos&f=DL380e_G8_Datasheet.pdf

Figure 4-13. HP ProLiant DL380e Gen8 Server.

This server will be used as testbed environment and, eventually, for demo purposes also (enabling remote access). We consider this is sufficient for the Demo 2 needs, but it is in line with the standard market offering for dynamic compute and storage requirements.

b) A general purpose laptop to simulate the Edge Cloud component (see Figure 4-10 above). During the development phase, this component will be emulated as a Virtual Machine, but for the demo it will be a bare-metal component.

Software

The software used for this demo can be split into two groups:

- Already available software projects.
- Specific software for this demo.

Our approach is, whenever possible, to get benefit from some already available general purpose software projects, especially for the ETSI-MANO implementation. In addition, it will be necessary to develop some software specifically for this demo.

In the following sections we describe these two groups. Additionally, we also describe the testbed platform that we are currently using to integrate and test this software.

Already available software projects.

The process of selecting the different already available software projects has been quite long and complex. The main issue has been to select among the different possibilities to meet the specific requirements for this demo. Currently, there are many options implementing the different ETSI NFV MANO blocks in slightly different ways. We've focused on several of them for this demo, but none of them fully meets the specific requirements for this demo. The main drawback is related to the physical placement of NFs. Most of the currently available projects are primary focused on the IT market, but not on Telco applications. For the typical IT cloud applications, the NF placement is not an issue; what it is normally expected is a high degree of decoupling between NFs and their physical placement. Conversely, for telco applications -and specifically for this demo- it is a strict requirement to be able to determine the precise host where a NF would be deployed. We need to distinguish between different slices (representing edge and core networks) and we must be able to decide whether a NF will be executed on a specific host depending on the QoS/QoE. Ffor low latency scenarios, the NF should be executed in a host at the edge while for regular MBB scenarios a placement in the CN if sufficient. The challenge is to get this not only for the lower layers in the VNF MANO architecture (NFVI or VIM), but also the orchestrator layer (NFVO).

Thus, one of the biggest determining factors when choosing the implementations has been the possibility of making changes to bridge this gap. Considering this, the approach was obviously focused on Open Source projects, since we will need to improve and change the software to meet the specific requirements for this demo, and specifically, those Open Source projects where the possibility to incorporate changes is more feasible for us. References from [3] to [14] and [16] are the main Open Source projects we've taken into account. In the following paragraphs, we describe the selection for each functional block in the ETSI MANO architecture[8].

Virtual Infrastructure Manager

For the VIM implementation, the selected project is OpenStack [4]. This is a free and Open Source platform for cloud computing that could be used to control large pools of compute, storage, and networking resources throughout a datacentre. The project was started by the American National Aeronautics and Space Administration (NASA) and Rackspace, but currently is managed by the OpenStack Foundation, a non-profit corporate entity which supervises the development, distribution and overall implementation.

Our selection has been mainly based in the following facts:

- OpenStack is a kind of de-facto standard; it is a well-known project that generally is the dominant choice for management and orchestration functions.
- It is Open Source (Apache 2.0 license) with a wide and active community working on it. It continues growing with periodic releases and updates with an active support from both individuals and corporations[9].
- We think OpenStack may be an option, not only for the demo purposes, but also for a possible implementation of the 5G Norma architecture in order to provide a real-world proof of concept[10].
- It can perform the VNF placement on specific hosts. As described before, this is a basic requirement for the demo.
- Integration with other orchestration modules, e.g. TeNOR, OpenMano, and OpenBaton.
- Atos has previous experience with this project, which can be exploited for the project.

Some of the most relevant OpenStack features are:

- OpenStack is agile and easy to deploy and scale.
- It provides standardized interfaces between the different NFV elements and the infrastructure.
- It offers well documented APIs, UIs, shared services, operations, automation options for VNFs, and other functions integration.
- High degree of compatibility and portability[11].
- Regarding security, it supports multiple forms of secure authentication, authorization and accounting (AAA).
- Regarding storage, OpenStack offers a variety of storage pools and supports block-IO from different vendors, as well as object file storage. Its built-in storage management automatically recovers failed drives or nodes. Replication and erasure coding also provides strong data integrity.
- Due to the wide Open Source community behind there is a big offer with popular open source and commercial network plug-ins and drivers.
- It offers a high degree of network/elements deployment automation.
- It is mainly written in Python (a widely adopted, interactive and object-oriented programming language).

[8] The OSS/BSS and the EM nodes are not included in this list because, for this demo, we obviously don't have operations and business support functions nor element management functions outside the ETSI MANO framework.

[9] Some relevant contributing companies are: RedHat, Dell, HP, IBM, Cisco, Canonical and SuSE.

[10] There are already many telecom and enterprise implementations using OpenStack: AT&T, China Mobile, SK Telecom, Ericsson, Deutsche Telekom, Comcast, Bloomberg, and more.

[11] For example OpenStack APIs are compatible with AWS (Amazon Web Services) so users don't need to rewrite applications for AWS.

As we've described, the VIM is the MANO function controlling the assignment of virtualized compute, storage and network resources from the NFVI to support the VNFs. The primary OpenStack modules that will be involved to get this for the demo are:

- The OpenStack compute module called Nova, which can be used to manage virtual or bare metal servers.
- The OpenStack virtual storage block called Cinder, which supports practically any storage back end.
- The OpenStack networking module called Neutron.

For the demo, we will use also the OpenStack web-based dashboard module called Horizon.

<u>VNF Manager & Orchestrator</u>

VNFM and NFVO modules are sometimes delivered as a single component, which is usually referred to as "Orchestrator with embedded VNF Manager". In this case, the TeNOR implementation [7] has been primarily selected for implementing these two blocks.

TeNOR is the NFV Orchestrator platform developed by the T-NOVA project [15]. T-NOVA is a European project in the FP7 framework programme that releases some of its work as Open Source. Among other modules, this includes the TeNOR component.

TeNOR is responsible for managing the entire NFV lifecycle service. It is a T-NOVA contribution from Altice Labs (former "PT Inovação e Sistemas") and "I2CAT Foundation", which is a non-profit research and innovation centre with a wide experience in different national and European Research, Development and Innovation (R+D+i) projects, leading research lines in new fixed & mobile network architectures.

We've selected this module mainly for the following reasons: It is an Open Source project[12], it integrates with OpenStack, Atos has previous experience using this component, and although we know that the current release doesn't meet all the requirements that we need for the demo (especially those regarding the VNFs placement), we believe it is feasible for us to modify the platform in order to add the missing functionalities. The improvements will be part of the innovations for this project, and will be released to the Open Source community. The following is a simplified diagram representing the high-level TeNOR architecture:

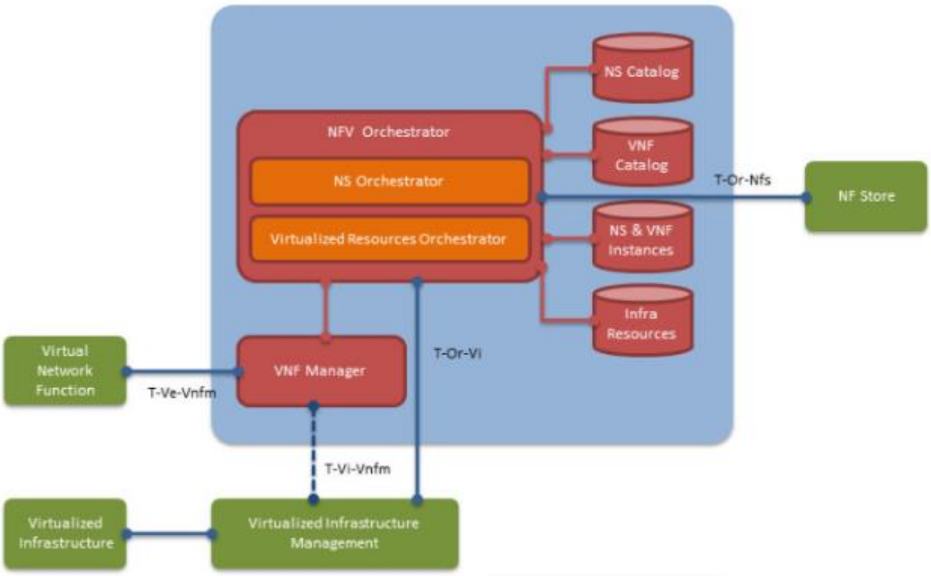


Figure 4-14. TeNOR Architecture.

[12]The software is distributed using different licenses; although Apache 2.0 is the most common the Boost license is also used for some libraries. The source code is released under GitHub (https://github.com/T-NOVA/TeNOR).

As we see, TeNOR is split in two main core submodules:

- Network Service Orchestrator (NSO), and
- Virtualized Resources Orchestrator (RO).

We can also see the embedded VNFM and the metadata & instance repositories and the main ETSI NFV MANO reference points (the "T"-prefix represents the specific TeNOR implementation).

The RO implements the Or-Vi reference point, i.e., the direct communication with the VIM. It is basically responsible for creating network and computes resources, and interconnecting VNFs.

On the other hand, the NSO responsibility is the management of the NS lifecycle and its procedures, including:

- NSs and VNFs on-boarding, i.e., management of Network Services deployment templates (NS Descriptors) and VNF Packages, as well as of the NSs instances topology, e.g., create, update, query and delete VNF Forwarding Graphs.
- NS instantiation, i.e., trigger instantiation of NS and VNF instances. Also, management of the instantiation of VNFs, in coordination with the embedded VNFMs, as well as validating of NFVI resource requests from VNFMs.
- NS update, i.e., support of NS configuration changes such as changing inter-VNF connectivity or the constituent VNFs.
- NS supervision, i.e., monitoring and measurement of the NS performance and correlation of the acquired metrics for each service instance. Data can be obtained from the VIM (performance metrics related with the virtual network links interconnecting the network functions) and from the VNFM (aggregated performance metrics related with the VNF).
- NS scaling, i.e., increases or decreases the capacity of a NS according to per-instance and per-service auto-scaling policies. The NS scaling can imply either increasing or decreasing of a specific VNF capacity, create or terminate new or old VNF instances, and increase or decrease the number of connectivity links between the network functions.
- NS termination, i.e., release of a specific NS instance by removing the associated VNFs and associated connectivity links, as well as the virtualized infrastructure resources.

NFVI

Beside the hardware, for the NFVI we've selected also the following software components:

- The Citrix XEN hypervisor [16]. This is an Open Source bare-metal hypervisor (type-1) from the Linux Foundation Project. It runs directly on the hardware and is responsible for handling processors, memory, and interrupts. On top of the hypervisor a number of virtual machines can be executed. It is widely used as basis for a number of commercial and open source applications.
- For the networking resources we have:
  - Open vSwitch [9]. This is a virtual switch designed to enable programmatic network automation. It supports standard management interfaces and protocols, e.g. NetFlow, sFlow and 802.1ag. This is also Open Source software released under the Apache 2.0 license.
  - OpenDayLight [8]. This is the Software Defined Networking platform. It will be used to dynamically control the Open vSwitch node in order to interconnect the other different nodes in the infrastructure. It is also an Open Source project hosted by the Linux Foundation.

Specific software for this demo.

The software specifically developed for this demo falls into two main categories:

- The software modifications previously mentioned, i.e., the modifications introduced in the Open Source projects. This software will be released to the Open Source community.

- The VNFs described in Section 4.2.1, i.e., the radio stack and the routing components. This software (provided by UC3M) is based in OpenAirInterface, which will be considered for a release to the Open Source community.

Additionally, it will be necessary to also deploy some convenience software for testing purposes. This software won't be shown in the final demo but it will be necessary for testing the management and orchestration processes, e.g., certain "dummy" network functions to test the VNF placement used until the final VNFs are available.

Testbed Platform.

