Deliverable D4.2 - Intermediate Report on the MANO Platform

Editor:	Diego R. López, TID Iván Vidal, UC3M	
Deliverable nature:	Report (R)	
Dissemination level:	Public (PU)	
Date: planned | actual	31 December 2018	23 January 2019
Version | No. of pages	1.0	59
Keywords:	5G, NFV, MANO, NFVI, service orchestration, NS, VNF, VVF, VxF, Multi-site	

Abstract

This document describes the relevant updates incorporated into the 5GinFIRE MANO platform during the second year of the project. The MANO platform has been used to support the experimentation activities of 5GINFIRE Open Call phase 1, carrying out the management and orchestration of the different network functions and services deployed during this experimentation period. The current 5GINFIRE MANO platform is based on OSM Release FOUR, an open-source implementation of the NFV MANO stack hosted by ETSI and can support multi-site deployments. Finally, the document also includes information regarding functional validation, as well as the ongoing work that is being conducted to provide further enhancements to the platform.

Disclaimer

This document contains material, which is the copyright of certain 5GINFIRE consortium parties, and may not be reproduced or copied without permission.

All 5GINFIRE consortium parties have agreed to full publication of this document.

Neither the 5GINFIRE consortium as a whole, nor a certain part of the 5GINFIRE consortium, warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, accepting no liability for loss or damage suffered by any person using this information.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732497. This publication reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Impressum

Full project title: Evolving FIRE into a 5G-Oriented Experimental Playground for Vertical Industries.

Short project title: 5GINFIRE.

Number and title of work-package: WP4 Core MANO Service Management and Orchestration.

Number and title of task:

- Task 4.3: Orchestration platform evolution and continuous integration.
- Task 4.4: Integration and support of open call MANO services.

Document title: Deliverable D4.2 - Intermediate Report on the MANO Platform.

Editor: Diego R. López, Telefónica Investigación y Desarrollo SA; Iván Vidal, Universidad Carlos III de Madrid.

Work-package leader: Diego R. López, company: Telefónica Investigación y Desarrollo SA

Copyright notice

Executive summary

In the NFV architecture, the MANO framework is in charge of the management and orchestration of the network functions, supporting their lifecycle, and the composition of these functions to build Network Services. Management and orchestration are applied following a set of requirements, usually expressed as policy statements, interpreted and enforced by the MANO framework, commonly referred to as well as the "MANO stack". The 5GINFIRE project selected one of the most advanced open-source MANO platforms currently available, Open Source MANO (OSM) as the base for its MANO framework, and from this selection has actively contributed to its consolidation and evolution, as well as taken advantage of this evolution to provide the highest-level management and orchestration support to the 5GINFIRE experiments (as demonstrated for the first open call).

The 5GINFIRE environment poses a series of challenges for the MANO platform to be used, mainly related to the high diversity of functions to be considered and the multi-domain nature of the infrastructure to be managed. The technical solution adopted by the project considers the utilization of a single orchestration domain, where an NFV orchestrator manages and coordinates the creation of network services on the participating infrastructures. These participating experimental facilities are at the core of the 5GINFIRE MANO deployment, each one with its independent resource management element (VIM). The 5GINFIRE MANO team has defined the procedures for the integration, validation, and monitoring of facilities as they join the multi-domain 5GINFIRE orchestration and management environment.

The WP4 5GINFIRE team has continued addressing the orchestration challenges in the project, updating the original MANO platform, consolidating the validation mechanisms, and enhancing the aspects related to a multi-site and multi-user environment. The original documentation set has been extended, including reference materials for experimenters and experimental facility admins. In other words, the MANO platform has grown, demonstrating its ability to incorporate new experimental sites, and providing a high-value experience for the deployment of NFV-enabled network testbeds.

List of authors

Company	Author
TID	Diego R. López, Juan Rodríguez Martínez
UC3M	Iván Vidal Fernández, Luis Félix González Blázquez
ITAv	Diogo Gomes, Eduardo Sousa, Inês Moreira
UNIVBRIS	Aloizio P. Silva
UFU	Flávio de Oliveira Silva
University of Thessaly (5GVINO)	Nikos Makris, Christos Zarafetas, Alexandros Valantasis, Thanasis Korakis.
Trinity College Dublin (WINS_5G)	Diarmuid Collins, Maicon Kist
PSNC (eHealth5G)	Damian Parniewicz, Bartosz Krakowiak

Table of Contents

Abbreviations

MANO: Management and Orchestration

NFV: Network Function Virtualization

NFVI: NFV Infrastructure

NFVO: NFV Orchestrator

NS: Network Service(s)

SO: Service Orchestrator

RO: Resource Orchestrator

VCA: VNF Configuration and Abstraction

SDN: Software Defined Networks

VIM: Virtual Infrastructure Management

VNF: Virtualized Network Function

VNFM: VNF Manager

VVF: Virtualized Vertical Function

VxF: Virtualized (Network or Vertical) Function

WIM: WAN Infrastructure Manager

1 Introduction

In the NFV architecture, the MANO framework is in charge of the *management and orchestration* of the Virtual Network Functions (VNFs), supporting the lifecycle of such functions, and the composition of these functions to build Network Services (NS).

This lifecycle management includes all events related to the execution of a VNF, since its initial incorporation to the cloud environment where it will run (the so-called *onboarding*) up to an eventual *decommission* of such a function. And naturally encompasses essential events such as particular *instantiations* and *activations*, and those related to the properties of cloud-based functions, like *scaling* events. Given the final goal of NFV is to provide network services, these events at the VNF level have to be coordinated at the NS level as well, requiring the MANO framework to *orchestrate* individual VNF-related events to ensure the proper behaviour of the services under its control.

Management and orchestration are applied following a set of requirements, usually expressed as policy statements within metadata structures (the *descriptors*) that define the intended properties a service and its component functions have to comply with. The MANO framework reads and processes these descriptors, managing the identified components, and interpreting how to satisfy the policy statements that define their intended behaviour. The 5GINFIRE approach implies a common NFV orchestrator, interacting at its *northbound* interface with the portal, and controlling a set of *Virtualized Infrastructure Managers* (VIMs), one at each participating site [2].

The 5GINFIRE project selected one of the most advanced open-source MANO platforms currently available, Open Source MANO (OSM) [1][4] due to: its high degree of maturity, its proven support for a distributed and diverse infrastructure in different administrative domains, the cloud-native mechanisms for function control and management, and the ability to apply *continuous integration* techniques to the platform. And, obviously, the open-source nature that guaranteed an open evolution roadmap that could incorporate in the mainstream the solutions provided by 5GINFIRE to address requirements not yet considered by the platform.

This document describes how the 5GINFIRE MANO platform has evolved from its initial deployment, by:

- Updating the platform itself as the OSM software evolved.
- Addressing the requirements of experimenters as they used the platform.
- Improving the procedures to include new experimental facilities and monitoring their integration into the framework.
- Incorporating new software elements to address the above requirements (from both experimenters and facilities), that were further contributed to the OSM community.

This document follows a structure similar to its predecessor [1], with Section 2 providing an overview of the architecture principles, and any relevant update made to them. Section 3 going into details on the main characteristics of the management stack at each testbed infrastructure, together with a description of their interconnection with the common orchestrator, and an update of how connectivity is provided among functions and experimenters in the distributed 5GINFIRE infrastructure. Section 4 discusses the enhanced functional validation of the MANO platform, including how they are evolving into supporting infrastructure monitoring and supporting telemetry mechanisms. Finally, Section 5 provides a survey of the future MANO platform enhancements, that will not only be included into the 5GINFIRE framework but also contributed to the OSM community to make them part of the OSM code base for future releases.

2 Design of the 5GinFIRE MANO platform

The 5GINFIRE ecosystem brought a series of particular challenges for the MANO platform to be used, especially related to the high diversity of functions to be considered and the multi-domain nature of the infrastructure to be managed. These requirements, and the general methodology for addressing them were presented in Deliverable D4.1 [1]. During this span, 5GinFIRE partners have confirmed those requirements, with the only considerations of even a greater importance of multi-site capabilities and the inclusion of new verticals, like IoT, eHealth, and so on.

The technical solution adopted by the project [2] [3] considers the utilization of a single orchestration domain, where an NFV orchestrator, implemented with Open Source MANO (OSM) [4], manages and coordinates the creation of Network Services (NS). We can define a Network Service as a composition of Virtualized Vertical/Network Functions, or VxFs. Each of these VxFs may be in turn deployed at any of the experimental infrastructures made available by 5GinFIRE partners (hereafter referred to as infrastructure providers).
Deliverable D4.1 described the deployment of a MANO platform encompassing three experimental infrastructures. The current architecture design of the 5GINFIRE MANO platform, including seven infrastructure providers, is depicted in Figure 1.

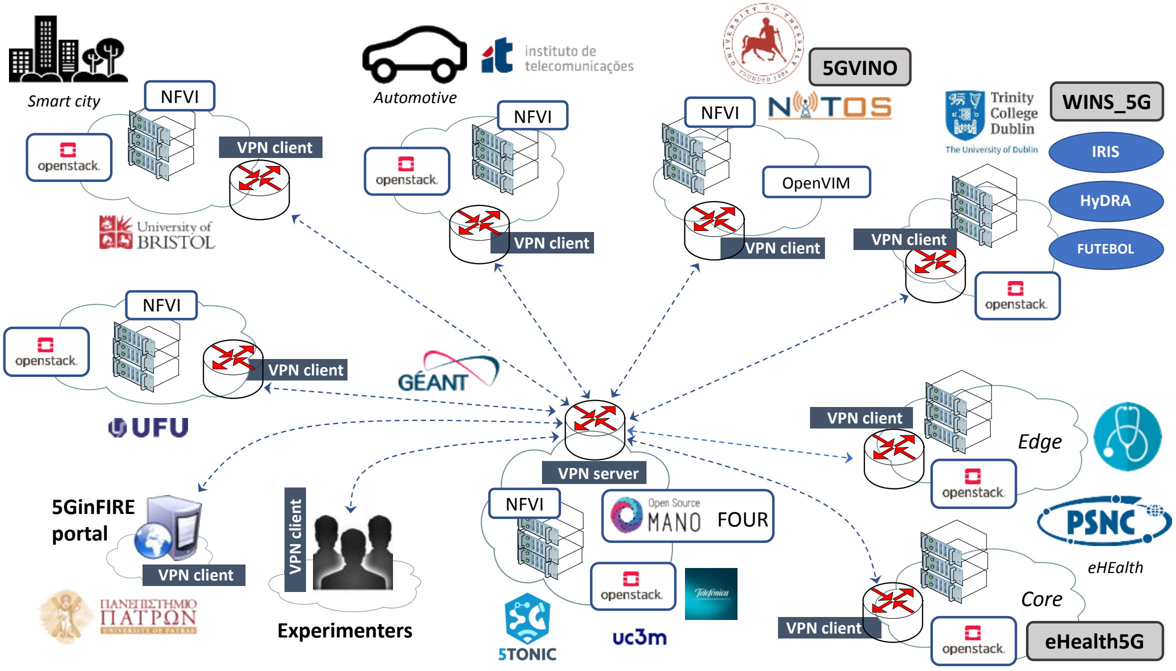


Figure 1: Overview of the 5GiFIRE MANO platform

The MANO platform supports multi-site experimentation activities across different vertical domains:

1) An infrastructure at the global 5G Telefonica Open Innovation Laboratory (5TONIC) [5], made available by TID and UC3M, to support experimentation with NFV functions and services.
2) An experimentation facility located at ITAv, providing access to an automotive testbed in the city of Aveiro (Portugal).
3) An infrastructure made available by UNIVBRIS, supporting experimentation activities over a smart city environment in the city of Bristol (UK).

4) An experimentation facility at UFU, located at Uberlândia (Brazil), enabling trialling with 5G applications with a particular consideration on the edge network resources.
5) An infrastructure provided by the NITOS testbed, i.e., 5GVINO, hosted by the University of Thessaly (Greece), which provides access to programmable resources for wireless networking, SDN and cloud computing facilities.
6) An eHealth experimental vertical facility, eHealth5G, hosted by the Poznan Supercomputing and Networking Center (Poland); this facility supports experimentation in the area of telemedicine and eHealth, offering access to: realistic eHealth equipment; a small Edge Cloud, close to eHealth devices; and a core cloud accessible via MPLS/Optical service provider network.
7) A reconfigurable radio testbed at Trinity College Dublin (Ireland), Iris, supporting radio hardware, cloud-RAN, NFV, and SDN technologies. This testbed has been extended and made available for experimentation activities in 5GinFIRE. We refer to this testbed extension as WINS_5G.

Each partner running an experimental infrastructure is in charge of the deployment and maintenance of a Virtualized Infrastructure Manager (VIM), compliant with the OSM software stack. On top of that, the NFV orchestrator of 5GinFIRE, deployed at 5TONIC, interacts with the VIMs of the testbed providers involved in a service deployment: it coordinates the allocation and setup of the computing, storage and network resources which are necessary for the instantiation and interconnection of the VxFs that compose the network service.

Additional sites with heterogeneous infrastructure and equipment, such as those that have come (and will come) from the Open Call process of 5GinFIRE, can be flexibly incorporated as needed, as long as they support a compliant VIM [6] and they set up the inter-site connection mechanisms defined in [1] and updated in section 3.2.

2.1 Structure of the orchestration service

Figure 2 presents an overview of the architectural design of the OSM software stack, as specified for Release FOUR [6]. This stack encompasses different modules that provide the orchestration functionality through their interoperation: a Northbound Interface (**NBI**), a Life Cycle Management (**LCM**) component, a Resource Orchestrator (**RO**), a Network to VNF Configuration (**N2VC**) module, a VNF Configuration and Abstraction (**VCA**) component, and a Monitoring (**MON**) module. Also, a unified message bus (provided by Apache Kafka) enables the asynchronous communication among the modules.

The NBI provides a point of contact for external entities (e.g., the 5GinFIRE portal or the Client interface of OSM) to interact with the OSM system, and it is aligned with the ETSI NFV specification SOL005 [7]. It includes the OSM Information Model (IM), which enables the authoritative activity of the different framework components over the data models of the OSM system (i.e., the NS and VNF descriptors).

The LCM supports the lifecycle management of NS potentially composed of multiple VxFs, coordinating their creation and deletion. For this purpose, the LCM interfaces with the RO and the VCA modules of the OSM architecture. Additionally, the LCM provides other essential enabling functionalities, such as the management of NS/VNF descriptors and packages.