The following figure represents the current testbed platform for the demo:

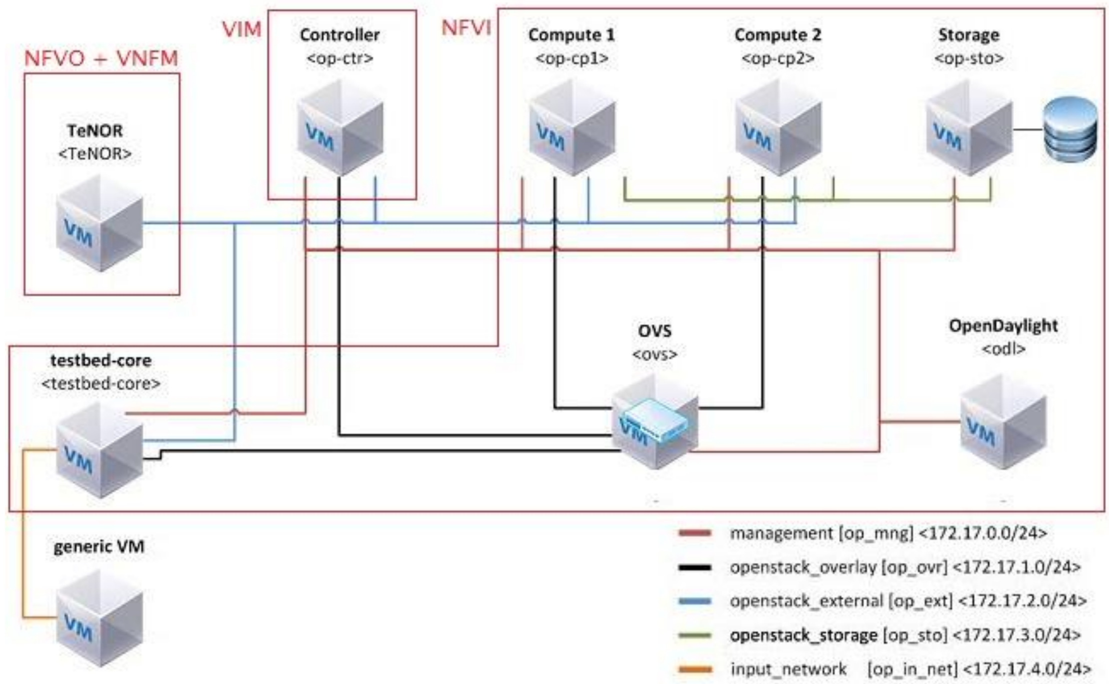


Figure 4-15. NFV Testbed Platform.

As shown, we have a specific VM for TeNOR (the orchestrator with the embedded VNF Manager). We also have a specific VM for the VIM module (which is running OpenStack). In the NFVI block we have:

- Computing resources: a couple of virtual compute nodes (op-cp1 & op-cp2). These are the Virtual Machines where the different VNFs will be executed. One of these nodes represents the edge network, while the other one represents the core network.
- Storage resources: a storage node (op-sto) providing the necessary storage resources for the computing nodes.
- Networking resources: In this case we have three different nodes:
  - OVS node: This is a VM running Open vSwitch (this is an Open Source component designed to enable programmatic network automation) [9].
  - ODL node: This is the SDN controller based on OpenDayLight [8]. It will instruct the Open vSwitch node to interconnect the other different nodes.
  - Testbed-Core node: This is just a gateway node to provide access to other nodes outside the NFVI (e.g., the external 'generic VM' node in the figure).

The NFVI block also comprises the hardware resources and the XEN bare-metal hypervisor, which are not explicitly represented in the figure.

Objectives and KPIs

The main objectives of this demo from the virtual resources orchestration viewpoint are:

1. Regarding the ETSI NFV MANO architecture:
   a. Demonstrate the feasibility, i.e., verify that it can be implemented as a proof of concept. In order to do this, we have to implement and validate a specific ETSI NFV MANO implementation able to manage and orchestrate the VNFs provided by UC3M, i.e., the radio stack and routing components VNFs.
   b. Demonstrate the applicability, i.e., verify that the ETSI NFV MANO implementation can be actually used for the specific purposes of this demo. In this sense, the MANO functionality should include the possibility of moving the VNFs between different specific nodes representing a hypothetic operator's network.
   c. Evaluate different MANO platforms. Although the demo will be performed on a specific ETSI MANO implementation, we consider it also necessary to evaluate and test different software platforms.
   d. Prove efficiency, i.e., the implementation can be achieved using reasonable amount of time and resources.
   e. Explore limitations in order to identify possible drawbacks in the architecture.

2. Regarding the information exchange with other WPs:
   a. Validate the technical requirements and KPIs coming from WP2.
   b. Evaluate the 5G NORMA architectural principles described in WP3. Although not all the functional building blocks described in WP3 will be implemented for this demo, some of the main components (those already present in the ETSI NFV MANO framework) can be demonstrated.
   c. Evaluate adaptive allocation of functions to different network nodes (WP4).
   d. Evaluate QoS/QoE mapping and monitoring control processes, orchestration functions and VNF life-cycle management as described in WP5.
   e. Provide feedback to the other WPs from the results obtained.

3. Demonstrate one of the project's key principles, i.e., the possibility of dynamic relocation of network functions between the edge of the network and a centralized cloud infrastructure, thereby enabling low latency communication.

4. Dissemination.
   a. The demo should serve as platform to be exhibited in public conferences and events.
   b. Since for this demo we are extensively using third party Open Source projects, another important objective for this demo is to release the software components specifically developed for this demo to the Open Source community.

Regarding KPIs, we consider that the degree of fulfilment of each of these objectives is a KPI itself. That way, we have 13 single KPIs for this demo: from 1a (ETSI NFV MANO feasibility) to 4b (dissemination to the Open Source community). We will refer the different KPIs using this numbering scheme.

Scenarios

From the 12 different use cases described in WP2, we consider that the scenario that best fits this demo is the *Quality-aware Communications* scenario. In addition, the key principle of (to dynamically moving network functions between the edge/core networks can be also relevant for other scenarios such as:

1. Traffic Jam
2. Massive Nomadic Mobile Machine Type Communications
3. Open Air Festival
4. Industry Control
5. Real-time Remote Computing
6. Vehicle Communications

In all these scenarios, the possibility to dynamically orchestrate the network functions reduces latency and increases throughput when it is required is an essential feature. For example, in scenarios 1, 2 & 3, the high concentration of people in certain areas requires the network to adapt and to configure accordingly to deliver a proper quality. Also, in scenarios 4, 5 & 6, the network functions should be preferably placed in the edge cloud in order to reduce latency and to improve response times.

The other use cases in WP2 have medium or low relation to this demo.

Status and Time Plan

In Annex B, you can find the updated Gantt chart with the different tasks and the current status for this demo. The main tasks are the following:

- Task D2.3.1: Analysis and Design.
  While writing this document, this task has already been completed. It is composed of 4 different subtasks: the creation of an initial conceptual design (Task D2.3.2), the identification of the different open source projects that could be used for the demo (D2.3.3) including the gap analysis between the selected projects and the demo specific requirements (D2.3.4), and finally task D2.3.5 to define the demo platform features such as architecture and topology for the demo, HW and SW resources, and interfaces.

- Task D2.3.6: Demo Platform Setup.
  This task comprises the activities for the demo platform set-up. This includes the orchestrator with the embedded NFVM (Task D2.3.14), the VIM (Task D2.3.10) and all the components in the Network Functions Virtualization Infrastructure block, which includes both hardware (task D2.3.7 & D2.3.15) and software (remaining tasks). This task is almost completed and the only pending task is to add the physical laptop to emulate the Edge Cloud (this is scheduled to be done at the end of the project).

- Task D2.3.16: Development & Implementation activities.
  This task includes the three subtasks regarding software development. The first one (integrate "dummy VNF" – Task D2.3.17) will allow us to start working with the demo platform in an early stage although the real VNFs (which are developed by UC3M) are not available. The second one (D2.3.18) is related to the QoS/QoE control module, which has to be developed specifically for this demo. The third one (D2.3.19) are the improvements in the open source projects that we use for the demo in order to enable the VNF placement. This task has not started yet.

- Task D2.3.20: Integration.
  This task refers to the integration sub-task to incorporate the developments of UC3M (VNFs and Remote Site components) in the VNF MANO platform (subtasks D2.3.21 & 22). This task has not started yet.

- Task D2.3.23: Testing.
  These are the different testing activities related to the previous tasks. We have different integration testing phases depending on the project status and components availability. The

test activities are executed in parallel during the different development stages. This task ends with the final demo setup testing to get the main results and conclusions from this demo. Currently, we are working in one of this testing subtasks (task D2.3.24) which is focused on the TeNOR orchestrator testing activities. The remaining subtasks have not started yet.

4.3 Demo 3. Secured Multi-Tenancy Virtual Network Resources Provisioning via V-AAA

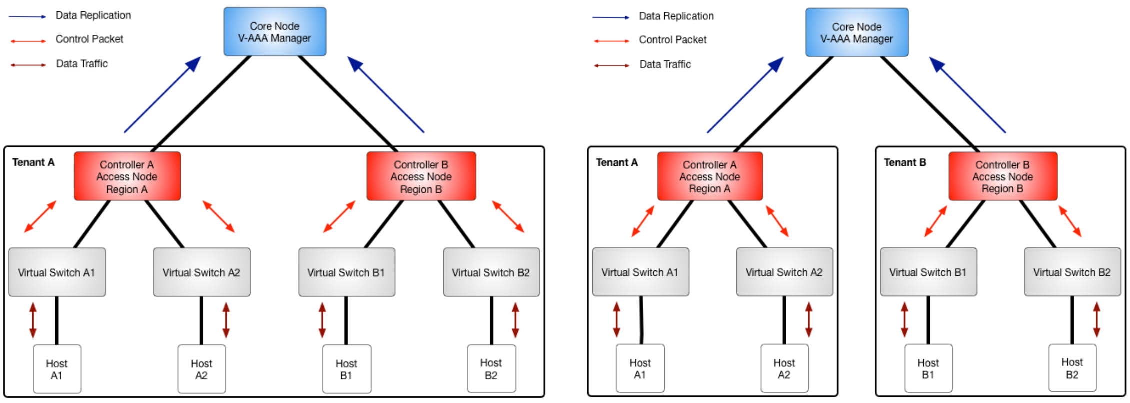
The KCL Virtual Authentication Authorisation Accounting (V-AAA) testbed is a complementary demonstration in 5G NORMA and provides infrastructure for conducting small-scale repeatable experiments of the 5G NORMA architecture, especially experiments that involve secure multi-tenancy and multi-tenant data isolation on the access network (edge cloud). The testbed is based on 1) commodity hardware, i.e., Raspberry Pi, home Wi-Fi router and switch and 2) open source software, i.e., Ryu controller, Open Authentication protocol version 2, couchDB, openVSwitch, that has been configured and extended to provide a hierarchical and distributed database cluster for Tenant isolation and replication of the Tenant data (e.g. billing data, Tenant service logs). This section provides information on the hardware, software, scenarios, and messages exchange of the testbed.

Showcase

In this section, two showcases are developed based on the commodity hardware testbed that is presented in Figure 4-16. These two showcases have been divided into two scenarios, i.e., in a single-tenant scenario and a multi-tenant scenario. More specifically, the multi-tenant scenario has been divided into two parts: direct request of network resources and indirect request of network resources. Mainly, these scenarios are differentiated by the network resource provisioning in deploying a network resource (e.g. virtual switch port) and obtaining the information for billing purpose. Once the network resources have been deployed, the billing information writes into the database and replicates this information to the hierarchical database at the core node. The details of message exchange of all scenarios are presented in the corresponding sections (Single-Tenant Scenario and Multi-Tenant Scenario subsections below).