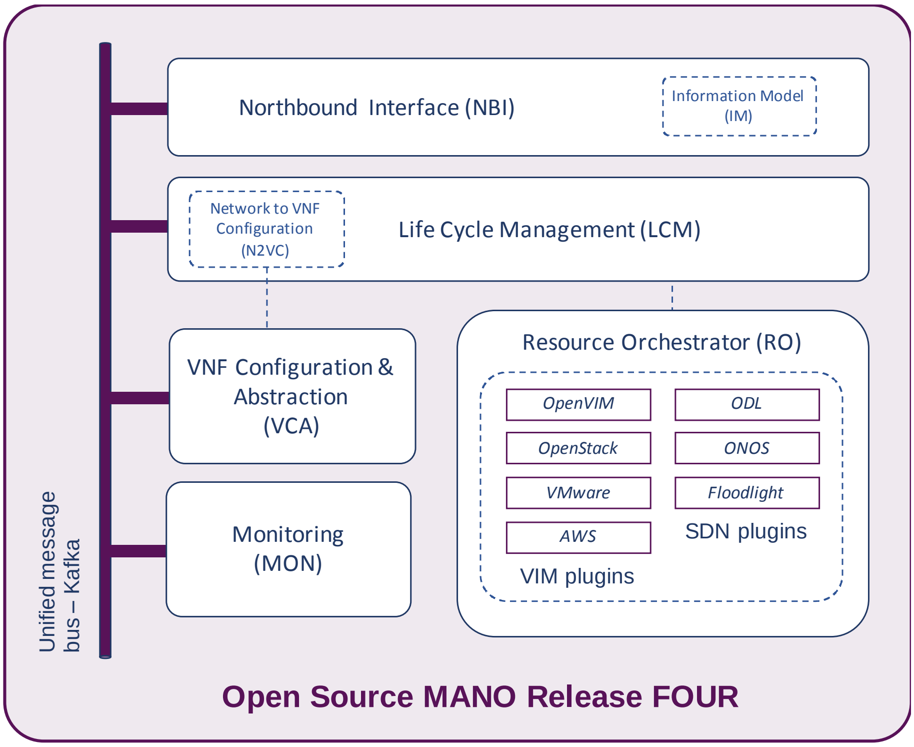

Northbound Interface (NBI)
Information Model (IM)
Network to VNF Configuration (N2VC)
Life Cycle Management (LCM)
VNF Configuration & Abstraction (VCA)
Resource Orchestrator (RO)
OpenVIM
ODL
OpenStack
ONOS
VMware
Floodlight
AWS
SDN plugins
VIM plugins
Monitoring (MON)
Unified message bus – Kafka
Open Source MANO Release FOUR

Figure 2: Structure of the OSM stack

The RO module coordinates the allocation and configuration of computing, storage and network resources under the control of the different VIMs and SDN controllers, to support the execution and interconnection of VxFs. OSM Release FOUR supports multiple types of VIMs through a plugin model, including OpenVIM, OpenStack, VMWare vCloud Director, and Amazon Web Services Elastic Compute Cloud. Additionally, this plugin model enables the RO to directly manage some SDN controllers, including OpenDaylight, Floodlight, and ONOS.

The N2VC provides the framework that enables the communications between the LCM and the VCA modules. The latter is aligned with the VNF Manager defined by the ETSI NFV reference architectural framework [8], supporting day-1 and day-2 configuration of VNFs. With this purpose, the VCA has an interface to *Juju*, which allows configuring VNFs through the execution of Juju charms[1] that can be specified within VNF packages.

The MON module provides the mechanisms to leverage external monitoring tools (such as OpenStack Aodh or OpenStack Gnocchi) and to steer the monitoring information to the unified message bus. This information (e.g., metrics or alarm events that occurred during the execution of a network service), can be consumed by fault and performance management solutions, and trigger the orchestration updates on the VNFs that are executed in the provisioning of a network service. In OSM Release FOUR, metrics and alarms include the average utilization of memory, the packets sent, or the CPU utilization, to cite a few of them.

2.2 Methodology for the MANO platform evolution

As described by Deliverable D4.1, WP4 has followed an iterative approach where components have been deployed after extensive testing and validation. Testing and

[1] Creating your own VNF charm (Release FOUR), OSM Wiki (last access on December 2018): https://osm.etsi.org/wikipub/index.php/Creating_your_own_VNF_charm_(Release_FOUR).

validation are mostly concerned with functional and integration aspects as it is necessary to assure the continuous functioning of the 5GinFIRE distributed testbed in which each site has its particularities.

While D4.1 mostly dealt with the installation of the distributed testbed, this document reports the procedures established for the operation, expansion (through open-calls) and foreseen upgrades (mainly OSM related). While 5GinFIRE experimenters expect the testbed to be stable, they also expect it to be updated with the most relevant software tools required for their experiments.

To achieve these objectives, WP4 continued the approach of having online meetings every two weeks to discuss all matters relevant to the operation, ongoing efforts and upgrade plans. All information is shared with the other project WPs, and ad-hoc cross-WP meetings have been called to discuss inter-WP issues, namely with WP3 and WP6.

Between these meetings, online messaging tools (such as Skype and Slack) have assisted partners in the quick resolution of issues triggered by experiments such as new requirements and punctual un-availabilities of the platform.

During the period, the production-level orchestration at 5GinFIRE has been based on OSM Release TWO, case being that in parallel OSM Release FOUR has been tested, and contributions identified in the scope of WP4 have been pushed towards the OSM community for inclusion in future releases. This aspect is very important as the project previously defined that “A modification to the 5GinFIRE MANO platform should not prevent or affect the correct behaviour of the orchestration service for scenarios that do not require the modification” as was the case with experiments of the first open call in which the project announced the availability of OSM Release TWO.

At the time of writing of this deliverable, the second open call experiments will be presented with OSM Release FOUR, which already includes some improvements/requirements identified in the project.

Among such improvements (further detailed in future chapters of this document) we identify the following as the most relevant:

- The configuration of VNFs via Ansible playbooks
- Integration of Keystone into the MANO platform (to be included in Release FIVE)
- SDN and WIM integration
- Support of monitoring framework based on OpenStack telemetry modules

3 Implementation of the 5GINFIRE MANO platform

3.1 Description of the NFV experimental infrastructures

The 5GinFIRE MANO platform manages and coordinates the creation of network services over the 5GinFIRE distributed infrastructure, as a composition of interconnected VxFs that can be deployed and operate at different sites.

In the following sections, we provide an overview of the MANO components and the NFV infrastructure available at each of the 5GinFIRE sites. With the purpose to serve as an updated reference, this section provides:

1. Up-to-date information on the distributed NFV infrastructure that was set up during the first year of the project lifetime, as specified in [1].
2. Report on the NFV infrastructure available at the experimental sites that have been integrated into 5GinFIRE, once the first phase of the first Open Call of the project has been completed.

3.1.1 5TONIC site

The global 5G Telefonica Open Network Innovation Centre (5TONIC) [5] has been established in Madrid (Spain) as a leading European hub for knowledge sharing and industry collaboration in the area of 5G technologies. The laboratory provides an open research and innovation ecosystem for industry and academia that promotes joint project development, joint entrepreneurial ventures, discussion fora, and a site for events and conferences, all in an international environment of the highest impact. 5TONIC also serves to evaluate and demonstrate the capabilities and interoperation of pre-commercial 5G equipment, services and applications. Currently, the 5TONIC laboratory has ten members: Telefonica, Institute IMDEA Networks, Ericsson, Intel, Commscope, Universidad Carlos III de Madrid, Cohere Technologies, InterDigital, Altran, and RedHat.

The 5TONIC laboratory, as a multipurpose environment, counts with multiple racks, which may be flexibly interconnected according to any experimentation requirements, along with a common infrastructure to aid experimentation, trials, and demonstrations with 5G products and services. In particular, secure access to external sites can be provided via VPN gateways, allowing different solutions to support management, control and data operations from remote network locations, depending on specific requirements. In the following, we describe the main infrastructure and equipment that is available at 5TONIC to support experimentation activities in the context of 5GinFIRE. A schematized overview of this infrastructure is shown in Figure 3.

Regarding the orchestration service, the 5GINFIRE MANO platform provides a dual orchestration software stack, supporting Release TWO [9] and FOUR [6] of the OSM software. This enables the coexistence of network services, and VxFs developed under both versions of OSM (due to different design criteria, Releases FOUR and TWO of OSM are not fully compatible), as well as the re-creation of experiments carried out during the second year of 5GinFIRE (when only Release TWO was available). Following this dual stack approach, two installations of OSM (one for Release TWO and another one for Release FOUR) run in independent virtual machines on a server computer with 16 cores, 128 GB RAM, 2 TB NLSAS hard drive and a network card with 4 GbE ports and DPDK support.

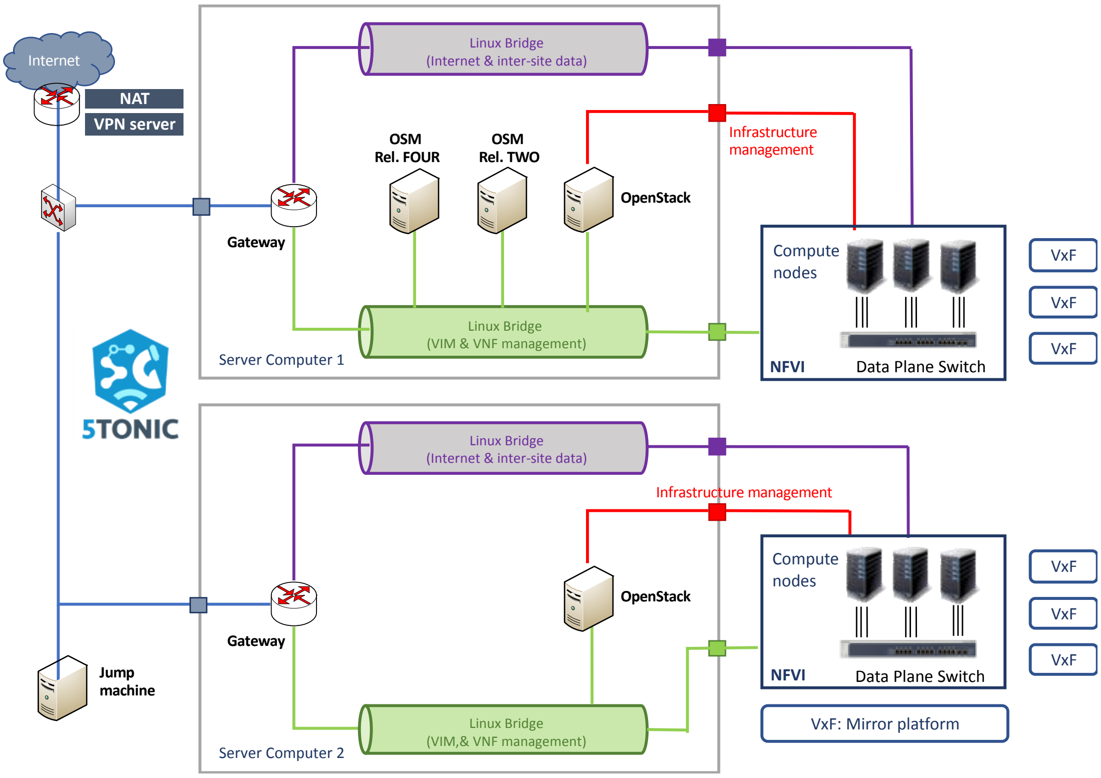


Figure 3: overview of the 5TONIC site

The same server computer hosts a VIM instance based on OpenStack Ocata [10], which allows allocating experiments to an NFV infrastructure composed by two high-profile servers, each equipped with eight cores in a NUMA architecture, 128GB RDIMM RAM, 4TB SAS and eight 10Gbps Ethernet optical transceivers with SR-IOV capabilities. These servers are currently interconnected in the data plane by a 24-port 10Gbps Ethernet switch. This NFVI forms part of the infrastructure of the IMDEA Networks Institute at 5TONIC, hence its utilization is coordinated and shared with other projects and demonstration activities.

An additional and independent instance of OpenStack Ocata is deployed in a separate server computer with six cores, 32GB of memory, 2TB NLSAS, and a network card with four GbE ports and DPDK support. This VIM controls a dedicated NFVI that is solely allocated to experimentation activities within 5GINFIRE. This infrastructure consists of a set of three server computers, each with the same hardware characteristics as the server computer hosting the VIM. These servers are interconnected by a GbE data-plane switch.

In both VIM instances, the OpenStack networking service was installed to support layer-3 services, and the ML2 plug-in of OpenStack was configured to use Linux bridges. The terms and conditions that govern the utilization the aforementioned NFV infrastructures are detailed in the 5GinFIRE website [11].

Finally, the experimentation infrastructure offered to 5GinFIRE includes some server computers (not shown in the figure) to support complementary functionalities, e.g., hosting client applications, deploying a network management system, and performing access-control functionalities. Regarding the latter, a physical jump machine serves to control the access of the 5GinFIRE experimenters to the NFV infrastructures. The security aspects of the 5GinFIRE MANO platform are covered in detail in Section 3.2.

3.1.2 ITAv site

The ITAv site provides an infrastructure based on Openstack Ocata. This infrastructure is connected to the 5GINFIRE OSM-based orchestrator (see Figure 4). Two servers compose the NFVI in ITAv:

1) Orphic
   - Cores: 24
   - Memory: 192 GB
   - Network: 4 x 1Gbps interfaces (supports passthrough, DPDK and SR-IOV)
   - Storage: 2 x 1TB SAS3 drives
2) Aeolus
   - Cores: 16
   - Memory: 256 GB
   - Network: 4 x 1Gbps interfaces (supports passthrough)
   - Storage: 2 x 1TB SAS2

ITAv's Openstack deployment has one controller node where all the services are installed and then there the components mentioned above, where only the compute service is installed.

ITAv's Openstack deployment has three VLAN networks, which are: Control, Data and Management. The Control VLAN is for the control plane packets, while the Data VLAN is used for the data plane packets. The Management VLAN is used to access the nodes in the Openstack deployment, and it is used by the Openstack services to communicate with each other, thus not being visible in the other VLANs.

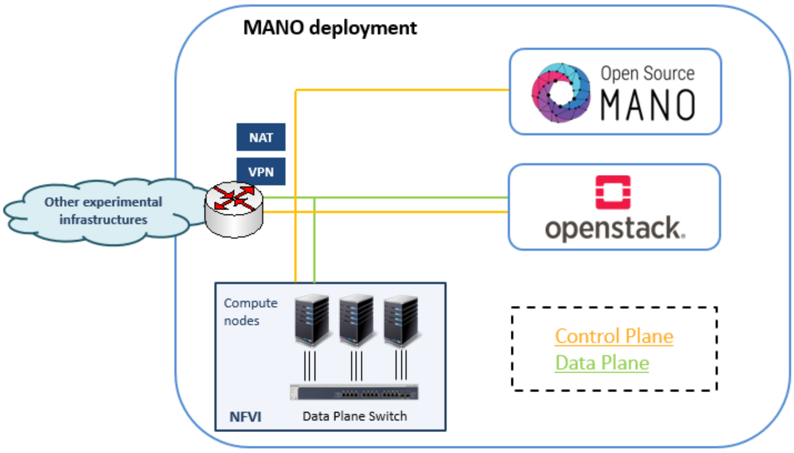


Figure 4: Deployment configuration at the ITAv site

Networks inside the Openstack deployment are handled using Neutron and Linux Bridges. Projects internal networks are created using VXLAN, and the external ones use VLAN networks.

This infrastructure is connected to a 24-port 1Gbps switch. The switch supports the control and data plane, which are separated using VLANs. Each of these VLAN networks can be extended in order to support and interconnect more devices (using layer 2 or layer 3 technologies).

At the ITAv site, there is also an OSM Release FOUR deployed, intended to be used for local deployments and for testing purposes. From the Data Plane, it is possible to reach through routing a network segment where Road Units (OBU, RSU) of WP5 are placed.

3.1.3 UNIVBRIS site

The overall 5GinFIRE UNIVBRIS site is based on the Network Function Virtualization Infrastructure (NFVI) foundation specified by ETSI NFV reference architecture [8] focusing on Smart City Vertical EVI. The UNIVBRIS site is composed of three main building blocks: compute, storage and network. All the virtualized resources being provided by the UNIVBRIS site are available through the OSM-based common orchestrator located at the 5TONIC site. In particular, the UNIVBRIS site's purpose is to provide an excellent testing platform for heterogeneous experimentations and at the same time guarantee computational resources or/and slicing for hosting, deploying, instantiating and supporting VxF's life cycle enabling to conduct rigorous, transparent and replicable testing of NFV ecosystem.

Figure 5 shows the high-level logical UNIVBRIS network topology. ENB-B is the eNodeB one of the building blocks of LTE network. There are two eNodeBs:

1. One is connected to the 5G BOX that is known as PNF UGW. The PNF UGW contains the core element of the LTE network (EPC), which is SDN enabled and provide WiFi and LTE connectivity for User Equipment (UE).
2. The other is connected to the KVM machine that contains the VNF UGW.

The overall network architecture is composed of two compute nodes, one storage node and one controller node based on OpenStack Pike. All nodes operate on top of Ubuntu 16.04 OS. This network is connecting via VPN Layer 3 to 5TONIC enabling multi-site interconnection. In addition, the datacentre is connected via 10km fiber to Millennium Square enabling outdoor experimentation.

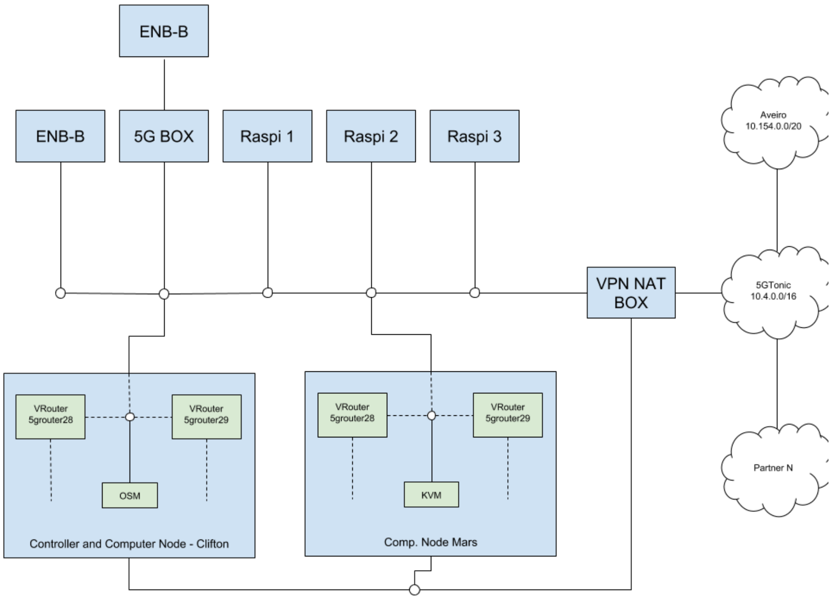


Figure 5: UNIVBRIS logical network topology

Figure 6 shows Millennium Square located at Bristol city centre with all the access technologies available.

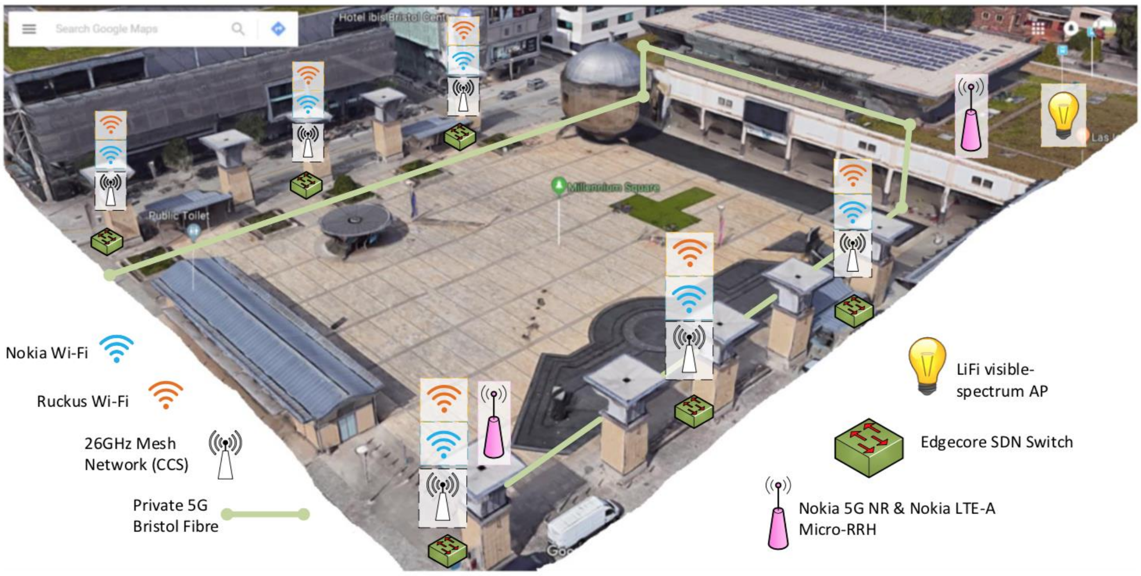


Figure 6: UNIVBRIS outdoor testbed and the available access technologies

A summary of the testbed constituent equipment and capabilities is:

1) Multi-vendor software-defined networking (SDN) enabled packet switched network:
   - Corsa switch (Corsa DP2100)
   - Edgecore switch (Edgecore AS4610 series & AS5712-54X)
2) SDN enabled optical (Fibre) switched network:
   - Polatis Series 6000 Optical Circuit Switch
3) Multi-vendor Wi-Fi:
   - SDN enabled Ruckus Wi-Fi (T710 and R720)
   - Nokia Wi-Fi (AC400)
4) Nokia 4G and 5G NR:
   - 4G EPC & LTE-A (Dual FDD licensed bands for 1800MHz and 2600MHz; with 15MHz of T&D licence in 2600MHz band)
   - 5G Core & 5G NR Massive MIMO (TDD band 42 at 3.5GHz; with 20MHz T&D licence)
5) Self-organising multipoint-to-multipoint wireless mesh network:
   - CCS MetNet a 26GHz with 112MHz T&D licence providing 1.2Gbps throughput
6) LiFi Access point:
   - pureLiFi LiFi access points supporting 43Mbps
7) Advanced fibre optics FPGA convergence of all network technologies enabling considerable flexibility, scalability and programmability of the front/back-haul, to provide experimentation with:
   - Elastic Bandwidth-Variable Transponders
   - Programmable Optical White-box
   - Bandwidth-Variable Wavelength Selective Switches (BV-WSS)

3.1.4 UFU site

The UFU Future Internet Testbed is in the Federal University of Uberlândia (UFU) at the Campus Santa Mônica in Uberlândia, MG, Brazil. Besides the 5GinFIRE infrastructure, deployed in the same location there are other resources from other testbeds from previous and current research projects where UFU has participated, such as FIBRE island (https://fibre.org.br/, formerly http://fibre-ict.eu), FIWARE (www.fiware.org), NECOS (http://h2020-necos.eu). The infrastructure at UFU connects physically with Algar Telecom, a regional telecom operator headquartered in Uberlândia, MG, Brazil, as presented in Figure 7.

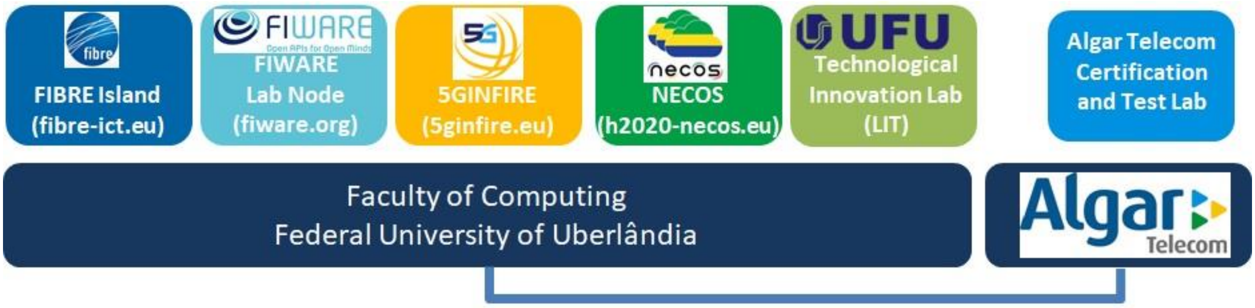

FIBRE Island (fibre-ict.eu)
FIWARE Lab Node (fiware.org)
5GINFIRE (5ginfire.eu)
NECOS (h2020-necos.eu)
UFU Technological Innovation Lab (LIT)
Algar Telecom Certification and Test Lab
Faculty of Computing Federal University of Uberlândia
Algar Telecom

Figure 7: UFU Future Internet Testbed

These different resources may enable different scenarios, such as the integration between 5GinFIRE and the FIBRE island, a FIRE-based testbed. This integration will be investigated after the final deployment of the new FIBRE control framework based on OMF6.

Concerning the 5GinFIRE infrastructure already deployed at UFU, presented in Figure 8, it includes three server computers. The compute nodes have a total of 56 cores, 160 GBytes of RAM and 4 TBytes of storage. Each server has four NICs with 1 Gbps ports and SR-IOV capabilities. A TOR switch, with 48x1GBps ports, interconnects the compute nodes. OpenStack Pike release is the local VIM. An OSM R4 is deployed for local testing purposes.

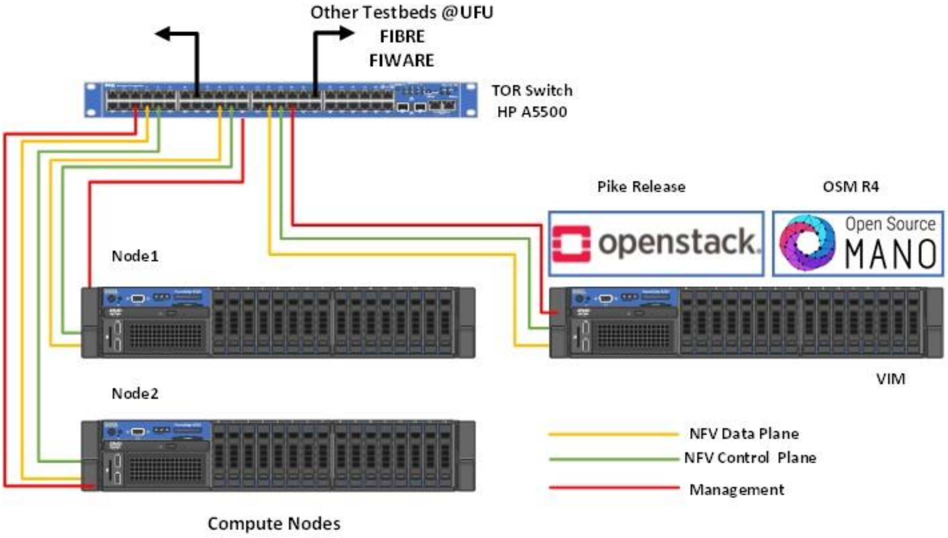


Figure 8 - 5GINFIRE Infrastructure at UFU

At UFU it is possible to run different 5G related applications on different verticals. The resources available at the edge (i.e., at UFU) allow having a primary focus the Smart City vertical by integrating applications and 5G oriented networks.

3.1.5 eHealth5G site

The eHealth5G facility (located in Poznan Supercomputing and Networking Center, Poznan, Poland) is a new testbed created thanks to accepted 5GinFIRE open call infrastructure project and is available for experimenters from December 2018. The eHealth5G facility extends the current 5GinFIRE architecture with a new eHealth Experimental Vertical Instance (eHealth EVI) providing 5GinFIRE experimenters with the possibility of performing experiments in the area of eHealth and telemedicine in a remotely accessible testbed designed for testing technical and usability aspects of services running on top of 5G NFV infrastructure composed of small Edge Cloud, being very closed to eHealth devices, and Core Cloud accessible via MPLS/Optical Service Provider network. The eHealth Vertical Industry infrastructure located in PSNC consists of cutting-edge eHealth equipment enabling eHealth cloud applications, products or services implementation and testing for hospitals, clinics, medical universities, medical or sports professionals.

The eHealth5G facility provides two independent OpenStack-based NFV infrastructures:

- Core Cloud with two Compute Nodes
- Edge Cloud composed of one Compute Node

Both Clouds's OpenStack instances act as VIMs and are managed by 5TONIC OSM. Both OpenStack instances use ML2 plugin for Openvswitch for managing network within Linux system. The Core Cloud OpenStack software stack is deployed on a IBM server computer with 24 cores, 48GB RAM, 1.2TB HD and four 1GbE interfaces. The Edge Cloud OpenStack software stack is deployed on a HP server computer with 40 cores, 160GB RAM, 2.4TB HD and 4x1GbE and 2x10GbE. The HP server follows NUMA architecture, allows process pinning and is DPDK compatible. The server computers where OpenStack is deployed are also used as Compute Nodes. Additionally, two more IBM compute servers are used - the first one acts as second Compute Node in the Core Cloud and the second is used to deploy VMs with additional software for managing network equipment. Unfortunately, none of those server computers supports SR-IOV.

Compute Nodes are connected to statically configured switches. The communication Service Provider network is also statically configured and provides L3 routing services for interconnecting EVI infrastructure, Edge Cloud and Core Cloud. The EVI infrastructure with eHealth equipment is available in two locations in Poznan city (first in the main PSNC premise, called CBPIO and located on Jana Pawla II Street, and second in a building on Zwierzyniecka Street). Access to both locations is available with the usage of Poznan metropolitan network called POZMAN.

External access to the facility as well as communication with other 5GinFIRE facilities has been established using two VPN gateways. First OpenVPN gateway provides access to eHealth5G Core Cloud, whereas the second OpenVPN gateway provides access to the eHealth5G Edge and Core Cloud and eHealth EVI. The eHealth5G infrastructure connectivity is illustrated in Figure 9.