More distinctly, Figure 4-16 depicts different scenarios with their data flow as a clearer demonstration for understanding the network traffic directions. The single-tenant scenario network traffic direction is represented in Figure 4-16a, whereas the multi-tenant scenario network traffic direction is represented in Figure 4-16b.

These scenario data flows are driven by delegation access and identity protocol, and data replication (many-to-one synchronisation). The delegation access and identity protocol is aimed to provide a verification of a tenant's identity, authorise the tenant to access a specific service, and issue an identity and access token to the tenant. In this testbed, the Internet Engineering Task Force (IETF) Open Authentication protocol version 2 (OAuth v2) and OpenID Connect are applied as the delegation access and identity protocol. Basically, the OpenID Connect and OAuth v2 enable service provisioning application to verify the identity of tenants based on authentication performed by an Authorization Server (AuS), then the AuS issues the ID token and access token to the tenant. The tenant, the AuS and other server interactions and message exchange can be found in the following sections.



a) single-tenant scenario b) multi-tenant scenario

Figure 4-16. A logical entities representation of data flow in single-tenant and multi-tenant scenarios.

Single Tenant Scenario

This is based on single-tenant network resource provisioning and billing information synchronisation. This scenario can be seen as a Mobile Virtual Network Operator (MVNO) when provisioning a network resource on an access network (edge cloud) and over-the-top of a Mobile Network Operator (MNO) infrastructure. On the next page, Figure 4-17 shows the single-tenant network resource provisioning message sequence chart from the tenant's request of the proof of identity to obtaining ID and access token, from sending the access token to deploying the network resource (virtual switch port), and from accessing the database to replicating the data to the core node database. The details of this scenario message sequence chart are as follows:

1. Tenant sends an identity request dispatched to their V-AAA's Identity Provider Server (IdS).
2. IdS gets the tenant's profile from the database and verifies the tenant's identity.
3. Tenant is granted an access code with tenant's identity.
4. A Get Token request is dispatched to the V-AAA's AuS for a specific network resource.
5. Tenant AuS's response is to send ID Token and Access Token with a specific virtual switch, time duration and access rights.
6. Network resource request with the Access Token is dispatched to the tenant's virtual switch for deploying a virtual network port.
7. A validation of identity and token request is dispatched from a virtual switch to the tenant's V-AAA's AuS.
8. A validated confirmation response is dispatched from the tenant's V-AAA's AuS to a virtual switch
9. A network service deployment confirmation is dispatched from the virtual switch to the tenant's provisioning application.
10. A network resource deployment confirmation is dispatched to the tenant. The record of network resource deployment is written to the access node database,
11. The access node databases pushes data replication for the tenant's billing information to the core node database.
12. A billing information request is dispatched from the V-AAA Manager to the database in the core node. Core node database responds and dispatches the complete tenant billing information.

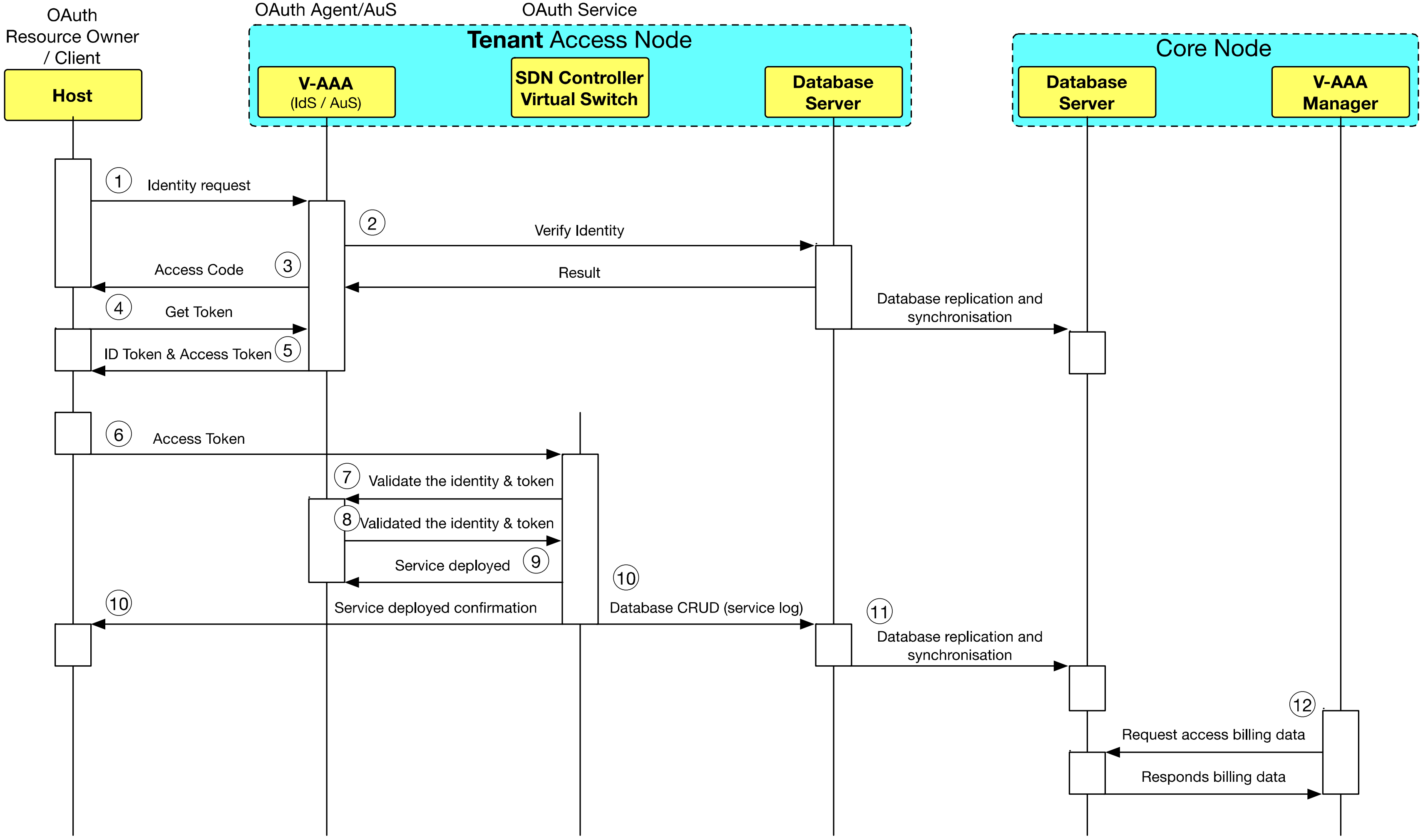


Figure 4-17. Single tenant network service provisioning, billing information replication and synchronisation sequence diagram.

Multi-Tenant Scenario

This section is divided into two multi-tenant scenarios: a direct request of network resources and an indirect request of network resources. Basically, when the tenant verifies the identity and the V-AAA discovers that the tenant has insufficient network resources in their network slice or according to their service-level-agreement, the *direct* and *indirect* requests of the network resources are initiated. The V-AAA might directly ask another tenant's V-AAA or may indirectly request resources from another tenant's V-AAA via a V-AAA Manager. These two scenarios assume that the service discovery has been done by the V-AAA network resource broker.

Direct Request of Network Resources

This is based on the multi-tenant network resource provisioning and billing information synchronisation. This scenario can be seen as an MNO which has two MVNO subscribers (tenant A and tenant B) operating on the same infrastructure. Tenant A requests an extra network resource (e.g. virtual switch port) and initiates an identity validation request message to the V-AAA for requesting the extra network resource. However, when the V-AAA receives the request and discovers that it is unable to fulfil tenant A's request due to insufficient network resources in tenant A's network slice, the V-AAA immediately uses the service broker to broadcast a network resource request to other V-AAAs under the same MNO infrastructure. Once the V-AAA receives a response from (Tenant B) V-AAA, it replies with Access Code information, and the V-AAA generates and sends the access code to tenant A's provisioning application. Tenant A can then use the access code and send a Get Token request to the replying (Tenant B) V-AAA. For simplicity, in this demonstration, we assume this network resource availability discovery process has been achieved and the access token is already obtained via the V-AAA. Furthermore, this access code contains another (Tenant B) V-AAA basic information (e.g. IP address etc.), which allows Tenant A to get an access token and provision the extra network resources as they are needed. Figure 4-18 shows the multi-tenant direct request network resource provisioning message sequence chart from tenant requesting the proof of identity to obtaining ID and access token, from sending the access token to deploying the network resource (e.g. virtual switch port), and from accessing the database to replicating the data to the core node database. The details of this scenario message sequence chart are as follows and illustrated in Figure 4-18.

1. Tenant A and tenant B send an identity request to their V-AAA's Identity Provider Server (IdS).
2. IdS gets the tenant's profile from the database and verifies the tenant's identity.
3. Tenant is granted an access code with tenant's identity. Tenant A V-AAA discovers that there are insufficient network resources and replies that it can use Tenant B network resources with a granted Tenant B Access Code.
4. A Get Token request is dispatched to the Tenant B V-AAA AuS for a specific network resource.
5. Tenant B AuS's response is to send ID token and access token with a specific virtual switch, time duration and access rights.
6. Network resource request with the access token is dispatched to tenant B's virtual switch for deploying a virtual network port.
7. A validation of identity and token request is dispatched from a virtual switch to tenant B V-AAA's AuS.
8. A validated confirmation response is dispatched from Tenant B V-AAA's AuS to a virtual switch.
9. A service deployment confirmation is dispatched to Tenant B's V-AAA.
10. A network resource deployment confirmation is dispatched from virtual switch to Tenant A provisioning application.

11. The network resource deployment confirmation message is dispatched to tenant B. The record of network resource deployment is written to tenant B's database.
12. A network resource deployed log is dispatched to the tenant A's database.
13. The access node databases pushes data replication for tenants billing information to the core node database.
14. A billing information request is dispatched from the V-AAA Manager to the database in the core node. Core node database responds and dispatches all tenant billing information

Indirect Request of Network Resources

This method is based on the multi-tenant network resource provisioning and billing information synchronisation. This scenario can be seen as a MNO which has two MVNO subscribers (tenant A and tenant B) operating on the same infrastructure. Tenant A requests an extra network resource (e.g. virtual switch port) and initiates an identity validation request message to the V-AAA for requesting the extra network resource. However, when V-AAA receives the request and discovers that it is unable to fulfil tenant A's request due to insufficient network resources in tenant A's network slice, the V-AAA immediately contacts and informs the V-AAA Manager to request extra network resources and generates a V-AAA Manager's access code to tenant A.

When V-AAA Manager receives the request for extra network resources from tenant A's V-AAA, V-AAA Manager uses the service broker to broadcast a network resource request to all V-AAAs under the same MNO infrastructure. At this stage, tenant A has no knowledge where actually the network resource has been allocated nor which tenant has been provided with the network resource. However, on the other hand, once the V-AAA Manager receives the response from (Tenant B) V-AAA's network resource availability, the V-AAA Manager has the knowledge. Meanwhile, tenant A uses the access code and sends a Get Token request to a V-AAA Manager, then the V-AAA Manager forwards the Get Token to the replying V-AAA. The replying V-AAA (Tenant B) generates an access token for the request and sends it to a V-AAA Manager and then the V-AAA Manager forwards the access token to tenant A. For simplicity, in this demo, we assume that this network resource availability discovery process has been achieved and the access token is already obtained via the V-AAA Manager. Figure 4-19 shows the multi-tenant *indirect* request network resource provisioning message sequence chart from the tenant's request of the proof of identity to obtaining ID and access token, from sending the access token to deploying the network resource (e.g. virtual switch port), and from accessing the database to replicating the data to the core node database. The details of this scenario message sequence chart are as follows and illustrated in Figure 4-19.

1. Tenant A and tenant B send an identity request to their V-AAA's Identity Provider Server (IdS).
2. IdS gets the tenant's profile from the database and verifies the tenant's identity.
3. Tenant is granted an access code with tenant's identity. Tenant A V-AAA discovers that there are insufficient network resources, and replies a V-AAA Manger access code.
4. A Get Token request is dispatched to the V-AAA Manager AuS for a specific network resource from another (tenant B) V-AAA.
5. Tenant B AuS's response is to send ID token and access token with a specific virtual switch, time duration and access rights to a V-AAA Manager and the V-AAA Manager forwards the access token to tenant A.
6. Network resource request with the access token is dispatched to tenant B's virtual switch for deploying a virtual network port.
7. A validation of identity and token request is dispatched from a virtual switch to tenant B V-AAA's AuS.
8. A validated confirmation response is dispatched from tenant B V-AAA's AuS to a virtual switch.
9. A service deployment confirmation is dispatched to tenant B's V-AAA.
10. A network resource deployment confirmation is dispatched from a virtual switch to tenant A provisioning application.

11. The network resource deployment confirmation message is dispatched to tenant B. The record of network resource deployment is written to the tenant B’s database.
12. A network resource deployed log is dispatched to the tenant A’s database.
13. The access node databases push data replication for tenants billing information to the core node database.
14. A billing information request is dispatched from the V-AAA Manager to the database in the core node. Core node database responds and dispatches all tenant billing information.

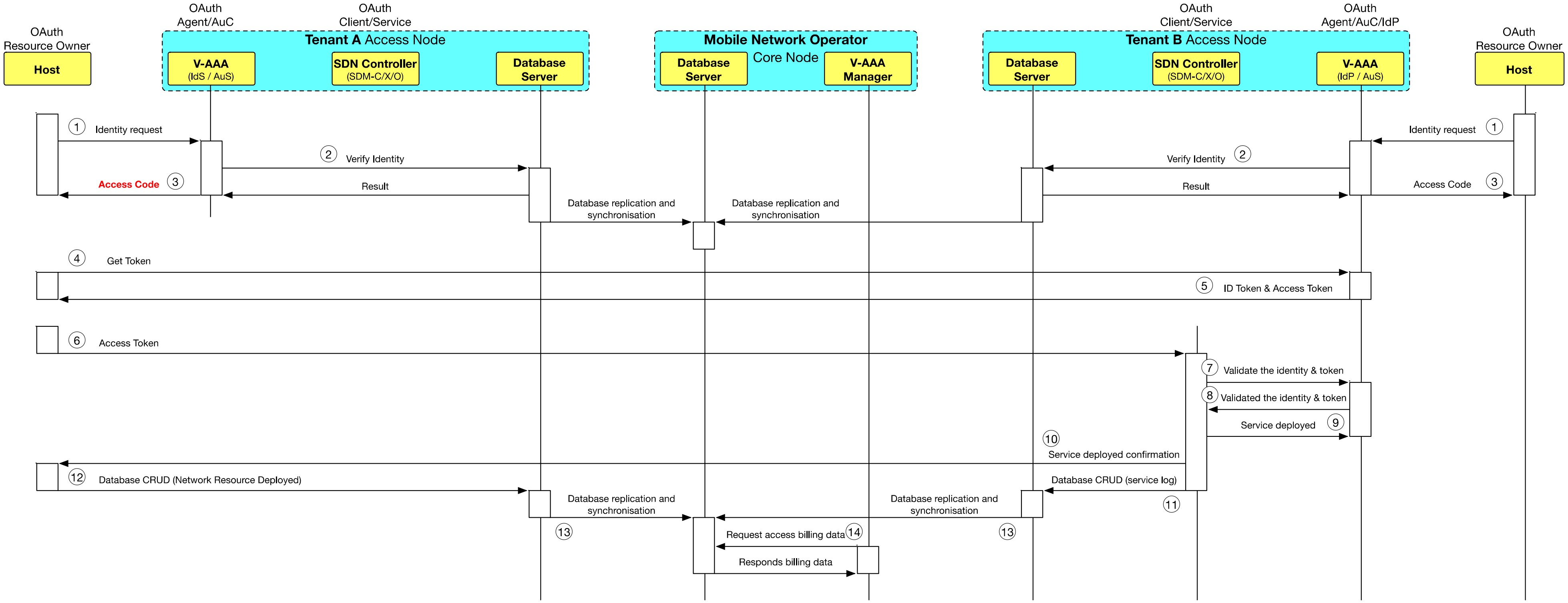


Figure 4-18. Multi-tenant direct network resource negotiation scenario messages exchange sequence.

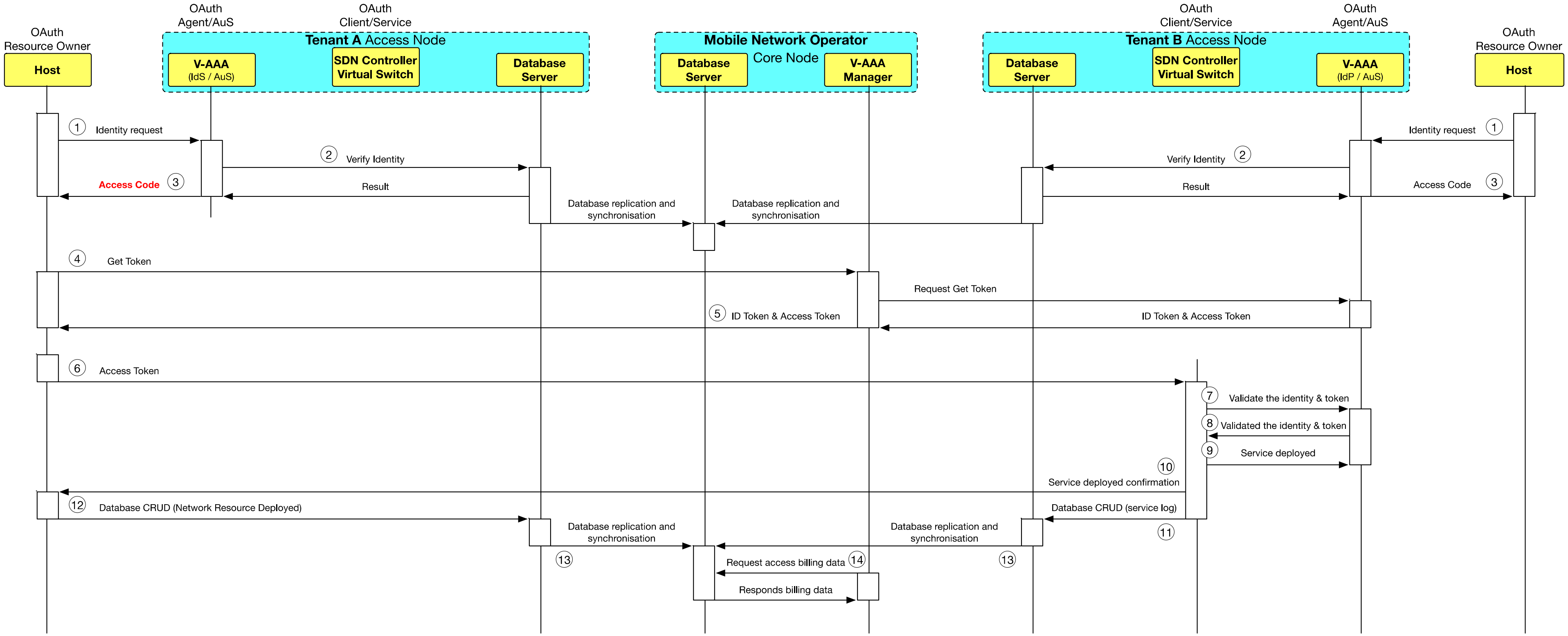


Figure 4-19. Multi-tenant indirect network resource negotiation scenario messages exchange sequence.

Testbed Architecture

Tenants might violate the data protection policies, confidentiality policies, and illegally use other tenants' subscriber information. Therefore, data between tenants must be securely isolated from each other in a network infrastructure sharing setup. The aim of this demo is to reveal the flexibility of the 5G NORMA architecture and its support for data isolation of different tenants, e.g. billing, lawful interception, marketing evaluation, or self-organising network.

The actual tenant data which is subject to isolation may be related to customer data, network service provisioning data and other important system transition data that will not be visible to other tenants under the same MNO. Furthermore, tenant isolation mainly avoids that tenants use each other's data illegitimately. While a tenant's data is isolated, the MNO would have a complete set of data from each tenant. Figure 4-20 illustrates the 5G NORMA V-AAA architecture that has been divided into two levels of administration (MNO administration and tenant administration) and three domains (the core network domain, access network domain and host domain). These three domains reflect the essential architecture of a telecommunication system with central cloud and edge cloud, and the two levels of administration reflect the flexible architecture that can enable multi-tenancy and multi-network slicing, and the effectiveness of 5G telecommunication network management.

This testbed also shows the power of modern database applicability in data availability in two types of structures: hierarchical structure and distributed structure. The data availability in hierarchical structure exists between the core node and access nodes, or MNO and tenant. The collected data in the hierarchical database can be used for billing of tenants. The distributed structure exists between the access nodes belonging to a tenant, or exists between a tenant's network slices under same MNO or different MNO. The collected data in the distributed database are under a tenant's administration at the edge cloud, which can be used for tenant's end-user billing purpose. For simplicity, this testbed only considers each tenant would have one network slice and does not share the physical resources with another network slice.

The testbed will make use of commodity hardware for the demonstration of the softwarisation. This commodity hardware testbed network topology can be found in Figure 4-21 that is mainly based on Raspberry Pi, household Wi-Fi router and Ethernet switch. Based on this commodity hardware, we propose both a basic scenario with a single-tenant and a more complex multi-tenant scenario. Figure 4-21 gives the commodity hardware topology of this testbed that can be configured and reconfigured through open source software to illustrate these scenarios.

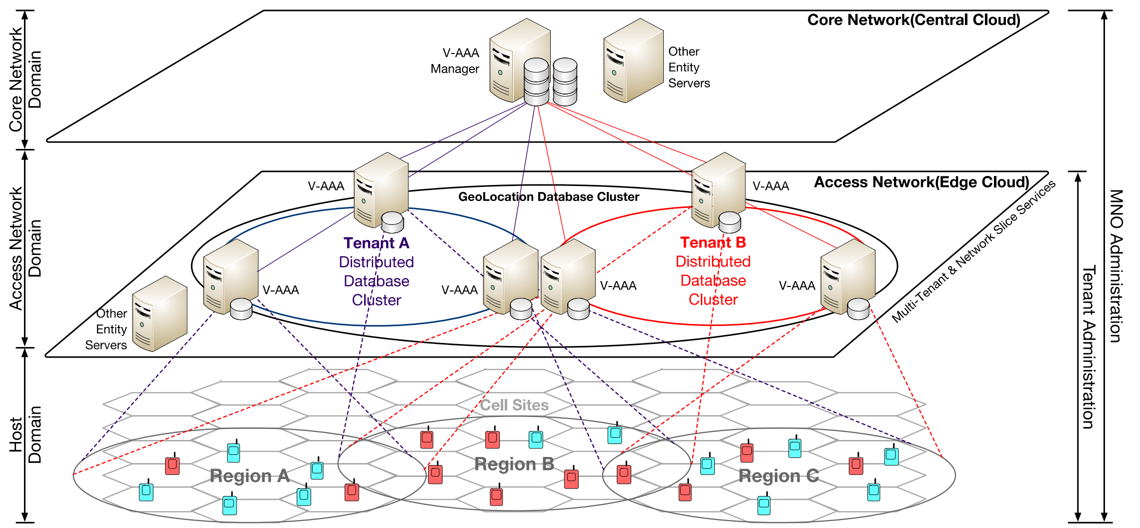


Figure 4-20. Hierarchical / distributed database cluster for multi-tenancy network slicing.