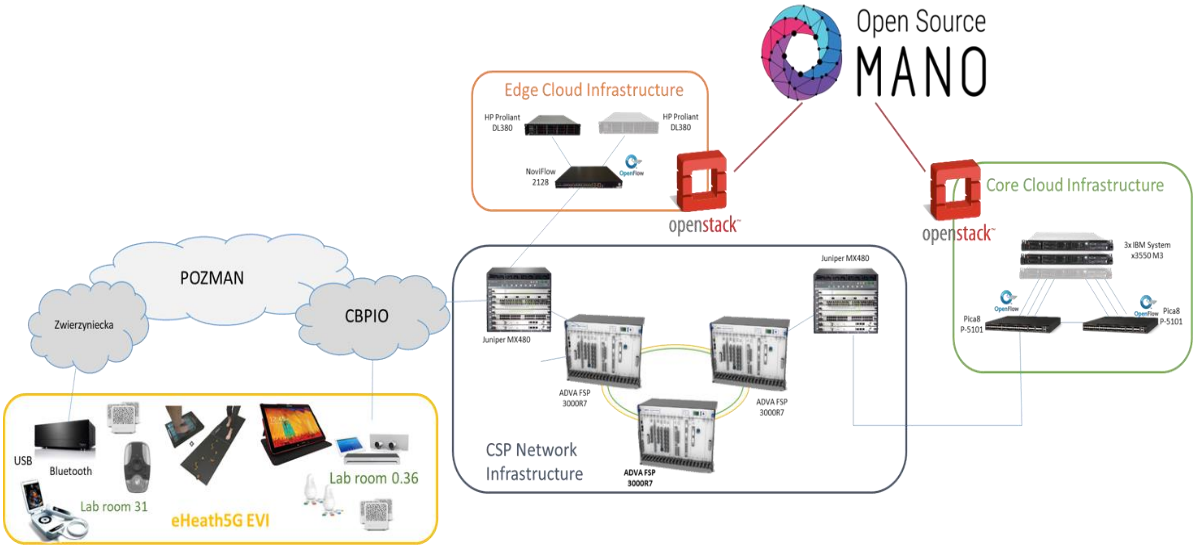


Figure 9: eHealth5G infrastructure connectivity

The NFV infrastructure hardware details are summarized next:

- 1x HP ProLiant DL380 Gen 9 compatible with Intel DPDK (no SR-IOV supported)
- 3x IBM System x3550 M3 (no SR-IOV supported)
- 2x Pica8 P-5101 with OpenFlow 1.3/1.4 (currently configured statically)
- 1x NoviSwitch 2128 (currently configured statically) ‑ Carrier-grade OpenFlow 1.3 switch based on EZchip NP-5 with experimental extensions (DPI, metadata injection, VXLAN, security)
- 2x Juniper MX480 (universal service provider edge router) offering IP routing/Ethernet switching, MPLS, L2/L3 VPNs (VPLS, EVPN, MPLSoGRE, VXLAN) equipped with MS-DPC cards for advanced network traffic processing and analysing (e.g.: traffic sampling, packet inspection)
- 2x Adva Optical FSP 3000R7 equipped with high-speed multimedia SDI cards (10TCC-PCN-3GSDI+10G) allowing for multiplexing and real-time transport of digital SD and HD video content in native optical OTN format (technology essential for support any high-resolution video streams like UHDTV 4k/8k, requiring up to 50Gbps bitrate or for any video 3D technology which is to be used in modern telemedicine solutions)

3.1.6 WINS_5G site

WINS_5G - the reconfigurable radio testbed at Trinity College Dublin provides virtualized radio hardware, software virtualisation, Cloud-RAN, Network Functions Virtualisation (NFV), and Software Defined Networking technologies to support the experimental investigation of the interplay between 5G radio and future networks.

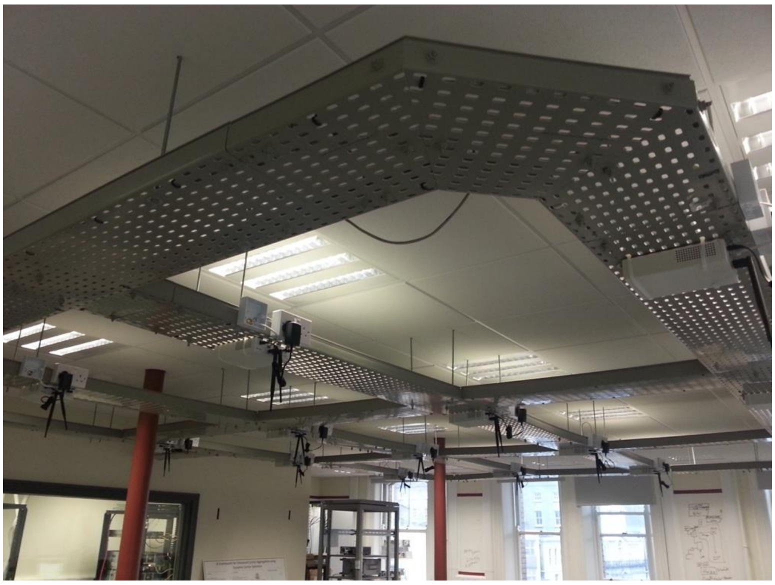


Figure 10: Ceiling mounted ETTUS N210 USRPs at the WINS_5G Testbed

WINS_5G pairs underlying flexible radio and computations resources with various hypervisors in the form of software radio frameworks, virtualized network functions (VNFs), and network service and slicing capabilities to realize various research and testing configurations. These platforms are connected to a private computational cloud, (also called the Virtualized Infrastructure Manager, or VIM) supported by OpenStack Rocky, allowing experimenters to deploy an array of computational environments. To expose the functionality of these platforms for different applications, we employ a variety of radio hypervisors that freely enable prototyping of wireless systems, as exemplified by GNURadio and the HyDRA radio slicing framework. These radio hypervisors combined with dynamic distributed network functions, enable the realization of heterogeneous radio platforms that can support malleable and adaptable networks. WINS_5G is ideally equipped to investigate the combination of network slicing, virtualisation functions, and physical layer approaches into coexisting and coherent next-generation commercial networks, including, but not limited to, 5G. Figure 10, illustrates the ceiling mounted N210 USRP grid at WINS_5G, at the CONNECT Centre in Trinity College Dublin.

A high-level overview of WINS_5G testbed architecture is depicted in Figure 11. The *physical layer*, at the bottom, represent the tangible resources including servers, switches, USRPs, and so forth. The *virtualisation and control layers* in the middle are supported by virtualisation technologies such as OpenStack to support cloud computing, OpenFlow to orchestrate and manage the USRPs, and HyDRA to virtualise the radio. The open source HyDRA radio hypervisor, licensed under the GPLv3 and developed by Trinity College Dublin researchers [12], offers USRPs virtualisation capabilities, enabling ETTUS N210 USRPs to support multiple 5G verticals simultaneously. HyDRA is responsible for allocating physical radio resources to EVIs based on requests received from the EVI manager, transform them into virtualized resources, and assigning these virtual resources to a virtual RF front-end. Virtual RF front-ends (vRF front-ends) operate as if they were interfacing directly with a standard SDR RF front-end by sending/receiving digitized IQ samples. HyDRA ensures isolation while allowing each EVI to adopt its own PHY and MAC technology, and configure its central frequency, bandwidth, and sampling rate on-the-fly. The *vertical* and NFV MANO layers are supported by the Open Source Network Function Virtualization (NFV)

Management and Orchestration (MANO) (OSM) software stack. These elements interact with the physical and virtualisation layers to dynamically instantiate radio EVIs at the WINS_5G testbed. The WINS_5G production system is connected to the 5TONIC OSM server via VPN, which orchestrates and controls EVI instantiation.

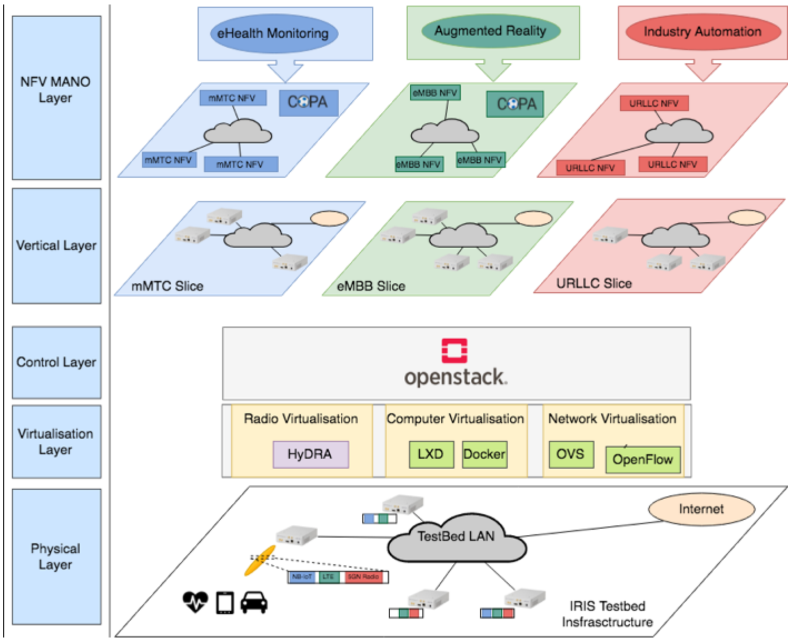


Figure 11: WINS_5G Overview

A summary of the WINS_5G testbed equipment and software capabilities include:

- 1 x Dell PowerEdge R440 server (acting as an OpenStack Controller and Compute node)
- Ubuntu 18.04 server
- 4 x USRP N210s with SBX daughter boards supporting frequencies between 400 MHz and 4.4 GHz.
- USRPS capable of between 5 and 10 MHz of Bandwidth
- 8 x SMA Direct Connect Wi-Fi 2.4GHZ/5GHZ Antennas.
- 1 x 10GB Ethernet Controller (SR-IOV) to N210 USRPs on Dell S4048T-ON switch.
- OpenStack (Rocky) with KeyStone, Glance, Nova, Neutron (type: self-service networks), and Horizon.
- 5 x ports on a Dell Networking S4048T-ON SDN switch.
- Open Network Operating System (ONOS), orchestrating USRPs
- HyDRA radio virtualization technology

3.1.7 5G-VINO site

5G-VINO builds upon the existing efforts of the University of Thessaly (UTH) in deploying and operating cutting-edge infrastructure. UTH is operating since 2007 the Network Implementation Testbed using Open Source platforms (NITOS), which has evolved over the years to a compact solution for evaluating bleeding-edge ideas on the forefront of networking related research. The NITOS testbed is one of the largest single-site open experimental facilities in Europe, allowing users from around the globe to take advantage of highly programmable equipment. The testbed is an integral part of larger federations of resources, such as OneLab [13] and Fed4FIRE [14], enabling experiments with more heterogeneous resources. NITOS has an established user base of over 4000 users in the past years, with over 20 researchers using the infrastructure on a daily basis. In short, the current offering of the testbed is the following:

- Over 100 nodes equipped with IEEE 802.11 a/b/g/e/n/ac compatible equipment, and using open source drivers. The nodes are compatible also with the IEEE 802.11s [15] protocol for the creation of wireless mesh networks. The nodes feature multiple wireless interfaces, and are high-end computers, with quad-core Intel Core i5 and Core i7 processing capabilities, 4/8 GBs of RAM and SSD disks.
- Commercial off-the-shelf (COTS) LTE testbed, consisting of a highly programmable LTE macrocell, multiple femtocells, an experimenter configurable EPC network and multiple User Equipment (UE), such as USB dongles and Android Smartphones [16].
- Open Source LTE equipment, running over commodity Software Defined Radio (SDR) equipment, by the adoption of the OpenAirInterface (www.openairinterface.org) platform [17]. The platform is allowing multiple configurations for creating highly customizable beyond 4G networks.
- COTS WiMAX testbed, based on a highly programmable WiMAX base station in standalone mode (no ASN-GW component), along with several open source WiMAX clients.
- A Software Defined Radio (SDR) 5G testbed, consisting of 10 USRPs N210, 12 USRPs B210, 4 USRPs X310 and 4 ExMIMO2 FPGA boards. MAC and PHY algorithms are able to be executed over the SDR platforms, with very high accuracy.
- A millimeter wave testbed, operating in the V-band (60GHz), based on six nodes [18]. The platforms support high data-rate point-to-point setups, with beam steering capabilities of up to 90 degrees with a step of 7.5 degrees.
- The nodes are interconnected with each other via 5 OpenFlow [19] hardware switches, sliced using the FlowVisor [20] framework.
- A Cloud Computing testbed, consisting of 96 Cores, 286 GB RAM and 10 TBs of hardware storage. For the provisioning of the cloud, OpenStack is used.
- Multiple WSN clusters, supporting the IEEE 802.15.4, 802.11 and LoRaWAN protocols [21], gathering measurements such as temperature, luminosity, air quality, radiation emission, etc.

The equipment is distributed across three different testbed locations in the city of Volos, and can be combined with each other for creating a very rich experimentation environment. The nodes are running any major UNIX based distributions.

A typical example of a NITOS node is given in Figure 12. Each node is equipped with two Ethernet interfaces (1 Gbps each), one used for controlling the node (e.g. establishing a secure shell connection onto it) and one of them is terminated in a hardware OpenFlow switch that is user controlled and is free of addressing so that each experimenter sets up their own settings for communication. The node is also equipped with two WiFi cards, one compliant with the 802.11ac standard and one with the 802.11n, that can be used by the experimenter in any possible mode (Access Point, Managed, Ad-Hoc, Monitor, Mesh). Some of the nodes are equipped with LTE USB dongles, that can be used to connect to the NITOS provided network, by using AT commands [22]. The dongles are using NITOS specific SIM cards for connecting to the LTE network. Finally, some nodes are equipped with USB3.0

based USRP devices (USRP B210 [23]). These can be used to execute software-based base stations, using tools such as srsLTE [24] and OpenAirInterface [17], which are widely used for prototyping 5G radio technologies. Each node has access to the mmWave infrastructure of the testbed, through the experimental Ethernet interface.

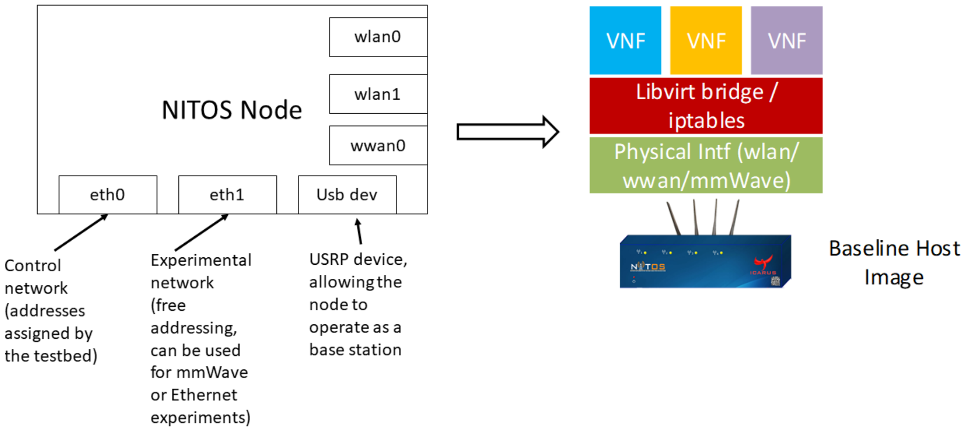

wlan0
wlan1
NITOS Node
wwan0
eth0
eth1
Usb dev
Control network (addresses assigned by the testbed)
Experimental network (free addressing, can be used for mmWave or Ethernet experiments)
USRP device, allowing the node to operate as a base station
VNF
VNF
VNF
Libvirt bridge / iptables
Physical Intf (wlan/ wwan/mmWave)
Baseline Host Image

Figure 12: NITOS node architecture

Through the 5GinFIRE extensions, each NITOS node can be used as a compute node. All of the nodes are interconnected through Ethernet technologies and are highly customizable in terms of wireless networking. Several technologies are available for experimenters, allowing them to setup VNFs in the testbed that uses a wireless link. The key technologies applied to the 5G-VINO extension regard Commercial Off-The-Shelf (COTS) LTE link provisioning, WiFi Link provisioning and mmWave point-to-point connection.