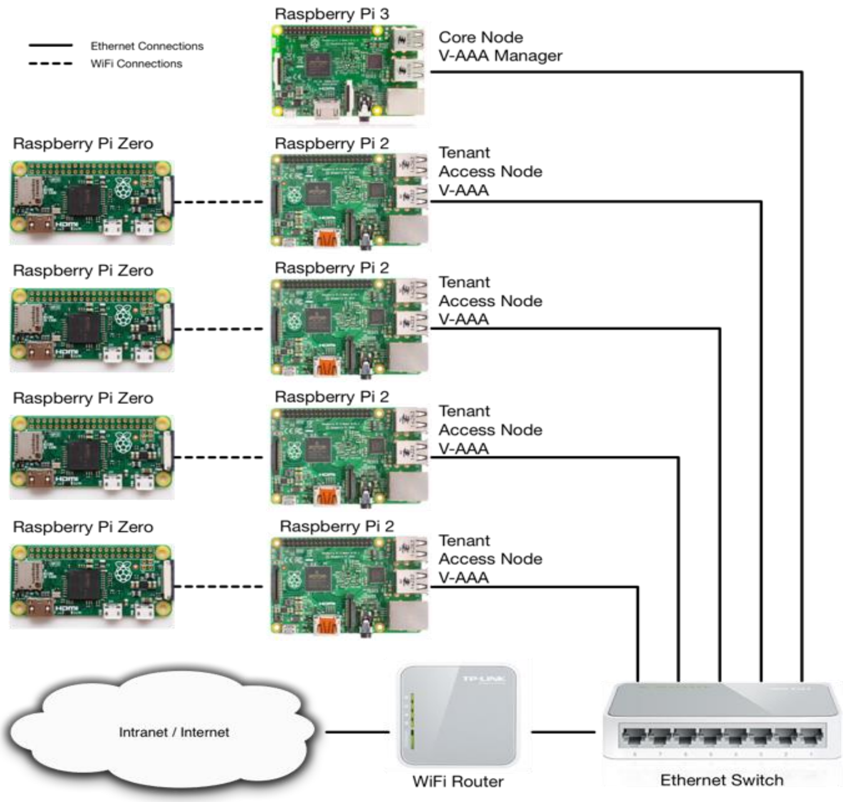


Figure 4-21. Commodity hardware testbed network topology

Hardware and Software

In this section, we provide the testbed hardware and software specifications. The testbed's commodity hardware is presented in Table 4-5 and open source software is presented in Table 4-6.

In the previous section, three billing scenarios were presented, which will be implemented based on the open source library in Table 4-6. Furthermore, a Graphical User Interface of a service provisioning and billing system will be implemented in Python for showing the message details, logs, database entries, and billing information. The source code of these scenarios, configuration of software and setup instruction will be released to the public based on the Apache 2.0 license. A number of custom Raspbian operating system live bootable images with source code and specific configurations instruction to each logical element in the testbed will be available on the 5G NORMA website.

Item	Device	Logical Name	Model Number	Quantity
1	Raspberry Pi 3	Core Node	Model B	1
2	Raspberry Pi 2	Access Node	Model B	4
2	Raspberry Pi Zero	Host	Zero	4
3	TP-Link 8-Port 10/100Mbps Switch	-	TL-SF1008D	1
4	TP-Link 300Mbps Wi-Fi Pocket Router/AP/TV Adapter/Repeater	-	TL-WR810N	1

Table 4-5. Commodity HW list of Secured Multi-Tenancy Virtual Network Resources Provisioning via V-AAA testbed.

Item	Software	Version	Address
1	Ryu SDN Controller	4.6	https://github.com/osrg/ryu
2	Open Virtual Switch	2.5	http://openvswitch.org
3	CouchDB	1.6.1	http://couchdb.apache.org
4	Python	2.7.x / 3.5.x	https://www.python.org
5	OpenID Connect	1.0	http://openid.net

Table 4-6. Open source SW list of Secured Multi-Tenancy Virtual Network Resources Provisioning via V-AAA testbed.

Objectives and KPIs

The main objective of this demo is to isolate the data and billing information between tenants. Tenants would not be able to obtain other tenant's data without permission. On the other hand, the MNO would securely acquire a complete set of data from the tenants. Another objective is to provide a secured network resource provisioning from direct and indirect requests between tenants or via the V-AAA Manger. However, the billing information securely remains isolated between tenants and the MNO still has the complete set of data and billing information.

To measure or evaluate these objectives, the data and billing information can be retrieved from the tenant's administrative domain and MNO administrative domain; then the assertion of the data set and the billing information can be compared or evaluated on a GUI. Furthermore, the secured network resource provisioning evaluation can be done by comparing the original information and the data inserts to the database, and every message during the message exchange can also be visualised in the GUI.

Scenarios

There are twelve use cases presented in deliverable D2.1. In this demonstration, we particularly focus on the enhanced mobile broadband and fixed-mobile convergence use cases.

For instance, in the enhanced mobile broadband use case, which is characterised in providing faster services at higher quality, anywhere and at any time. This includes the situation when the user density suddenly increases in a specific area and the tenant's or MNO's infrastructure requires more network resources to cope with the high demand of mobile broadband services. Additionally, the operating mobile broadband services should not be interrupted while extending or increasing the network resources for dealing with the high demand of network traffic. Of course, a secured method for provisioning the network resources and billing or exchanging the billing information is designed in this use case.

Another example is the fixed-mobile convergence use case. Here, 5G services should be delivered via a converged access-agnostic core, i.e., where identity, mobility, or security are decoupled from the access technology. 5G should have a unified database and information system architecture across all network slices. This unified database and information system architecture allows the MNO to process all billing information and only allow Tenant(s) to access their customer billing information.

Based on these use case requirement, we have developed a unified database and information system architecture that is called 'hierarchical / distributed database structure' which provides tenant isolation, tenant data and billing information isolation. Particularly, the hierarchical / distributed database structure also allows MNO to obtain all tenant data and billing information.

Status and Time Plan

In this demo, we use the *agile* management methodology, which has a series of iterations, i.e., analyse, design, develop and test. The overall Gantt chart with all the tasks can be found in Annex A. The time plan has been divided into four iterations of agile development cycle:

First iteration of agile development cycle:

Analyse: task D3.1 runs from M1-M9 of the project and focuses on examination of protocols, study of telecommunication standards and their authentication methods that can be extended or applied to the demo such as diameter protocol, 3GPP LTE AAA, ETSI NFV authentication, and geolocation database concept.

Design: task D3.2 runs from M9-M11 of the project and focuses on the design of hierarchical and distributed database structure and the authentication process applying to multi-services scenarios as description in Section 2 and shown in Figure 4-20.

Develop: task D3.3 runs from M11-M13 of the project and it focuses on the development of the demonstration scenarios based on the IETF Open Authentication Protocol version 2.0 (OAuth v2).

Test: task D3.4 runs from M12-M13 month of the project, testing the basic message exchange.

All tasks from this first iteration are already completed.

Second iteration of agile development cycle:

Analyse: task D3.5 (M13-M14) on analysis of the demonstration software entities (e.g. V-AAA, V-AAA Manager, OpenVSwitch and SDN controller), examination of the interaction details between software entities and feasibility studies on the secured protocol technique (e.g. Transport Layer Security –TLS– and Datagram Transport Layer Security –DTLS– protocol), analysis of different types of database (based on the Structured Query Language –SQL– and Not only SQL –NoSQL–) that is suitable for the hierarchical and distributed database structure for tenant isolation, and tenant data and billing information isolation, are processed.

Design: task D3.6 (M13-M16) on design of a data model which is suitable for the chosen database, tenant isolation and tenant data and billing information isolations, design of a hierarchical / distributed database cluster structure in the testbed, and planning of the integration and invocation the database server with OAuth v2, are performed.

Develop: task D3.7 (M15-M17) on implementation of the data model with a chosen database and integration of the database with OAuth v2, writing of a script for configuring and reconfiguring the demonstration software entities, and development of a method invocation between logical entities, will be performed.

Test: task D3.8 (M17-M18) on testing of the database cluster that can deliver many-to-one data replication, database replication queries in: store, retrieve and update. Also, the OAuth v2 message exchanges between the demonstration software entities (e.g. V-AAA, V-AAA Manager, OpenVSwitch and SDN controller) will be performed.

Third iteration of agile development cycle:

Analyse: task D3.9 (M19) on examination of a suitable threat model, and right access control and identity control methodologies to the OAuth v2

Design: task D3.10 (M19) on design of the evolution process, the data structure (using JavaScript Object Notation –JSON–) of network resource attribute (value token) and the threat testing scenarios

Develop: task D3.11 (M19-M21) on development of a simple access control and identity provider server (IdS) and establishment a threat testing scenario

Test: task D3.12 (M21-M22) on testing of the system with various roles and identity access of the AuS and usage of various roles to access network resources

Fourth iteration of agile development cycle:

Analyse: task D3.13 (M22) on investigation of various types of charts that are suitable for this demonstration's data visualisation

Design: task D3.14 (M22-M23) on design of the GUI for capturing the security log indicator

Develop: task D3.15 (M23-M25 and M28-29) on implementation of the GUI security log indicator and refinement of the software

Test: task D3.16 (M24-M29) on testing of the single-tenant scenario and the multi-tenant scenarios with the proposed approach in this section

4.4 Demo 4. Online Interactive 5G NORMA Business Cases Evaluation Tool

Showcase

The online interactive tool will demonstrate the key outputs and sensitivities of the evaluation cases selected as part of the overall project evaluation framework (Deliverable D2.2). The tool will show the economic feasibility of the 5G NORMA architecture by looking at three different evaluation cases. The three 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
- Evaluation case C.3: Multi-service evaluation case – comparison of single and multi-service networks covering MBB, V2X and Critical Machine Type Communication (uMTC) services

The final network cost modelling is intended to be based on real-life areas that are representative of areas with different geographic features (population density, clutter, and others) called *geotypes*. We expect that the final version of the tool will cover different geotypes such as urban, suburban and rural. However, the initial network cost modelling focuses on one area: Central London, which is representative of the urban geotype.

The socioeconomic analysis will provide for the tool metrics based on the increase in profitability and the cost savings that the 5G NORMA infrastructure can deliver compared to the relevant comparator network in each evaluation case.

In addition to the headline metrics for economic validation, we expect that the tool will report a number of other results and intermediate outputs such as:

- Operating and capital expenditure, service revenues, return on investment
- Changes in volumes of equipment deployed over time
- Utilisation of spectrum over time
- Utilisation of network elements over time.

Interactive tool – approach

The interactive tool will use a web-based interface. The values provided 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 will use the 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. We will characterize these in order to model how certain parameters (and different 5G capabilities) would influence the outputs, and seek to identify limiting cases. Key attributes expected to be of interest will examine the impact of the key 5G NORMA innovations with variations in attributes such as site density, different services, site chain type, traffic growth, and different sharing options. The tool should be able to demonstrate that 5G NORMA innovations support lower cost and more capable networks that can support the full range of traffic classes.

Interactive tool – architecture and development environment

The demonstrator will be an online tool with a user friendly web interface that will be created using mainly Hypertext Mark-up Language (HTML). The background algorithm will be developed using a server based language like PHP (PHP Hypertext Pre-processor) that interacts with a database (possibly written in MySQL). PHP is a widely used open source general-purpose scripting language that is primarily used for web development and can be embedded into HTML. In addition, MySQL is an open source relational database management system based on Structured Query Language (SQL). We are not planning to release the code developed as an open source as the tool is going to be used specifically for 5G NORMA.

Specific tools will be reviewed and selected based on the sites that will be used to host the interactive tool. The user will be able to select from a dropdown menu the required options. The algorithm will contain a series of SQL queries that will interrogate the database according to the specifications made by the user. The output will be a series of graphs and tables containing the relevant information. The architecture of the demonstrator is shown in Figure 4-22 below.

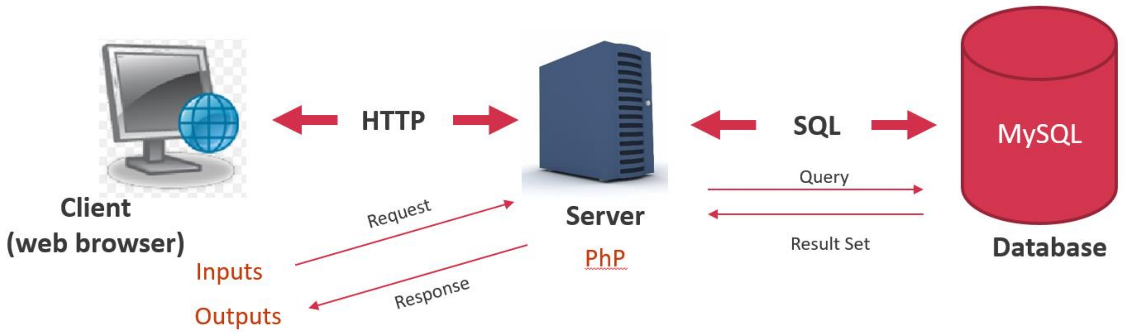


Figure 4-22. Architecture of the demonstrator

The demonstrator will be saved on a server, e.g. on a RW server or possibly the 5G NORMA website. We will need to identify what database (or equivalent) would be available on the 5G NORMA site to avoid issues during the integration of this self-contained tool.

Objectives and KPIs

The objective of the tool is to demonstrate whether the economic case for the 5G NORMA architecture is validated according to the project evaluation cases and to enable users to understand the key assumptions and sensitivities on which the analysis is based.

We expect the modelling will demonstrate, among other things, that the 5G NORMA innovations support lower cost networks than would otherwise be the case and the impact of the functionalities that can be supported by the 5G NORMA infrastructure but not evolved 4G infrastructures.

The tool will allow the user to change certain parameters to see the impact on the evaluation cases. A range of sensitivities will be run for each evaluation case and these results will be fed into the tool. In other words, the results of the sensitivity analyses will be pre-loaded into the tool and the user will be presented with pre-set options for changing parameters.

The types of parameters that we expect the user may be able to change in the model include:

- The difference between 5G and LTE-A Pro ARPU (Average Revenue Per User) by service, i.e., varying from pessimistic to optimistic assumptions of the additional value delivered by 5G network capabilities
- The growth rate of service demand in broad lower, central and higher sensitivities
- Up to 3 different configurations of antenna, edge cloud and central cloud sites reflecting variations in the distribution of processing load between the edge and core
- Variations in the amount of spectrum available in discrete bands, e.g., above and below 6 GHz
- Variations in the unit costs of a limited number of key equipment types.

Scenarios

The scenarios are the three project overarching evaluation cases (as described in the ‘Showcases’ section above), and the tool will be built around them.

Status and Time Plan

The economic validation tool is dependent on the delivery of the socio-economic analysis in WP2. A Phase 1 network cost model (based on an evolved 4G radio access network with adjustments for expected 5G developments such as increased spectrum efficiency) has been completed and written up in deliverable D2.2 [25]. The approach to develop a network cost model that reflects the key 5G NORMA network architecture innovations is defined in D2.2. As a result, work on the initial demonstrator design is now in a position to begin in 4Q16. 5G NORMA begun to investigate the tools needed to develop the web-based interface and to discuss the initial specification for the tool. Finishing these tasks is highest priority.

The plan is to develop the code using the standard client-server approach. The client will use a web browser, while the application will sit on a server (hosted on the RW website and possibly also accessible from the 5G NORMA website as previously mentioned). A first prototype of the front end and basic capability is planned for November 2016 and a second for December 2016. The first software release, populated by initial results from the Phase 2 network cost modelling is scheduled for March 2017.

The modelling will be further developed over the remaining lifetime of the project as 5G NORMA produces a more detailed description of its architecture (WP3, 4, & 5), as evaluation cases are added and as other geotypes are considered in the modelling.

In Annex A you can find a detailed time plan with the most relevant tasks for this demo.

4.5 Integration

Nomor: Integrating SDMC with Hardware eNodeB

The *Native Multi-service Architecture* demo will require an integration of software and hardware elements from *Nomor* and *Azcom*, respectively. The combined demo is planned to be integrated in multiple smaller steps instead of performing everything in one final step. This approach is proposed because it enables both partners to detect and solve any possible inconsistencies in integration at an earlier stage. It also gives the partners a better idea at an earlier stage about the possible pitfalls or weak points about the demo integration and provides them with ample time to fix or avoid such problems. The main steps of integration process can be summarized as following:

1. A novel communication protocol shall be designed and mutually agreed by Nomor and Azcom, as a first step. This protocol is responsible for taking care of all communication between SDMC and the hardware eNodeB(s), and is therefore a first step for the integration of both entities.
2. The mutually agreed communication protocol shall be implemented by both partners. The communication protocol is then integrated with the SDMC application by Nomor; and with the hardware eNodeB by Azcom.
3. The next phase is the preliminary testing of communication between SDMC and hardware eNodeB. This task shall be performed with Nomor by developing and providing a basic working SDMC application to Azcom. This lets Azcom to run the SDMC application at their test-bench and verify the correct exchange of communication messages between both entities.
4. Based on the results from preliminary testing, some necessary changes and improvements shall be performed at both ends with mutual agreement.
5. During the next phase, i.e., during further development of SDMC application and its intelligent decision logic, some new communication messages might need to be integrated into the communication protocol. In that case, both partners shall revise and evolve the existing communication protocol, and also their applications in order to define new messages.
6. Upon completion of SDMC's optimization logic, a more complete test bench shall be set up for testing of the demo. This will be an on-site testing, i.e., both partners Azcom and Nomor shall be present on the site of test-bench during testing. This integration phase shall test and evaluate all main aspects of the demo and not only the communication part. These main aspects involve evaluating the correct functioning of all network elements, i.e., SDMC, hardware eNodeB, and the connected UEs. Beside, this step of integration shall also test the effectiveness of SDMC's optimization logic and the resulting KPIs.
7. As a final step for integration, a demo storyline shall be developed, based on the results from previous integration and testing phases. This storyline shall be designed to demonstrate the usefulness and benefit of SDMC for the future 5G mobile networks and to show the advantages of using the envisioned network (de)composition between edge-cloud and network-cloud.

Integrating Orchestration and Network Functions

Demo 2 is composed of two parts: the network slicing framework down to the RAN provided by UC3M and the QoE/QoS aware orchestration implemented by ATOS. The two demonstrators are designed to be eventually integrated in a natural manner to emphasize even more the innovations that are planned to be shown.

The UC3M demo part focuses on the network functions of both CN and RAN: by allowing different RAN functional splits and implementing network function sharing techniques, it provides the fundamental building blocks that can finally be used by an advanced network orchestrator, such as the one implemented by ATOS, to provide service aware orchestration. As described in Section 4.2, the last part of the activities for UC3M will be devoted to the finalization of the demo, which includes testing and debugging under different situations and loads, but also the integration with the ATOS orchestration. As described in Section 4.2, the orchestration of the network functions used to provide the two services (i.e., Mobile Broad Band and Low Latency) can be seen as a static setup, i.e., the scenario is allocated with each network function already placed in the right computing cloud. The integration with the ATOS NFV MANO implementation will allow to create this scenario on-demand, adding elasticity to the scenario and providing a more natural use case for a real network scenario.

In order to integrate the two demonstrators, the following steps need to be fulfilled:

- A detailed requirement list that has to be considered by the orchestrator while performing the network function placement operation. Among the requirements that must be fulfilled by the orchestrator there are the objectives *Minimum Throughput* or *Maximum Delay* among network functions, or incompatibility of two network functions of being instantiated into the same physical machine.
- The set of VNFs used by UC3M should be provided to the ATOS Orchestrator, creating hence two *blueprints* associated to each service.

This interaction is planned to be performance iteratively with a strict interaction between the NF provider and the Orchestrator. The current plan is as follows (see Figure 4-23below).

UCM3 will provide:

- The VNFs to be integrated in the Atos ETSI NFV MANO implementation, i.e., Radio and Routing Components functions.
- The remote site, consisting on the Ettus USRP B210 connected to a controller computer (a general purpose laptop).
- End-User devices. They will be physical devices (a mobile phone or a dongle LTE).

On the other hand, Atos will provide:

- General purpose laptop(s) to emulate Edge Cloud and Core Network sites. We are considering also the possibility to emulate the Core Network using a remote connection to a testbed located at Atos. This possibility is more realistic because the core network and edge networks will be distant, as it happens in a real deployment. The resulting inconvenience is that we depend on a remote connection to perform the demos.
- Server to execute the NFV ETSI MANO implementation.

Figure 4-23 shows the final deployment for the demo with the responsibilities for each partner.

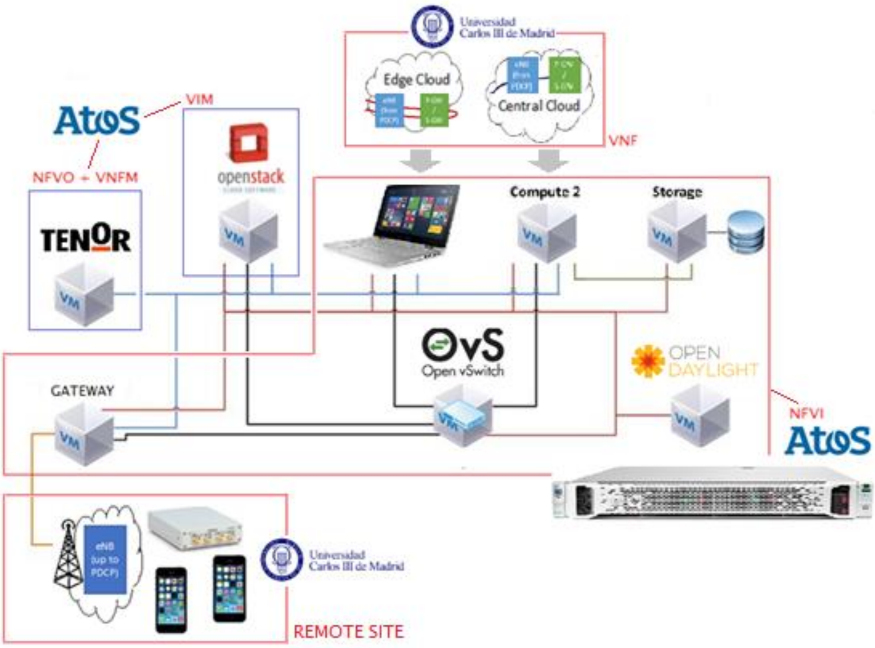


Figure 4-23. UC3M-ATOS Integration Plan

More specifically, the following tasks are envisioned:

Demo 2. Service-aware QoE/QoS Control (Atos / UC3M)	Task
Common Tasks	
Identification of Virtual Functions to be included in the PoC platform (T6.1)	D2.1.1
Identification of services to be included in the PoC platform	D2.1.2
Definition of experiments that need to be performed (step-by-step demo procedure). From task T6.1	D2.1.3
└ Definition of associated KPI-based expected results for the PoC success.	D2.1.4
Module testing and preliminary results (T6.2)	D2.1.5
└ Definition of the virtual containers format	D2.1.6
└ Definition of the per slice orchestration strategies	D2.1.7

Table 4-7. Demo2 Integration Subtasks

Detailed time-plans are provided in Annex A. Below the description of each task:

D2.1.1 – D2.1.3

Those tasks were related to the definition of the joint demonstrator itself and were performed for the description of the Demo 2 available in Section 4.2. However, the definition may still suffer slight changes according to the development process.