3.2 Inter-site and external communications

3.2.1 Addressing plan

As described in [1], the IP address space to be used by 5GinFIRE is allocated following these two rules:

1) 5G infrastructure providers will use the private address space 10.154.0.0/16 for control and data plane communications, i.e. an address range not in use at the sites that provided the first stable version of the 5GinFIRE MANO platform.
2) To simplify routing configurations inside 5TONIC, this specific location will use the private address space 10.4.0.0/16 to support control and data plane communications.

The 5GinFIRE network operations centre has been in charge of the allocation of IP address ranges to entities within the address space 10.154.0.0/16. The current allocation is shown in Table 1. In addition, the VPN service at 5TONIC has been configured to use subnetworks 10.154.255.0/24 and 10.154.254.0/24, being the latter the address range that has been allocated to VPN endpoints.

Table 1: Current allocation of IP addresses to sites

Site	IP address range
5TONIC	10.4.0.0/16
ITAv	10.154.0.0/20
UNIVBRIS&BIO	10.154.16.0/20
UFU	10.154.32.0/20
WINS_5G	10.154.48.0/20
5GVINO	10.154.64.0/20
eHealth5G	10.154.80.0/20

3.2.2 Requirements to support the interconnection of external sites

The requirements originally set for the interconnection of additional external sites have evolved, in line with the experience gained during the expansion of the project infrastructure. Every interconnection request is treated on a case-by-case basis, according to the following non-exhaustive list of requirements that must be fulfilled:

1) Utilization of a VIM solution compliant with the 5GiFIRE MANO platform (see section 2.1).
2) Utilization of an appropriate IP address space, not conflicting with the address space assigned to the 5GinFIRE infrastructure providers so far. Interconnecting entities must use a range of IP addresses within the network prefix 10.154.0.0/16. This range will be determined by the 5GinFIRE network operations centre, according to existing allocations and the entity's needs.
3) Obtaining VPN credentials to connect the entity's infrastructure to the network overlay architecture of the 5GinFIRE MANO platform.
4) Configuration of the site with the allocated IP address space. Deployment of address translation functions in case that this is needed.
5) Installation and configuration of the VPN endpoints that are necessary at the external site.
6) Configuration of appropriate VIM networks, to enable the exchange of control and data-plane information originated and terminated at the VxFs deployed at the entity's datacentre.
7) Set up of the appropriate mechanisms to support the delivery of control and data-plane information across the local network segments of the external entity (i.e. from the VPN endpoints towards the VIM/SDN controller and VxFs, and vice versa).

3.2.3 Support of Internet access to experiments

Initially, no global decision was taken by 5GinFIRE with regards to the access to Internet from the deployed VNFs. Rather, this decision was postponed until the end of the first Open Call so we could use direct input from experimenters on their exact necessities.

Experimenters confirmed their requirements regarding Internet access for actions like the downloading of new software packages, or the upgrade of some VNF features, to name just a few. This, and the connectivity in the management plane to control their VNFs (which was already available) were their only requirements.

As a result, it was agreed that no common access point to the Internet was going to be provided, but the sites could directly provide this connectivity to their on-boarded experiments. As a security measure, it was also agreed that the Internet connectivity would be provided in the data plane only, completely decoupled from the management plane (where critical infrastructure, like OSM, could be accessible).

As next steps, and beyond Internet access, it was also agreed to analyse in a deeper way what other "external" connections to 5GinFIRE existed at the different sites. The fact is that none of the sites is totally isolated for this project, and most of the infrastructure is shared with other activities at which not controlled (by 5GinFIRE) connections might be created. The idea with this activity is to understand the implications of such scenario from a security point of view and analyse if any of the security measures should be revisited.

3.2.4 Securing the orchestration infrastructure

[1] described initial security policies. On the one side, as the management network was shared by all experimenters, it was required that non-trivial credentials were configured on VxFs. On the other side, rules were configured in the Jump Machine providing external access, to avoid undesired flows like those towards the MANO components, for example. An update for such rules is given in Section 3.2.5.

However, as 5GinFIRE experimental facilities grew, both in number and capabilities, the security policies for the MANO Platform were revisited, mainly due to the fact that the 5GinFIRE infrastructure at each site is not fully isolated (as described in Section 3.2.3). In particular, new firewalling capabilities have been included.

Since the number of critical components to be secured at 5GinFIRE is relatively low (one OSM at 5TONIC, plus one VIM per site), instead of deploying a firewall application at each site, "iptables" rules have been defined. These rules are then to be applied in the OSM and VIM virtual machines at the logical interface attached to the management network.

The rules for OSM differ slightly depending on the OSM Release, since they make use of a different set of TCP ports. For OSM Release TWO, the ACL looks like this[2]:

```
# OSM Release TWO iptables
# Accept established traffic, i.e., responses to flows started at OSM
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT -i <interface_name>
# Permit required IP addresses for full access (as many as required): e.g. 5TONIC operator, Portal, etc.
sudo iptables -A INPUT -s <source_subnet> -j ACCEPT -i <interface_name>
# Permit required ports (specific for Release TWO):
sudo iptables -A INPUT -p tcp --dport 8000 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 4567 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 8008 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 9090 -m state --state NEW -j ACCEPT -i <interface_name>
# Permit required ports
sudo iptables -A INPUT -p tcp --dport 8443 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT -i <interface_name>
# Drop not required flows
sudo iptables -A INPUT -i <interface_name> -j DROP
sudo iptables -A FORWARD -i <interface_name> -j DROP
```

[2] Variable parameters in the ACL are included in italics and between angle brackets: e.g. *<interface>*

In OSM Release FOUR, however, some ports can be eliminated from the list, while some others are added (in particular, those required for the monitoring features). The ACL includes:

```
# OSM Release FOUR iptables
# Accept established traffic, i.e., responses to flows started at OSM
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT -i <interface_name>
# Permit required IP addresses for full access (as many as required): e.g. 5TONIC operator, Portal, etc.
sudo iptables -A INPUT -s <source_subnet> -j ACCEPT -i <interface_name>
# Permit required ports (specific for Release FOUR monitoring):
sudo iptables -A INPUT -p tcp --dport 3000 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 12340 -m state --state NEW -j ACCEPT -i <interface_name>
# Permit required ports
sudo iptables -A INPUT -p tcp --dport 8443 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT -i <interface_name>
# Drop not required flows
sudo iptables -A INPUT -i <interface_name> -j DROP
sudo iptables -A FORWARD -i <interface_name> -j DROP
```

Finally, the proposed ACL for the VIM, is:

```
# Openstack iptables
# Accept established traffic, i.e., responses to traffic started from VIM
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT -i <interface_name>
# Accept traffic from OSM
sudo iptables -A INPUT -s <OSM_source_IP> -j ACCEPT -i <interface_name>
# Permit required IP addresses for full access (as many as required): e.g. 5TONIC operator
sudo iptables -A INPUT -s <source_subnet> -j ACCEPT -i <interface_name>
# Drop not required flows
sudo iptables -A INPUT -i <interface_name> -j DROP
sudo iptables -A FORWARD -i <interface_name> -j DROP
```

3.2.5 Provision of access to experimenters

The external access mechanism, by which experimenters can access their experiments in the management plane, was already described in Deliverable D4.1 [1]. Since then, it has been successfully used in the first Open Call, so the mechanism has not been changed moving forward: experimenters establish a VPN connection with 5TONIC, which redirects them to a Jump Machine (JM) from which they can SSH their VMs.

Of course, the ACL in the JM that governs to which IP addresses in the management plane experimenters can or cannot access has been extended, to account for the new sites coming from the Open Call. The filter definition looks like this:

# Jump Machine filter definition

1. If ICMP → apply ICMP policer and accept
2. If dest_IP == OpenVPN → apply SSH policer and accept
3. If dest_IP == Infrastructure → drop
4. If dest_IP == VxF Management → apply SSH policer and accept
5. Rest → drop

With the first term, ICMP is permitted, so connectivity tests can always be executed.

To understand terms 2 to 4, it is worth reminding that the JM only has one physical interface with two uses: VPN clients make SSH to the JM first, and once logged in, they make a second SSH to their VxFs. Term 2 permits that only the authorized OpenVPN clients access the JM. Instead, term 4 permits SSH access only to the list of authorized VxFs. This term, together with the “drop” at term 5, avoids connections to components like the OSM or the VIMs.

Term 3 accounts for potential ranges which, although inside the VxF Management range, should also be forbidden.

With this definition, it is very easy to include new sites, since site owners only have to provide to the 5TONIC operator:

- Subnet within their full range to be used for VxFs in the management plane (VxF Management range)
- Infrastructure (= forbidden) subnets within this VxF Management range, if any

The current configuration in the JM (Juniper M7i node) is as follows:

```
policy-options {
   prefix-list OpenVPN-addresses {
      10.254.0.0/16;
      10.255.0.0/16;
   }
   prefix-list Infra-5TONIC {
      10.4.16.1/32;
      10.4.16.11/32;
      10.4.16.15/32;
      10.4.48.1/32;
      10.4.48.11/32;
      10.4.48.13/32;
   }
   prefix-list Infra-ITAv;
   prefix-list Infra-UNIVBRIS;
   prefix-list Infra-WINS5G;
   prefix-list Infra-eHealth5G;
   prefix-list Infra-5GVINO;
   prefix-list Infra-UFU;
   prefix-list VxFs-5TONIC {
      10.4.16.0/21;
      10.4.48.0/21;
   }
   prefix-list VxFs-ITAv {
      10.154.0.0/21;
   }
   prefix-list VxFs-UNIVBRIS {
      10.154.24.6/32;
      10.154.24.7/32;
      10.154.24.8/32;
      10.154.28.0/24;
      10.154.29.21/32;
   }
   prefix-list VxFs-WINS5G {
      10.154.48.0/24;
      10.154.50.0/24;
      10.154.52.0/24;
      10.154.54.0/24;
      10.154.56.0/24;
      10.154.58.0/24;
      10.154.60.0/24;
   }
   prefix-list VxFs-eHealth5G {
      10.154.82.0/23;
      10.154.90.0/23;
   }
   prefix-list VxFs-5GVINO {
      10.154.64.0/23;
   }
   prefix-list VxFs-UFU {
      10.154.38.0/24;
   }
}
```

The above is the full definition of prefix lists, to be referenced later on in the filter. Those prefixes named “Infra-“ plus the testbed name include the infrastructure ranges (those to be forbidden). Instead, those name “VxFs-“ plus the testbed name include the VxF Management ranges.

```
firewall {
   policer incoming-ssh {
      if-exceeding {
         bandwidth-limit 1m;
         burst-size-limit 15k;
      }
      then discard;
   }
   policer outgoing-ssh {
      if-exceeding {
         bandwidth-limit 1m;
         burst-size-limit 15k;
      }
      then discard;
   }
         count ICMP-packets;
         accept;
      }
   }
   term incoming-OpenVPN {
      from {
         destination-prefix-list {
            OpenVPN-addresses;
         }
      }
      then {
         policer incoming-ssh;
         count OpenVPN-packets;
         accept;
      }
         count Infrastructure-packets;
         discard;
      }
   }
   term 5GinFIRE-VxF {
      from {
         destination-prefix-list {
            VxFs-5TONIC;
            VxFs-ITAv;
            VxFs-UNIVBRIS;
            VxFs-WINS5G;
            VxFs-eHealth5G;
            VxFs-5GVINO;
            VxFs-UFU;
         }
```

```
policer icmp {
   if-exceeding {
      bandwidth-limit 100k;
      burst-size-limit 15k;
   }
   then discard;
}
filter protect-5GinFIRE {
   interface-specific;
   term ICMP {
      from {
         protocol icmp;
      }
      then {
         policer icmp;
   }
   term 5GinFIRE-infra {
      from {
         destination-prefix-list {
            Infra-5TONIC;
            Infra-ITAv;
            Infra-UNIVBRIS;
            Infra-WINS5G;
            Infra-eHealth5G;
            Infra-5GVINO;
            Infra-UFU;
         }
      }
      then {
      }
      then {
         policer outgoing-ssh;
         count VxF-packets;
         accept;
      }
   }
   term default {
      then {
         count default-packets;
         discard;
      }
   }
}
```

In the firewall configuration above we have several policers, and the filter itself. The policers permit ensuring that a single connection will not be able to consume too much bandwidth. The filter is the implemented configuration matching the general definition described above. Finally, explicit ports are not included in any of the filter rules; this is controlled in any case by the 5GinFIRE infrastructure, since the only access the JM allows for external experimenters is via SSH.

The mechanism described in this section, as mentioned before, has been already in use during the first phase of experiments. As lessons learned, we realized that it was not fully clear at the beginning for experimenters what they needed to do to obtain external access. To solve this, and beyond the "How To" that was already available in the Wiki, the key message to include management interfaces at their VNFs was also included in the Wiki pages that dealt with how to build VNFs.

Finally, as a sort of survey, it was requested from experimenters to define additional connectivity requirements that they thought were missing, obtaining a single response:

- The possibility for devices connected to the VPN to access the VxFs in a port different than port 22 (SSH).

Since the JM does not have the hardware required to implement port forwarding, the simplest option to achieve this behaviour is the redirection capabilities that SSH permits, not blocked in 5GinFIRE.

3.3 Mechanisms for evolution and continuous integration

One of the most salient characteristics of current software development practices is the support for continuous integration techniques, allowing a seamless evolution of the operational environments as their software bases evolve. In this respect, the agreement made by 5GinFIRE partners, in order to ease future integration and maximize the impact of the project results, is to present and contribute any modifications to the 5GinFIRE MANO platform to upstream integration by the OSM community. Moreover, given the extremely active nature of the foreseen MANO software evolution, 5GinFIRE infrastructure providers will proactively consider the incorporation of any breakthrough features that result from the evolution of the platform software base. These updates will be planned and executed at each site without impacting the operational availability of the MANO framework as a whole.

Table 2 collects information about the infrastructure and equipment that is available at each site to support validation and testing of partners' developments over a pre-production environment, as well as the validation of new components, functionalities, and releases of the software base of the 5GinFIRE MANO platform.

Table 2: Pre-production environment of 5GinFIRE

Site	Description
5TONIC	UC3M maintains an independent small-scale MANO deployment, to support testing of new releases of OSM and OpenStack, which provide the basis for the MANO framework deployed at 5TONIC. This small-scale deployment will also support experimentation with other OSM components and plugins, e.g., to test new types of VIMs, as well as the validation and debugging of new components and extensions to the 5GinFIRE MANO platform. At the time of writing, the small-scale MANO platform at UC3M has six mini-ITX computers (each with an Intel Core i7 2.3 GHz, 8 GB RAM, 128 GB SSD, and 4 GbE ports with DPDK capabilities). These can be used to flexibly build diverse datacenter configurations. For example: an OSM stack, two OpenStack VIMs to enable multi-site experiments over two datacentres, and a set of compute nodes for each datacentre. To support research and experimentation activities related to virtualization over single board computers, an additional mini-ITX computer hosts an installation of OSM Release FOUR and an OpenStack VIM. This setup enables the deployment of lightweight VNFs using containers over an NFVI conformed by seven Raspberry Pi 3 Model B (this corresponds to a work in progress at UC3M, see section 5.4). To complement the testing environment at UC3M, so we were able to test in parallel, another setup was created at TID premises. The initial goal was the evaluation of the OSM Monitoring framework, but in the end also some tests regarding different OpenStack versions have been executed. This pre-production testing environment is not permanent at TID, but it will be used for as long as it is available. Current infrastructure includes: • 1 Dell PowerEdge R730 server, 128GB RAM, and Haswell architecture, to host both the OSM (Release FOUR) and the VIM (Openstack Queens) Virtual Machines. • 1 Dell Power Ede R720, 64GB RAM, and Ivy Bridge architecture, used as NFVI.
ITAv	ITAv will maintain an independent small-scale MANO deployment, to support testing of new releases of OSM and OpenStack, which provide the basis for the MANO framework deployed at ITAv. This small-scale deployment will also support experimentation with other OSM components and plugins. It will also help validate new components developed and new hardware/software integration. In this small-scale deployment, it will be possible to test the project's AAA component under development (see section 0) without causing testbed downtime. It will also allow to test a third-party platform/solution to monitor VNFs and their configuration process. This software will help us in the future to determine the reasons why a VNF failed or why the configuration was not successful.
UNIVBRIS	UNIVBRIS will maintain a NFVI, which is SDN enabled providing to the experimenters the proper platform for VxF experimentations. The number of

Site	Description
	compute nodes will be increased according to the experiment demands. Mobile Edge Compute (MEC) nodes will be available over the 5GUK Test Network infrastructures to support the Smart City scenario mainly to include communication with Raspberry PIs Model B. The MEC nodes will enable to move the VxFs from the Cloud to the Edge Future evolution of the UNIVBRIS platform will focus on the integration of OpenStack with ODL, as well as with OSM MANO. For this end, all the procedures are going to be performed in a separated infrastructure, called UNIVBRIS Testing, before being deployed at production environment. UNIVBRIS Testing is a controlled environment inside of the Smart Internet Lab at UNIVBRIS that enables different kind of configurations and testing to guarantee the proper operation of the extension being experimented. Once everything is operated adequately the next step is to perform the migration to the production environment.
UFU	UFU will continue its evolution to support new experimentation scenarios. We plan to integrate new devices in the MEC related to smart city vertical such as Raspberry Pi, and Arduino controlled sensors. Another future task is to integrate the 5GINFIRE facility with a live frequency below 1 GHz, that will be provided by the Brazilian Regulatory agency in partnership with a local telecom operator, called Algar Telecom. Considering the different resources from other testbeds at UFU we want to investigate new FIRE integration scenarios considering the infrastructure deployed in Brazil from different EU-Brazil joint calls such as FIBRE, FUTEBOL, and NECOS, thus bringing to 5GINFIRE new test scenarios and capabilities.
eHealth5G	PSNC plans to established a third OpenStack NFVI basing on integrated telecommunication platform Radisys ATCA-4770 equipped with six ATCA-4745 CPM blades with dual Intel Xeon E5-2600, 128GB SAS disk, and 64GB RAM each. This testing NFVI infrastructure will be managed by local installation of OSM instance and will be available to any pre-production tests.
WINS_5G	WINS_5G maintains a preproduction environment with an OSM server (Release 4), OpenStack, and independent access to ports on a Dell Networking S4048T-ON SDN switch. This environment will test OpenStack and OSM upgrades before they are migrated to the WINS_5G production system and help debug problems on the production system. It will also support open call experimenters, deploying and testing developed OSM network services, without impacting the production environment or other experimenters. Currently, this system includes four USRP N210s with SBX daughter boards supporting frequencies between 400 MHz and 4.4 GHz, 1 x Dell PowerEdge R440 server acting as an OpenStack controller and compute node, and 2 x Dell PowerEdge R620 servers operating as compute nodes. This system will act as the main test environment to support integration of EU projects including FUTEBOL, and Fed4FFIRE+ into the WINS_5G architecture in 2019. Furthermore, it will support WINS_5G extensions for Openflow, OSM, the Open Network Operating System (ONOS) integration and testing. The preproduction environment currently hosts OSM Release 4, with plans to add OSM

Site	Description
	release 5 VM in 2019. Once applications and network services have been validated in the preproduction environment, they will be migrated to the production system.
5GVINO	University of Thessaly maintains an isolated MANO framework that is being used for local tests in the testbed. The VIM of the testbed is being exposed to the 5GinFIRE orchestrator and experiments can take place over the testbed. There are no dedicated nodes from the testbed provided to 5GinFIRE; since NITOS is using a reservation mechanism for exposing the testbed to Fed4FIRE, each experiment needs to reserve some of the physical nodes of the testbed prior to orchestration. Therefore, the team has developed a solution for dynamically retrieving all scheduled experiments for NITOS and prior to orchestration the framework automatically reserves the compute nodes needed for the experiment using the testbed's SFA API. Hence, the testbed is available for using it in two manners: 1) for testing purposes in order to develop new solutions/VNFs and 2) to test already developed VNFs through the 5GinFIRE framework.

3.4 Tools for prototyping and testing

3.4.1 The Mirror platform

To enable experimenters and developers to carry out a first validation of their NS and VxFs, 5GinFIRE provides a functional clone of the OSM stack currently in production state. This clone, known as the mirror platform, complements the service offered by the 5GinFIRE portal providing platform users with access to an OSM installation that supports the following functionalities:

1) Onboarding of VxFs.
2) Onboarding of NS.
3) Access to OSM logs, to get information on any onboarding errors.

This way, the mirror platform allows platform users to verify if a NS or VxF can be onboarded to the 5GinFIRE MANO system, before formally requesting the onboarding of the component to the production MANO platform through the 5GinFIRE Portal. Its main purpose is to agilely detect errors in NS/VxF package specifications early in the experimentation process. This way, users can autonomously do a preliminary validation of their NS/VxF packages, making sure that they are compliant with the OSM software and can effectively be onboarded to the production MANO platform. This allows reducing potential interaction cycles and delays that would otherwise be necessary to set up an experiment with new NS/VxFs, optimizing the experimentation process.

Regarding the mirror platform, the following points must be considered:

- The mirror platform does not attach any VIM tenant, hence it cannot be used to test the deployment of a NS.
- The mirror platform may be restarted with certain periodicity, and data may be deleted for maintenance purposes. Hence, it cannot be used as a stable platform to keep copies of NS and VxFs.

3.4.2 Local prototyping & testing toolset

In addition to the mirror platform, 5GinFIRE partners have also prepared a local toolset that provides a complete and functional orchestration environment to support prototyping and testing. This toolset includes:

- An installation of OSM Release FOUR
- A VIM emulator solution, Vim-emu [25]

The local toolset is provided as a single virtual machine, which can be downloaded by experimenters and other interested users from the 5GinFIRE website. With this toolset, experimenters may also test if a NS or VxF can be onboarded to OSM (and hence to the 5GinFIRE MANO system).

On the other hand, *Vim-emu* is capable of emulating the functionality of a VIM and an NFVI, providing a network emulation framework and supporting the deployment of VNFs as Docker containers. This way, the local toolset also complements the functionalities of the 5GinFIRE Portal, providing a mechanism to assist experimenters in the prototyping and testing of NSes.

4 Functional validation

4.1 Integration tests

This section describes the functional tests and experiments that have been carried out to verify the appropriate operation of the 5GinFIRE MANO platform, once the version of the baseline OSM software of the platform has been updated to OSM Release FOUR[3]. These tests have served to validate the capacity of the platform to deploy multi-site experiments on the sites made available by 5GinFIRE infrastructure providers.

The functional tests that have been performed are described in Table 3. In a first test (test ID #1), the OSM stack at 5TONIC was used to deploy a reference NS at every site offered by the 5GinFIRE infrastructure providers. The NS was composed of two interconnected VNFs (see Figure 13), which did not require additional configuration through the Juju interface of the OSM stack. Upon the successful deployment of the NS, both VNFs were completely functional and capable of exchanging data using the virtual data link that was configured between them. This test served to validate the capacity of the overlay network architecture of 5GinFIRE to support control plane communications between the OSM stack and each external VIM, to coordinate the allocation and configuration of computing, storage and network resources at the VIM's site.

Table 3: functional tests to validate the overlay network architecture of 5GinFIRE

Test ID	Description	Sites	Result
1	Deployment of a reference NS on each site, not requiring day-1 configuration of VNFs	5TONIC, ITAv, UNIVBRIS, UFU, eHealth5G, WINS_5G, 5GVINO	Success
2	Deployment of a reference NS on each site, requiring day-1 configuration of VNFs	5TONIC, ITAv, UNIVBRIS, UFU, eHealth5G, WINS_5G, 5GVINO	Success
3	Deployment of a reference NS using the multi-site capabilities of the 5GinFIRE MANO stack	5TONIC, ITAv, UNIVBRIS, UFU, eHealth5G, WINS_5G, 5GVINO	Success

In a second test (test ID #2), the OSM stack at 5TONIC was utilized to instantiate a different reference NS at every site. In this case, the NS consisted of two interconnected VNFs[4] that required day-1 configuration operations via Juju. The NS is represented in Figure 14. After the automated deployment of the virtual machines of the NS, both VNFs were successfully

[3] Migration from OSM Release TWO (i.e., the version of OSM used during the first phase experiments that took place in 2018) to OSM Release FOUR was successfully carried out in December 2018.

[4] The Ping Pong network service is available at (last access: December 2018): https://osm-download.etsi.org/ftp/osm-3.0-three/examples/ping_pong_ns/

instantiated and configured through their corresponding Juju charms, and the NS was completely functional, being both VNFs capable of exchanging data through a virtual data-link. This experiment served to validate the capacity of the overlay network architecture of 5GinFIRE to support inter-site control communications related to the lifecycle management of NS and VNFs.

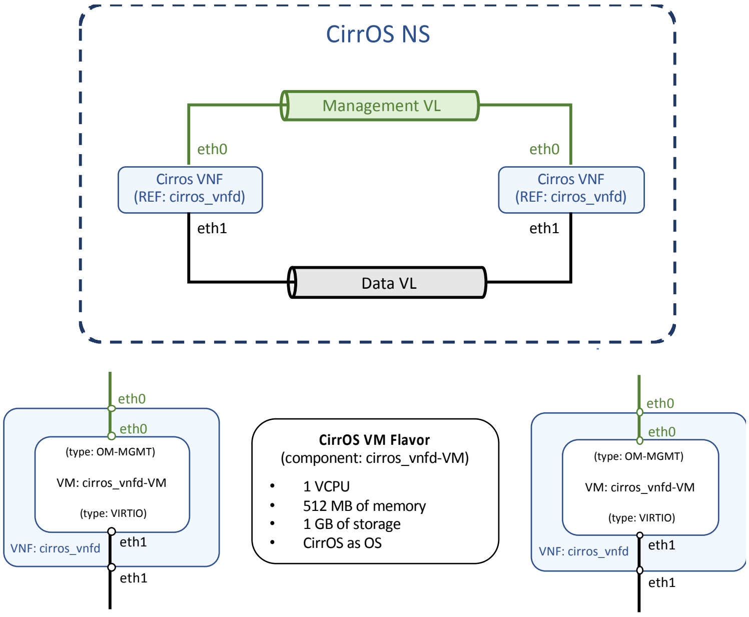


Figure 13: Reference NS #1 used to validate inter-site communications

Finally, in a third test (test ID #3), the OSM stack was used to deploy the same network service as in the previous test, although leveraging the multi-site support of the 5GinFIRE MANO platform. In this case, the OSM stack was instructed to deploy one of the VNFs at one of the datacentres of 5TONIC, while the other was instantiated one of the other sites. After the successful deployment of the NS, both VNFs (the one at 5TONIC and the other one at the chosen site) were able to exchange data through the VPN-based overlay network of 5GnFIRE. The deployment of the NS was repeated for each of the sites offered by 5GinFIRE infrastructure providers. These deployments allowed validating the multi-site capacity of the 5GinFIRE MANO platform.

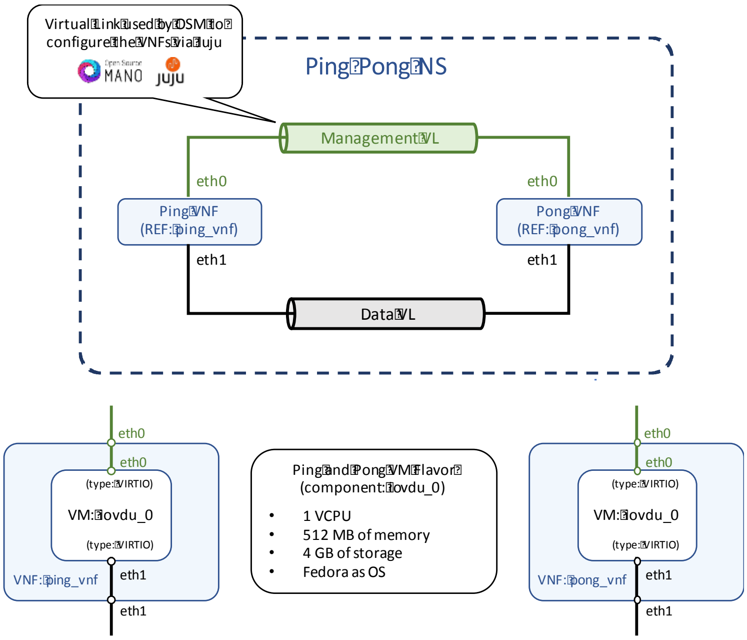


Figure 14: Reference NS #2 used to validate inter-site communications

4.2 Performance of the orchestration service

We carried out a number of experiments to explore the performance of the 5TONIC MANO platform and its scalability properties. In a first experiment, we evaluated how the deployment of an NS is affected by existing deployments. With this purpose, we performed 16 successive deployments of an NS with a single VNF, measuring the time required for each deployment. We repeated each cycle of 16 deployments 30 times. The left side of Figure 15 (labelled as *Test 1*) shows the average deployment time for the first, second, fourth, eighth and sixteenth deployment of the NS. According to the obtained results, we observe that the deployment time of an NS is not significantly affected by previous deployments.

In a second experiment, we studied how the deployment time of an NS is affected by the number of its constituent VNFs. With this purpose, we deployed an NS at 5TONIC with a number of interconnected VNFs (1, 2, 4, 8 and 16). For each case under consideration, we repeated the deployment 30 times and calculated the average deployment time. The results of this experiment are shown in the right side of Figure 15 (labelled as *Test 2*). According to these results, the average time required to deploy an NS with a single VNF is lower than 40 s. This time increases almost linearly with the number VNFs that composes the NS.