D 2.1.5 – D2.1.7

These tasks are related to the actual HW/SW integration between UC3M and ATOS parts. Details such as the format of the virtual containers or the orchestration strategies have to be made by consensus in order to fully demonstrate the intended 5G NORMA functionalities. Especially, the orchestration strategies have to be carefully designed in order to take into account the requirement associated to the different slices. This work is expected to take place in 2017.

5 Improvements and expected outcomes

Nomor: Native Multi-Service Architecture Demo

Nomor's major contributions with regards to Work-Package 6 are in Demo 1, i.e., the *Native Multi-service Architecture* demo. The main expected result to be gained from this demo involves showing the benefits and effectiveness of having network function (de)composition for future 5G networks. This involves demonstrating how the network reconfiguration between network-cloud and edge-cloud can help to improve performance.

For the evaluation of the expected gains, two main KPIs shall be examined, i.e., achieved user throughput and end-to-end latency. The demo is planned to show the effect of network (de)composition on these KPIs in different scenarios and use cases. For this purpose, different demo setups shall be created. Network (de)composition is controlled by the Software Defined Mobile Network Controller, based on different active services in the network. It is envisioned that service-aware network (de)composition performed by SDMC should improve the end-user performance in terms of the throughput and latency KPI. Hence, the results from the demo shall be gathered and presented in the form of throughput and latency KPIs for different cases and different network configurations. The effect of the presence of certain services shall also be taken into account during the evaluation of these results.

The results of the demo are expected to reveal a number of important lessons and conclusions. First, the demo presents a proof of concept for the co-existence and functioning of SDMC alongside other network elements such as eNodeBs and UEs. In addition, the demo shows the advantages of having a flexible network architecture and practically demonstrates the concept of network reconfiguration between central-cloud and edge-cloud. The demo shall also reveal some of the real-world constraints and drawbacks of using different network configurations in different scenarios.

Azcom

Azcom hardware demos are composed of two different releases, the first one already shown at the Mobile World Congress 2016 in Barcelona and a second one, which will be delivered jointly with the Nomor software demo. The MWC 2016 demo was focused on the function de-composition and relocation of network functions. In particular, the demo focused on the placement of the S-GW adaptive to the corresponding latency requirements. In order to demonstrate this, the current EPC components were moved to the eNB where the complete data layer was processed in a system on chip. Depending on the latency requirements, the U-Plane processing can be assigned either centrally or at the eNB.

The basic idea of the second release of the hardware demo (Native Multi-service Architecture, see Section 4.1.1) is to control the MAC scheduler of a LTE based eNB, by a scheduler controller, a SDMC-like component, provided by Nomor. It aims to adapt the resources block for different users depending on the specific service and its requirements. New scheduler policies could be implemented in software and then provided to the eNB using specific interfaces, which is work in progress.

Both demos aim to show the flexibility of the new architecture to satisfy different QoE/QoS. In particular, the hardware demo described in Section 4.1.1 covered the requirement group 8 (QoE/QoS awareness) and 10 (Low latency support) as defined in D2.1. The application of remote driving falls in "Real-time remote computing" use case of D2.1 and the evaluated KPI was the E2E latency.

The second release of the hardware demo aims to cover the requirement group 1 (fast network reconfiguration within a network slice), the requirement group 8 (QoE/QoS awareness) and group 10 (Low latency support) as defined in D2.1. For the evaluation of the expected improvements, achieved user throughput and end-to-end latency shall be inspected. In addition, expected input for these activities is feedback from WP3, WP4 and WP5 on needed evaluations.

Nomor and UC3M Collaboration for Large Scale Scenario Simulations

In this task, Nomor and UC3M collaborated for running simulations on a large scale scenario. UC3M gathered real-world mobility data from the city of Bologna in Italy. Those data, in form of SUMO mobility files, were used to generate the mobility traces using SUMO. These traces, containing information about movement of cars along the roads of Bologna city, could be imported into Nomor's network simulator in order to execute network simulations using LTE as the radio access technology. This way, it was possible to simulate the network performance of LTE for a real-world city. In addition to the mobility data of cars, UC3M has also gathered information about the actual placement of eNodeB base stations inside the city of Bologna for four different operators.

The data gathered from this real-world scenario contains mobility information for the main metropolitan area of the city of Bologna. The data contains a very large number of cars, i.e., user equipment for the simulation. In addition, there are also a large number of eNodeB base stations located in the city area. Nomor's network simulator is a full system-level simulator and simulates at TTI-level granularity, i.e., it needs to perform very complex computations. Increasing the number of elements, i.e., eNodeBs and UEs, increases the complexity of such computations exponentially and makes the simulation slow down. Because of these reasons, some changes were needed to be made in the system simulator before such large scale simulations could be carried out.

For carrying out these large scale simulations, the following major tasks were performed:

1. *Increase the Capability of Simulator:* In this task, some changes were made to the simulator code to be able to support a larger number of UEs.
2. *Remove the Central Controlling Unit (CCU)*: The CCU was a part of the simulator that was responsible for taking care of mobility modelling. However, it was observed that such a division of tasks between CCU and the simulator was slowing down the operation. Therefore, the CCU was completely removed from the picture and all mobility functionality was integrated inside of the simulator that resulted in significant speed improvements.
3. *Porting the Simulator to a 64-bit architecture:* Because of the very large number of eNodeBs and UEs, the simulator needed to be able to access more system memory. However, the maximum memory that a 32-bit application can access is limited to 2GB. Therefore, the system simulator was ported from 32-bit to a 64-bit application to overcome this limit.
4. *Fixing Hazard Computation Algorithms:* The car-to-car communication, which is the main focus of these simulations, makes use of a hazard computation algorithm. This algorithm computes hazard states for different UEs, which result in triggering small packets from cars in hazard state. These packets are then transmitted to other cars via the interface (V2X). This hazard detection algorithm becomes complex as the number of UEs increases because of the exponentially increasing number of UE-pairs. Because of these reasons, some changes and optimizations were made to this hazard detection algorithm.
5. *Converting Buildings Data:* The buildings in the city of Bologna also need to be imported into the system simulator in order to model shadowing caused by buildings for path-loss computations. However, the buildings data should be ideally available in the form of Open Document Architecture (ODA) files. But for Bologna simulations, the

buildings data was available in the form of Open-street-maps. Therefore, the data should be first converted into the right format for the simulator. The building coordinates were also translated and shifted in order to align them perfectly with the mobility traces.
On the other hand, the buildings data for the entire Bologna map also contained a very large number of buildings that was not possible to use in the simulator. For this reason, some changes were made to remove the limit of the maximum number of buildings supported by the simulator.

6. *Tracing Key Performance Indicators:* The KPIs resulting from the simulations are traced in the form of binary files that can be read by Matlab for further processing. Because of the very large number and size of these files, tracing was limited to only a limited number of important and relevant KPIs.
7. *Plotting results in Matlab:* The results from the trace files were imported in Matlab and plotted in the form of time plots and Cumulative Distribution Functions (CDFs). This not only helps in examining the results in a better way and for including them in a research paper, but also reduces the size of data for exchanging with other partners.
8. *Bug fixing:* One of the major challenges of this task was to keep track of bug reports such as memory leaks or segmentation faults, and to fix those bugs.

ATOS: Evaluate Open Source Mano

Although, as explained in Section 4.2, we have chosen TeNOR as VNF Orchestrator and VNF Manager, we are also considering OSM (Open Source Mano) [5] as a backup or test alternative for this module.

5G NORMA may also evaluate the OSM project if sufficient resourcs are available. Also, we could use OSM as backup in the case of problems with using TeNOR. Although OSM is a very promising project, we've selected TeNOR because there were not sufficient resources to test the new OSM platform (OSM release 1 has been announced just one week before delivering this document, i.e., Oct. 2016). In addition and similar to TeNOR, OSM still doesn't meet all requirements that are imposed by the demo (especially the VNFs placement function).

6 Conclusions

In this document, we described: i) the activities performed within 5G NORMA WP6 so far, and ii) the envisioned demonstrator plans and the roadmap towards their integration. More specifically, we explain how the proposed demonstrator will help to assess both the feasibility and the performance increase of most of the innovative concepts proposed by 5G NORMA:

- Software Defined Mobile Network Control: the concepts of network control using the SDMC is demonstrated by Demo 1 in which the slow scheduler controls through a well-defined interface the fast scheduler, using an approach that can be easily extended to the SDMC application concept
- Network function (de)-composition and placement: demonstrated by Demo 2, in which the network functionality is split into several blocks, which can be orchestrated in several ways according to the service. Orchestration is one of the main focuses in Demo 2.
- Multi-service and context-aware adaptation of network functions: demonstrated by Demo 1. The scheduler functionality is dynamically changed according to different characteristic in order to optimize the overall behaviour. This concept was already showcased in by an early demonstrator (the MWC'16 Demo 1), which focused on the live experience of low latency remote driving applications.
- Multi tenancy: included in Demo 2. Network slicing and, more generally, resource sharing is one of the key features of future 5G Network and will be investigated throughout the development of the demonstrator.
- Security: all the innovative KPIs envisioned by 5G NORMA need an increased security. Many of the concepts developed by 5G NORMA Task 3.3 are going to be demonstrated in Demo 3.
- Economic aspects: one of the goals of the project is to evaluate the economic impact of the proposed innovations. The software developed for Demo 4 will provide automatic ways to access tangible economic metrics.

All planned demonstrators are well linked with the work carried out in different work packages in 5G NORMA. Namely, innovative concepts and investigations developed in the framework of 5G NORMA Work Packages 2, 3, 4, and 5 have a direct link with the demonstrators developed in Work Package 6.

Also, we remark that a substantial part of the software and hardware platform that is being developed for the four demonstrators rely on Open Source. This choice both increases the reproducibility and the sustainability of the demonstrators, but also extends the impact of the 5G NORMA project.

Two functional demos (MWC16-1 & MWC16-2 demos described in section 3) were successfully presented in several international events (most notably Mobile World Congress and EuCNC) and similar venues will be targeted as showcase during the remainder of the project.

7 References

[1] ETSI, “Network Functions Virtualisation, Introductory White Paper”, ETSI Technical Report, Oct. 2012.

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

[3] Opnfv, https://www.opnfv.org

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

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

[6] OpenStack Nova, https://wiki.openstack.org/wiki/Nova

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

[8] OpenDayLight, https://www.opendaylight.org

[9] Open vSwitch, http://openvswitch.org

[10] Tacker, https://wiki.openstack.org/wiki/Tacker

[11] OpenMano, https://github.com/nfvlabs/openmano/blob/master/README.md

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

[13] Open-O, https://www.open-o.org

[14] Gohan, http://gohan.cloudwan.io

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

[16] XEN, https://wiki.xen.org/wiki/Main_Page

[17] MathGL, http://mathgl.sourceforge.net/doc_en/Main.html

[18] Azcom Technology baseband board, http://www.azcom.it/index.php/products/hardware-platforms/small-cell-bbu

[19] Ericsson , “Ultra-Reliable and Low-Latency (urllc) 5G Communication”, EuCNC, 2016

[20] "Simulation of Urban MObility" (SUMO) open-source simulation package: http://sumo.dlr.de/wiki/Main_Page

[21] Ettus, https://www.ettus.com

[22] Nuand, https://www.nuand.com

[23] https://greatscottgadgets.com/hackrf

[24] Juniper, Mobile Edge Computing Use Cases & Deployment https://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000642-en.pdf

[25] Charles Chambers, Ade Ajibulu et al. 5G NORMA Deliverable D2.2. Evaluation methodology for architecture validation, use case business models and services, initial socio-economic results. https://5gnorma.5g-ppp.eu/wp-content/uploads/2016/11/5g_norma_d2-2.pdf.