The results of these experiments suggest that the 5TONIC MANO platform has the potential to instantiate moderately large NS with appropriate deployment times, considering the key performance indicators established by the 5G Infrastructure Public Private Partnership [26] (i.e., average service creation time not higher than 90 minutes). Once the migration to OSM Release FOUR has been completed, the project aims at repeating the aforementioned experiments using the NFV infrastructures provided by 5GinFIRE infrastructure providers. This will serve to delimit the performance that can be achieved by the 5GinFIRE orchestration service using distributed NFVI deployments.

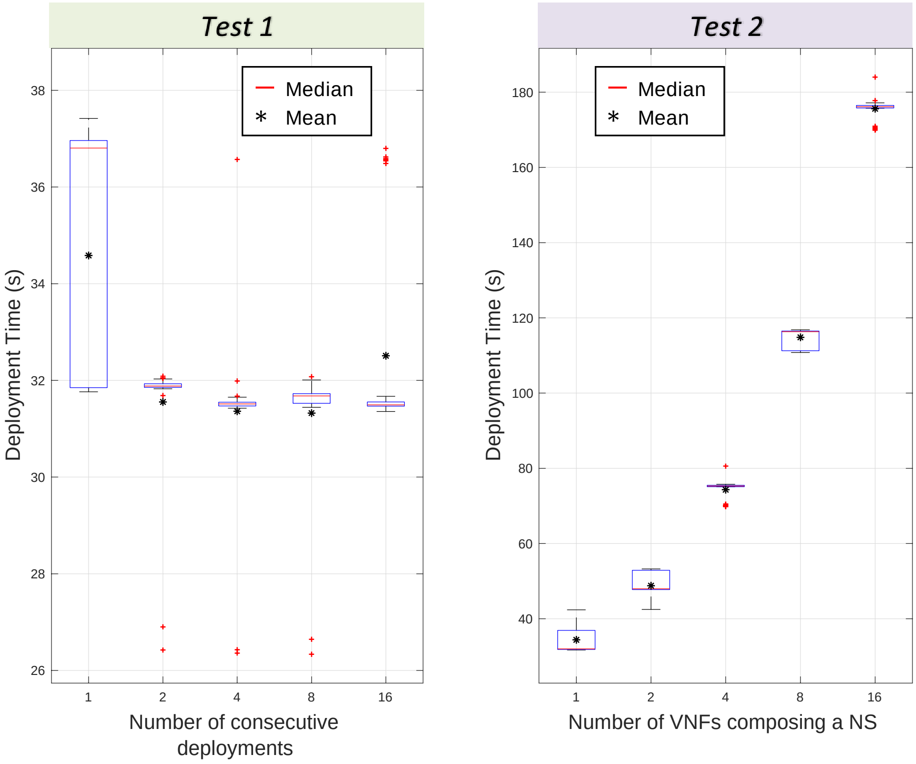


Figure 15: Performance of the orchestration service

5 Enhancements to the orchestration platform

5.1 Configuration of VNFs via Ansible playbooks

As we already commented, the VCA module is the OSM component in charge of the configuration of VNFs, generally aligned with the VNF manager entity of the ETSI NFV reference architectural framework. It presents an interface to Juju [27], allowing the configuration of VNFs through the execution of a limited form of Juju charms, called VNF Configuration charms or proxy charms[5].

Aiming at increasing the range of configuration options available to VNF developers in OSM, as well as facilitating the portability of existing VNF developments, 5GinFIRE explored the utilization of other well-known and wide-used mechanisms for VNF configuration during the first project's year, identifying Ansible [28] as a technology of particular interest.

Under the aforementioned considerations, we developed a base charm layer, *ansible-charm*, that allows the configuration of a VNF using an Ansible playbook. This base charm layer includes the Juju base layers vnfproxy[6] and ansible-base[7], and provides a template ready for customization that allows creating a proxy charm that supports the execution of an Ansible playbook using the Juju framework of OSM. Moreover, due to the interest shown by the community, the details about the utilization of this development was included in the OSM Wiki[8].

Finally, starting from the preliminary work compliant with the OSM Release TWO, the development has been included into the source code of OSM Release FOUR[9], which is the current version of the 5GinFIRE MANO stack. Our future work includes the continuation in the maintenance and consolidation of this contribution across the new releases of OSM.

5.2 Integration of Keystone within the MANO platform

OSM Release TWO lacked the proper means to authenticate users and authorize their actions. Having identified this issue, and in order to address 5GinFIRE requirements, a decision was made to develop a solution to address this situation. As requirements for the new solution, 2 main ones were defined:

1) Ease of integration with multiple backends (ex. LDAP, Kerberos)
2) Widely used solution

After some design iterations, the choice was made to reuse Openstack Keystone since it fulfilled all the aforementioned requirements and was already part of the 5GinFIRE ecosystem. Also, it provided a configurable Role Based Access Control system which could be reused and integrated into OSM in order to authorize user actions. After some discussion with the OSM community, an agreement was reached where an Authentication Plugin

[5] Creating your own VNF charm (Release FOUR), OSM Wiki (last access: Dec. 2018): https://osm.etsi.org/wikipub/index.php/Creating_your_own_VNF_package.

[6] vnfproxy, Juju charm layer (last access: Dec. 2018): https://github.com/AdamIsrael/vnfproxy

[7] Ansible Base Layer for Charms (last access: Dec. 2018): https://github.com/chuckbutler/ansible-base

[8] Example VNF charms (last access: Dec. 2018): https://osm.etsi.org/wikipub/index.php/Example_VNF_Charms

[9] Ansible-charm contribution to the OSM Release FOUR source code (last access Dec. 2018): https://osm.etsi.org/gerrit/#/c/6760/

System would be developed inside OSM Release 4 NBI, to then connect with Keystone (Figure 16). The Authentication Plugin System would allow other developers from the community to write their own plugins for other Authentication Systems, while Openstack Keystone would be the default used by OSM.

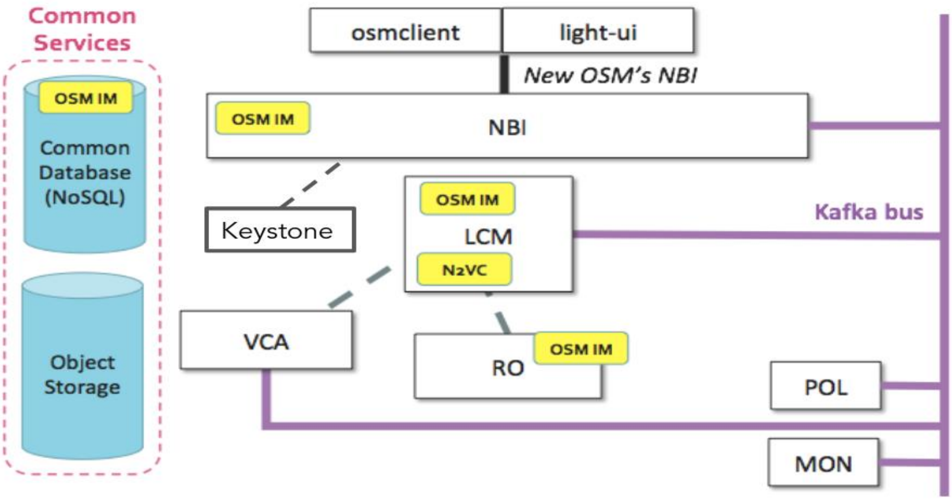


Figure 16: Keystone integration in OSM internal architecture

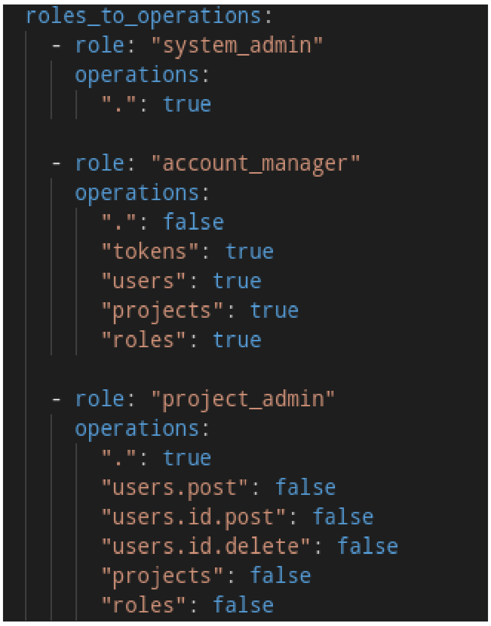
For user action authorization, the RBAC system provided by Openstack Keystone was reused. Keystone's RBAC system is highly configurable, which allows users to create their own roles and apply them to user on a per-project basis. This allows for a highly configurable environment which allows OSM to reap the benefits of a widely used component such as Openstack Keystone. Openstack Keystone does not solve all the problems by itself and three problems remained to be solved:

1) What roles are required?
2) How should those roles be specified?
3) How should roles be stored?

Once again, this was a discussion handled in unison with the OSM community, and four roles were defined:

- System Administrator (administration of the whole system)
- Account Manager (management of users and projects)
- Project Administrator (administration of users and resources assigned to a project)
- Project User (user permissions inside a project for VNF management)

Having defined the roles, it was necessary to discuss the permissions associated with each role, as well as how are these permissions described and stored. The implemented solution was a hierarchical tree structure (Figure 17). In this hierarchical tree structure, the value defined in a parent node will propagate to every element in the subtree, unless there is a leaf node with a value specifying otherwise.



```
roles_to_operations:
  - role: "system_admin"
    operations:
      ".": true

  - role: "account_manager"
    operations:
      ".": false
      "tokens": true
      "users": true
      "projects": true
      "roles": true

  - role: "project_admin"
    operations:
      ".": true
      "users.post": false
      "users.id.post": false
      "users.id.delete": false
      "projects": false
      "roles": false
```

Figure 17: Hierarchical tree structure for permission definition

Besides the specification, there is also the need to store these mappings. An important aspect in deciding the storage mapping is between saving space when in the database while keeping a fast lookup. It was decided to keep the original format saved in the database, which ensures higher fidelity to what the user expects while maintaining a high compression ratio since all the nodes are not expanded. While this preserves space in the database, it increases complexity and time when doing lookups to verify if a user is authorized or not. In order to solve this, a caching mechanism was introduced, where the fully expanded nodes are saved. When the system is initiated the permissions are read from a file (if it is the first the system is running) or from the database, then the permissions are expanded and stored in the caching mechanism. This allows for less complex lookups since it will query for the full path of the node, without the need to expand the lookup path. Since it does not read the permissions from the database, but it reads them from memory it will also speed up the process.

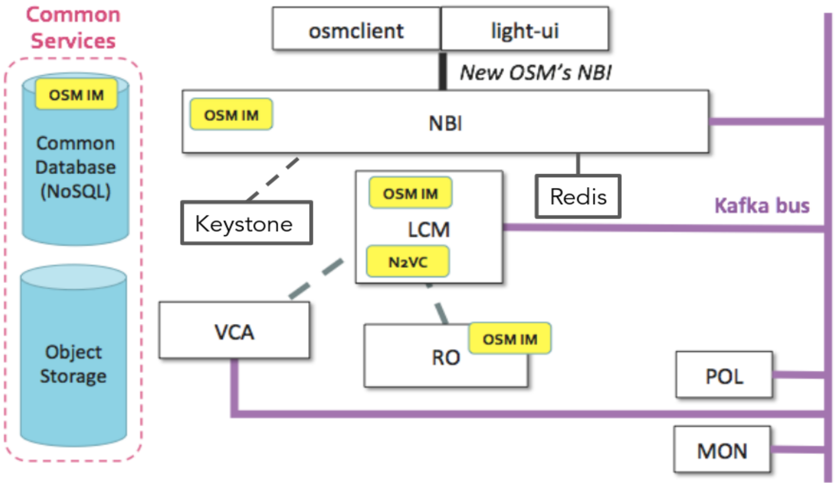


Figure 18: Redis cache in OSM internal architecture

The use of an internal caching system brings problems when trying to deploy OSM in Highly Available setups and to fix this issue the internal caching system was replaced by a Redis server (Figure 18). Redis allows all the instances of the NBI to be synchronized and avoids complex mechanism in order to synchronize them. It will also make the NBI stateless which eases its deployment in a Highly Available setup.

All these proposals and changes have been integrated into the OSM codebase as of release 5, yet they'll only be active when the final Redis cache code is merged, which should occur during the merge window for OSM Mid-point Release 5.0.1.

5.3 SDN and WIM integration

The previous versions of OSM MANO do not enable connectivity among VNFs located in different datacentres across different Wide Area Network (WAN). The specification of WAN Infrastructure Manager (WIM) in ETSI NFV MANO architecture makes this possible. In this direction, UNIVBRIS has developed a system that enables protocol agnostic WIM functionality in OSM based on heterogeneous SDN control in WAN. Figure 19 shows the SDN integration in OSM-RO for unified multi-SDN connectivity. In this figure, an SFC composed of VNF-1, VNF-2 and VNF-3 should be deployed via OSM MANO to different datacentres. The VNF-1 and VNF-2 must be deployed at PoP A and VNF-3 at PoP B. At the end, it is expected that VNF-2 communicates with VNF-3. The overall methodology can be summarized as follows.

- Exposing to WIM the VLANs and VIM endpoints for the instantiated 'external' VLDs (between PoPs - VIMs)
- VLAN Provider Networks
- Extending the SDN port mapping
- Reusing the SDN Controller integration APIs for registering WIM
- Multi-datacentre detection during NS deployment
- No changes to OSM information model required (NSD/VNFD)

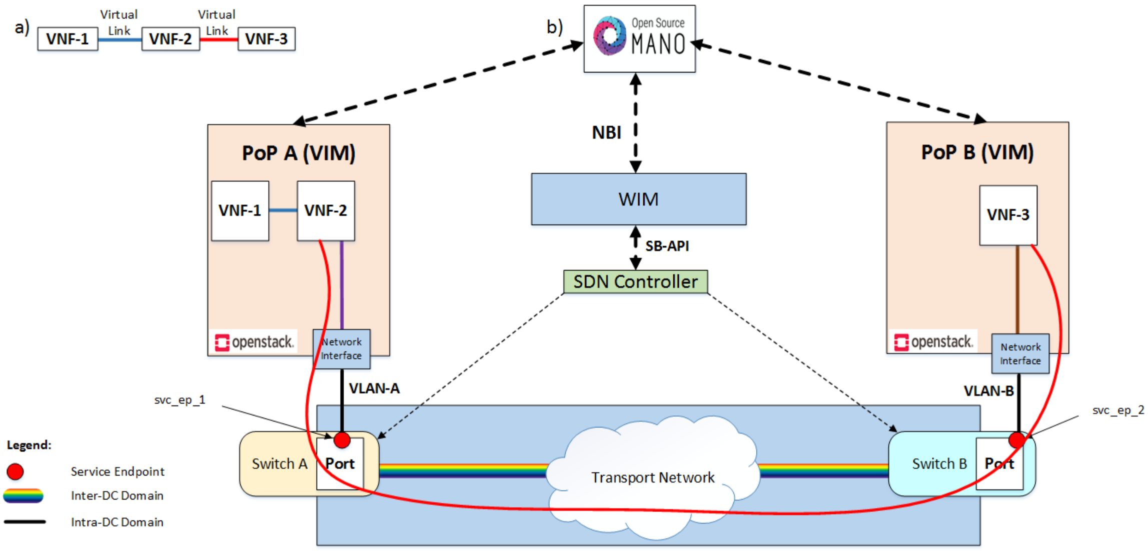


Figure 19: VNF chaining across multi-PoPs (Points of Presence) in OSM.