Annex A. Time plans.

Demo 1.

COMMON TASKS (AZCOM & NOMOR):

Work Plan by Month		2016							2017												
Task	Code	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
General Purpose	D1.1																			
Define general demo objective and constraints	D1.1.1																			
Collect required input from WP4 & WP5 to design software+hardware prototype (defined in task T6.1)	D1.1.2			■	■	■	■	■	■	■										
Definition of experiments that need to be performed (step-by-step demo procedure). From task T6.1	D1.1.4		■	■	■	■	■	■	■	■										
└ Definition of associated KPI-based expected results for the PoC success.	D1.1.5		■	■	■	■	■	■	■	■										
Define innovations provided by this demo	D1.1.6		■																	
Discuss next steps in function of the major milestones (Note 10)	D1.1.7	■																		
To provide two or more terminals for doing different services	D1.1.8																			
Module testing, testbench setup and preliminary results (T6.2)	D1.1.9					■	■	■	■	■										
Coordination for improvement of individual modules and of communication protocol based on testbench results	D1.1.10							■	■	■	■	■	■	■	■	■	■	■	■	

D1.1.1 & D1.1.8 already completed in a previous stage.

AZCOM:

Work Plan by Month		2016							2017												
Task	Code	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Hardware Specific Tasks (Azcom part)	D1.2																			
Design of hardware prototype (from task T6.1)	D1.2.1	■	■	■																
└ Initial design of the demo	D1.2.2	■	■																	
└ Updating of the design of the demo	D1.2.3		■	■	■															
To provide one eNodeB sets responsive to the commands received from SDMC (T6.2)	D1.2.4	■	■	■	■															
Provide an Evolved Packet Core (from task T6.2)	D1.2.5			■	■															
Description of Azcom eNB (BBU based on Texas Instruments Multicore DSP+ARM System-on-Chip) (Note 11)	D1.2.6	■																		
Description of Azcom eNB sw deployment (Note 9)	D1.2.7	■																		
Implementation of Az side of the interface towards the SDMC	D1.2.8				■	■	■													
SDMC support on Az software	D1.2.9					■	■	■												
Testbench Az premises	D1.2.10						■	■												
Integration with Nomor software SMDC	D1.2.11							■	■											
Final use cases testing	D1.2.12								■	■										
Get (purchase) all necessary hardware	D1.2.13	■	■	■	■	■	■	■	■											
Improvement of the intermediate demo	D1.2.14										■	■	■	■	■	■	■	■	■	

NOMOR:

Work Plan by Month		2016							2017											
Task	Code	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Software Specific Tasks (Nomor part)	D1.3																			
Design of software prototype (from task T6.1)	D1.3.1																			
Clarify functional view of Nomor SDMC controller (Note 8)	D1.3.2	■																		
Design communication protocol (Note 1)	D1.3.3	■	■																	
Implement the communication protocol (Note 3)	D1.3.4		■	■																
Implement a basic functional SDMC, i.e. with communication but without decision logic	D1.3.5		■	■	■															
Set up and test communication with a preliminary testbench using hardware eNodeBs and the basic SDMC	D1.3.6				■	■														
Design and implement decision logic for network reconfiguration at SDMC	D1.3.7				■	■	■													
Running simulations with network simulator and SDMC for estimating simulated gains of reconfiguration	D1.3.8				■	■	■													
Constructing a mapping between eNodeB's feedback and simulated gains from reconfiguration	D1.3.9					■	■													
Running tests on the testbench with SDMC, hardware eNodeBs and UEs	D1.3.10						■	■												
Improving communication, decision logic and mapping based on test results	D1.3.11						■	■												
Preparing a pre-release hardware demo for internal review	D1.3.12						■	■	■											
SDMC GUI adaptation to show eNodeB's activity (optional additional feature 1)	D1.3.13						■	■	■	■										
Finalizing the Intermediate Demo	D1.3.14								■	■										
Improvement of SDMC application based on feedback from Intermediate Demo.	D1.3.15										■	■	■	■	■	■	■	■	■	
Research on SDMC's role in 5G networks and possible design changes accordingly.	D1.3.16												■	■	■	■	■	■	■	
Get (purchase) all necessary SW and HW platforms	D1.3.17																			

Note: D1.3.1 and 1.3.17 already completed in an initial stage

Demo 2:

Common Tasks:

Work Plan by Month		2016							2017											
Task	Code	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Demo #2. Service-aware QoE/QoS Control (Atos / UC3M)																				
Common Tasks	D2.1																			
Identification of Virtual Functions to be included in the PoC platform (T6.1)	D2.1.1		■	■	■															
Identification of services to be included in the PoC platform	D2.1.2		■	■	■															
Definition of experiments that need to be performed (step-by-step demo procedure). From task T6.1	D2.1.3		■	■	■															
└ Definition of associated KPI-based expected results for the PoC success.	D2.1.4		■	■	■															
Module testing and preliminary results (T6.2)	D2.1.5													■	■	■	■	■	■	
└ Definition of the virtual containers format	D2.1.6														■	■	■			
└ Definition of the per slice orchestration strategies	D2.1.7											■	■	■	■	■	■			

UC3M:

Work Plan by Month		2016							2017											
Task	Code	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
UCM3 Specific Tasks (Network Slicing)	D2.2																			
Define innovations provided by the UC3M demo	D2.2.1																			
Get (purchase) all necessary HW and SW platforms (Question 2)	D2.2.2																			
Enable Remote Site (antenna site and the lower layer of the 3GPP RAN)	D2.2.3		■	■	■	■														
Study the feasibility of different functional splits	D2.2.4				■	■	■													
Implement the selected functional split	D2.2.5						■	■	■	■	■									
Study sharing techniques for NF	D2.2.6							■	■	■	■									
Implement the selected NF sharing technique	D2.2.7								■	■	■	■	■							
Provide UE terminals for the demo	D2.2.8									■	■	■	■							
Implementation of proposed solution in a small scale testbed & evaluation in conjunction with WP6 (DoW)	D2.2.9												■	■	■	■	■	■	■	

Note: D2.2.1 and D2.2.2 already completed in an initial stage

ATOS:

Work Plan by Month		2016							2017											
Task	Code	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Atos Specific Tasks (Virtual Infrastructure & MANO Implementation Demo)	D2.3																			
1. Analysis & Design	D2.3.1																			
├ Project Initiation (create conceptual design)	D2.3.2																			
├ Identify gap between selected open source products and 5G Norma specific needs	D2.3.4	■	■	■																
├ Identify and select open source projects to be used (ODL, Open vSwitch, OpenStack, TeNOR...)	D2.3.3	■	■	■	■															
└ Definition of demonstration platform characteristics (architecture, topology, HW & SW modules, etc.) - T6.1	D2.3.5	■	■	■	■															
2. Demo platform set-up	D2.3.6																			
├ Acquisition of hardware platforms	D2.3.7																			
├ Initial set-up (create users accounts, configure...)	D2.3.8	■																		
├ Install virtualization environment & OS (XEN)	D2.3.9	■																		
├ Install VIM (OpenStack)	D2.3.10			■	■															
├ Install SDN Controller (OpenDayLight)	D2.3.11				■															
├ Install Switch (Open vSwitch)	D2.3.12				■															
├ Install and configure gateway module	D2.3.13				■															
├ Install & test orchestrator module (TeNOR)	D2.3.14				■	■														
└ Add Physical laptop to emulate Edge Network	D2.3.15																■			
3. Development & Implementation (T6.2)	D2.3.16																			
├ Integrate dummy VNF's in NFV MANO architecture	D2.3.17							■												
├ QoS/QoE monitoring, mapping and control module	D2.3.18								■	■										
└ Improvements in orchestration modules to support VNF's placement	D2.3.19										■	■	■							
4. Integration	D2.3.20																			
├ Integrate UC3M VNF's in NFV MANO architecture	D2.3.21													■	■	■				
└ Integrate UC3M Remote Site (SDR)	D2.3.22																■	■	■	
5. Testing	D2.3.23																			
├ Integration tests phase 0: Verify the integration of the different NFV MANO components (TeNOR, OpenStack, etc.)	D2.3.24				■	■	■													
├ Basic System Testing. Perform test with initial set-up (virtualized edge cloud and dummy VNF)	D2.3.25							■	■											
├ QoS/QoE monitoring, mapping and control module testing	D2.3.26									■	■									
├ Test improvements in orchestration modules to support VNF's placement	D2.3.27											■	■	■						
├ Integration tests phase 1: Deployment and placement of VNF's provided by UC3M	D2.3.28														■	■	■			
├ System test using VNF's provided by UC3M	D2.3.29															■	■			
├ Integration tests phase 2: UC3M Remote site integration	D2.3.30																	■	■	
└ Full system tests with the final demo setup and results (T6.2)	D2.3.31																		■	

Note: D2.3.2 and 2.3.7 already completed in an initial stage

Demo 3:

Work Plan by Month		2016							2017											
Task	Code	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Demo #3. Secure Billable Multi-tenancy through Virtual AAA entities																				
Analysis-1 (diameter protocol, LTE AAA, ETSI NFV Authentication, Geolocation Database)	D3.1																			
Design-1 (Hierarchical Authentication, Multi-level service)	D3.2																			
Development-1 (diameter protocol agent; new protocol for billing)	D3.3	■	■																	
Test-1 (Signalling message exchange)	D3.4	■	■																	
Analysis-2 (Interaction with SDN Controllers, NoSQL and SQL database cluster, DTLS Transport Protocol)	D3.5		■	■	■															
Design-2 (VAAA Data Model, Database cluster, Integration Diameter Agent)	D3.6		■	■	■	■														
Development-2 (VAAA Data Model with DB, Integration Diameter Agent, Interaction with SDN Controller)	D3.7				■	■	■													
Test-2 (DB cluster: Data replication, DB: store, retrieve, update Message exchange between SDN controllers)	D3.8						■	■												
Analysis-3 (Threat Modelling, Access Controls)	D3.9								■											
Desingn-3 (Service Attributes value and data structures, Threat Testing Scenarios)	D3.10								■											
Development-3 (Access Controls,Threat Testing Scenarios)	D3.11								■	■	■									
Test-3 (Various role access server, Various role access VNF services)	D3.12										■	■								
Analysis-4 (Data Visualisation)	D3.13											■								
Design-4 (GUI security log indicator)	D3.14											■	■							
Develop-4 (GUI security log indicator and software refinement)	D3.15												■	■	■			■	■	
Test-4 (Testing scenarios)	D3.16													■	■	■	■	■	■	

Note: D3.1& D3.2 already completed in a previous stage.

Demo 4:

Work Plan by Month		2016							2017											
Task	**Code**	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Demo #4. Online Interactive 5G NORMA Business Cases																				
Phase 1 network cost model	D4.1																			
Final report D2.2 illustrating benefits (cost savings) of the 5G NORMA multi tenancy innovation	D4.2																			
Phase 2 network cost model	D4.3																			
Define innovations provided by the RW tool	D4.4																			
Define detailed requirements for the tool (tasks #RW2 & #RW3)	D4.5																			
Initial design of a web based tool & GUI with graphical and numerical outputs as demonstrator (Note 6)	D4.6																			
Final design of web based tool (pending on task #RW3)	D4.7																			
Release of web based tool with modelling data for 1st evaluation case	D4.8																			
Release of web based tool with modelling data for 2nd & 3rd evaluation cases	D4.9																			
Revised Phase 2 Modelling (and potentially additional geotypes)	D4.10																			
Updated SW release	D4.11																			
Get (purchase) all necessary HW and SW platforms	D4.12																			
Final SW Release	D4.13																			

Note: D4.2 already completed in a previous stage.