Basically, a Network service (NS) is composed of VNFs and virtual links. When deploying a NS via OSM the VNFs run on NFVI managed by VIMs (OpenStack in our case); the virtual links remain within VIM domain or external to VIM domain; the VLAN IDs of external virtual links are sent to OSM by VIM; OSM sends VLAN IDs of virtual links to WIM and finally the WIM provisions flows by connecting the VLAN flows. This solution is already integrated to OSM MANO platform and will be available at OSM MANO Release FIVE.

Regarding to SDN-assist in OSM, the UNIVBRIS has been working with SR-IOV and PCI-PASSTHROUGH. The last one works with the VNFs. However, the integration considering VNFs with SR-IOV does not work properly since the VNFs cannot be instantiated. According to UNIVBRIS findings there is a bug in OSM MANO Release 4 which we have solved as follows. In OSM MANO code, specifically at the "vimconn_openstack.py" RO container, the value for *hw:mem_pag_size, hw:cpu_policy, hw:numa_mempolicy* is hardcoded. Updating the value in *hw:mem_pag_size* in the running Docker container from *large* to *small* has solved the issue. After this, we are able to create the instances from OSM and we can see the SR-IOV virtual functions getting associated with the VMs with the correct MAC addresses.

5.4 Orchestration of lightweight NS over Single Board Computers

One enhancement to the current 5GinFIRE orchestration platform that has been explored is the use of Single Board Computers (SBCs) as compute nodes within an operational cloud-computing platform. This enhancement aims at supporting a limited-capacity and inexpensive infrastructure capable of instantiating lightweight VNFs, that is, VNFs with a computational cost that is not considerably high in comparison with the capacity provided by current infrastructure resources. The result of this development will be provided to the NFV community as best current practices to support the orchestration of lightweight NS over a NFVI conformed by constrained devices.

To develop the abovementioned enhancement, we selected the Raspberry Pi (RPi) model 3[10] as the SBC to act as a compute node in the 5GinFIRE infrastructure, and OpenStack (release Ocata [10]) as the open-source cloud-computing platform. This composition presents a significant challenge since the RPi does not support the hardware acceleration needed for the instantiation of traditional hypervisor-based virtual machines, which encourages the use of containers for virtualization.

The container virtualization technology has different benefits such that containers can be ported to different compute nodes with different capacities, they make an efficient use of the resources (since they do not require a separate Operating System), and they effectively isolate and share the resources with other containers in the same host, to name a few examples. On the other hand, containers are ephemeral in order to speed up the operations of creating, starting, replicating, or destroying a container. In this respect, LXC containers[11] provided by Linux operating system has been the selected technology to develop this enhancement.

The results of this development have been leveraged to carry out experimentation and validation activities in a research work of UC3M [29] [30], which explores the adaptable and automated deployment of applications over Small Unmanned Aerial Vehicles (SUAVs). This

[10] Raspberry Pi – Teach, Learn, and Make with Raspberry Pi (last access: December 2018): https://www.raspberrypi.org

[11] Linux Containers (last access: December 2018): https://linuxcontainers.org

research work is closely related with the work on the smart city use cases that is being conducted within 5GinFIRE. In the following, we provide a short summary of this work, to illustrate the potential of the aforementioned enhancement.

5.4.1 Overview of the research work

Our research work studies the applicability of virtualization technologies and standards to build flexible SUAV deployments, capable of agilely adapting to different missions in the civilian scope. With this objective, we consider the utilization of NFV, along with general purpose hardware and software platforms that can be on-boarded into small aerial vehicles. These platforms will provide the underlying substrate to execute network, transport and application level functions, which will be implemented in software and deployed over the SUAVs as required using virtualization technologies.

The use of NFV technologies in SUAV environments introduces a series of technical challenges that must be carefully considered. On the one hand, the hardware equipment that can be on-boarded in an SUAV, providing the underlying substrate to support the execution of virtualized functions, is typically limited in terms of size and weight. This is especially evident in the case of the small-sized UAVs that have recently emerged in the market, which could at best carry a small set of Single-Board Computers (SBCs). In this way, the hardware platform for the execution of virtual functions can present significant limitations in terms of computing capacity, networking and storage. On the other hand, such hardware must be integrated as part of the NFV infrastructure of an existing VIM solution, in such a way that it can be used by an NFV orchestration service to deploy virtualized functions. Additionally, it should be possible to make an appropriate distribution of VNFs to SUAVs, in order to indicate which VNFs should be executed on the same aircraft. Moreover, the control communications that are necessary to manage the infrastructure resources carried by the SUAVs should be maintained, even when these devices are flying. Finally, we must consider that the SUAVs have a battery, and therefore a limited operating time. For this reason, specific mechanisms may be required to support the replacement of these devices, including the migration of the virtualized functions hosted by them to new or existing SUAV units. We must bear in mind that these requirements are not common in traditional virtualization platforms, where compute nodes are typically high-performance servers, installed in a data-centre, and interconnected through a high capacity fixed network technology (e.g., an Ethernet network).

Given the complexity of these challenges and to guarantee the architectural and technological convergence of our work with existing solutions, our proposal has been designed in accordance with recognized standards in the NFV arena. In particular, we consider the NFV reference architecture proposed by ETSI [8]. In general terms, the design of our solution (see Figure 20) is structured into three main components.

1. The MANO system, located in a Ground Control Station (GCS), is the component in charge of the management and orchestration of available hardware and software resources, as well as the deployment and interconnection of the lightweight VNFs. The GCS offers a stable environment with appropriate resources for the operation of this component, which is convenient given its criticality.
2. The hardware and software infrastructure that supports the execution of lightweight VNFs, i.e., the NFVI. As we have already mentioned, this infrastructure will be on-boarded in the SUAVs.

3. The mission planner, also located at the GCS, which is the entity responsible for specifying the descriptors of the network services to be deployed, each one as a composition of VNFs, in addition to the configuration parameters of each VNF and the policies that determine the allocation of the VNFs to the SUAVs.

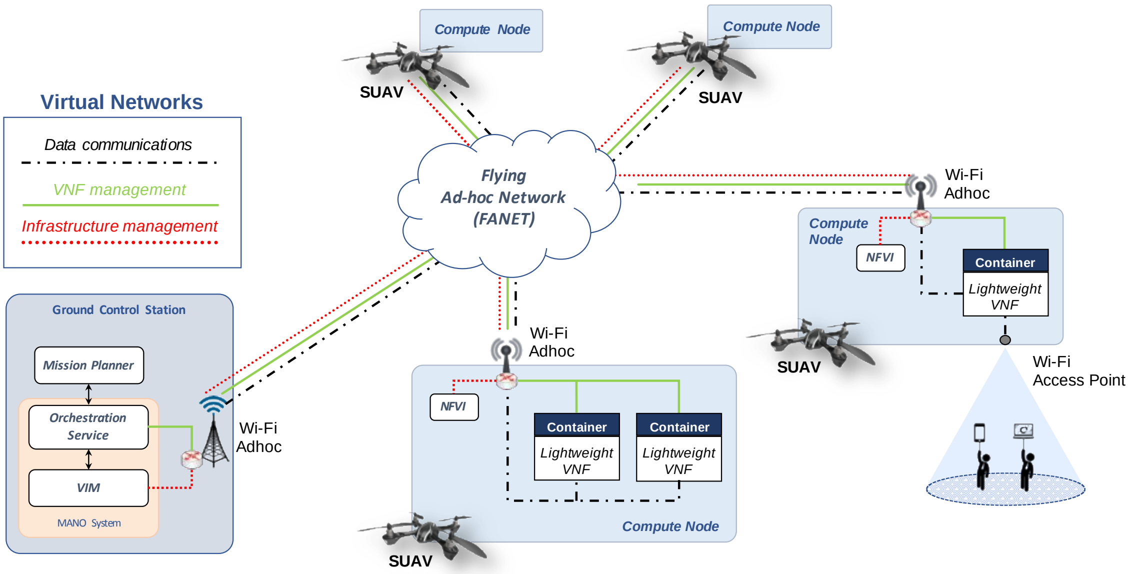


Figure 20: Overview of the platform design

Focusing on the architectural design of the SUAV device, each SUAV carries a general-purpose hardware and software platform. This platform, hereafter referred to as a compute node following cloud computing terminology, has a wireless communication interface that enables the exchange of data with every other component of our design that is within the radio coverage of the compute node (i.e., every other compute node and the MANO system). This wireless interface can be based on different technologies, e.g., it may provide a line-of-sight or Wi-Fi radio link. Additionally, as reflected in the figure, some compute nodes may include a secondary network interface to provide wireless access connectivity to ground units.

In our design, the compute nodes on-boarded at the different SUAVs form a wireless ad-hoc network that enables multi-hop data communications (that is, communications among compute nodes at different SUAV units, and between the compute nodes and the MANO system). These data communications across the ad-hoc network are supported with the execution of a Flying Ad-hoc Network (FANET) routing protocol at each compute node (e.g., AODV [31] or OLSR [32]) (we assume that the support of this routing protocol is enabled at each SUAV as a prior step to an operational deployment). Besides, to isolate the different traffic types exchanged over this wireless ad-hoc network, our solution defines a set of virtual networks that operate on top of the wireless ad-hoc network.

In particular, to support management operations towards the compute nodes (e.g., to monitor the available resources at each compute node, to instantiate and configure a VNF instance, or to terminate that instance), each compute node included in the NFV infrastructure supports two types of communications:

1. Communications between the VIM and the compute nodes (labelled as "Infrastructure management" in Figure 20), to control the computing, storage and networking resources of the compute nodes.
2. Communications between the Orchestration Service and the VNFs (denoted as "VNF management" in Figure 20) to manage the configuration and lifecycle of the lightweight VNFs

In our design, each of these types of management communications is delivered over an independent virtual network, which is created over the wireless ad-hoc network infrastructure offered by the SUAVs (i.e., the virtual network operates over the FANET routing protocol executed at every SUAV). Thus, management communications are isolated and delivered ``in-band'', through the aerial network infrastructure conformed by the compute nodes.

Regarding data communications among the lightweight VNFs that conform a network service (indicated as "Data communications" in Figure 20), these are also isolated using virtual networks that are built on top of the wireless ad-hoc network established by the compute nodes. The number and configuration of these virtual networks will typically be application-specific (e.g., IP telephony traffic would require an independent virtual network). Hence, they will automatically be created by the MANO system as required, under the indication of the Mission Planner.

Finally, as a specific design criterion regarding the deployment of virtual functions, we want to highlight that, given the limited-capacity hardware and software platforms that can be provided and/or transported by SUAVs, we use container virtualization as opposed to traditional hypervisor-based virtual machines, to support the deployment of the lightweight VNFs in our design. This is a crucial consideration that has a significant impact on the practical feasibility of the proposed design.

5.4.2 Validation of the Solution

In this section, we describe the most relevant aspects regarding the validation of our solution. Taking into consideration our experiments and the lessons learned with the utilization of Raspberry Pi model 3 as compute nodes in the 5GinFIRE infrastructure, a functional prototype of the aforementioned design was implemented. In addition, we developed a number of lightweight VNFs, to support the creation of moderately complex NS that have served to test and validate the proposed design.

5.4.2.1 Prototype Implementation

The use of open-source technologies was considered one of the basic principles used throughout the implementation process. In particular, we selected Open Source MANO (OSM) Release FOUR [6] to provide the functionalities corresponding to the orchestration and the lifecycle management of lightweight VNFs. With respect to the VIM, we utilized OpenStack Ocata [10]. Both the OSM stack and the VIM provide the MANO system of our design and were deployed over a virtual machine running in a mini-ITX computer (Intel Core i7 2.3 GHz, 16 GB RAM, 128 GB SSD, 4 GbE ports).

Regarding the SUAV platforms, we used a number of aerial vehicles DJI Phantom 3[12], each carrying an RPi. These RPis are used as the compute nodes of our design, supplying the needed resources in terms of computing, storage and networking, and supporting the execution of the lightweight VNFs. We incorporated the RPi boards as compute nodes of the OpenStack VIM. Besides, each RPi includes an integrated Wi-Fi interface, and a number of them also contain a secondary wireless interface provided by a Wi-Fi USB adapter. The first interface enables air-to-air and air-to-ground ad-hoc communications with other SUAVs and with the MANO system, respectively. The second interface is intended for deploying a wireless access point, capable of providing network access connectivity to mobile ground units (in a prior research work [33], we validated the suitability of these multi-interface devices to support multimedia communications).

With respect to the virtual networks that enable both control and data plane communications that take place in our platform, we used *Virtual eXtensible Local Area Networks* (VXLAN) [34]. VXLANs present a feasible solution to exchange control and data traffic over a wireless ad-hoc network since:

1. The VXLAN traffic can be sent over the Wi-Fi interface in ad-hoc mode that is available at every compute node (directly sending traffic from a lightweight VNF, deployed over a virtualization container, through a Wi-Fi ad-hoc network is challenging, as this is not currently supported by the Linux kernel of the RPis).
2. They can be dynamically created by the VIM, as instructed by the OSM stack, to interconnect lightweight VNFs hosted by different compute nodes
3. The utilization of VXLANs does not require additional network configurations (e.g., network routes) at the intermediate RPi boards that conform the network path between two communicating entities (e.g., between two VNFs).

5.4.2.2 Validation Scenario

This section describes the experiment carried out to validate the potential of our platform to execute realistic and moderately complex network services. In this experiment, we implemented the Network Service (NS) shown in Figure 21, including the NFV descriptors (NSD and VNFDs) and the set of lightweight VNFs that conform the NS. The NS provides the functionality of an IP telephony service, and supports the delivery of telemetry from every SUAV to an equipment in the ground control station of the SUAVs. The NFV descriptors and the lightweight VNFs have been developed and made available under an open source license[13].

Regarding the lightweight VNFs that enable the provision of the IP telephony service, the NS includes two virtual access points (Access Point VNFs in Figure 21), which support data communications between end user equipment (i.e., ground units) located at different areas. Each of these virtual access points holds a DHCP server, which automatically provides the required network configuration to allow IP access connectivity to connected users. On the other hand, the IP telephony service enables the exchange of signalling traffic to establish multimedia sessions between communicating endpoints, i.e., Voice over IP (VoIP) calls

[12] DJI. Phantom 3 Professional (last access: Dec. 2018): https://www.dji.com/phantom-3-pro

[13] 5GinFIRE MANO GitHub repository (last access: December 2018): https://github.com/5GinFIRE/mano/tree/master/descriptor-packages

among user equipment at ground units. To support this service, we developed a lightweight VNF providing the functionality of a VoIP server, based on the open source software Kamailio [35]. This VoIP server VNF enables the establishment of voice communications with the utilization of the Session Initiation Protocol (SIP) [36]. Finally, a lightweight VNF provides the functionality of a DNS server, resolving host names to IP addresses during the execution of the IP telephony service.

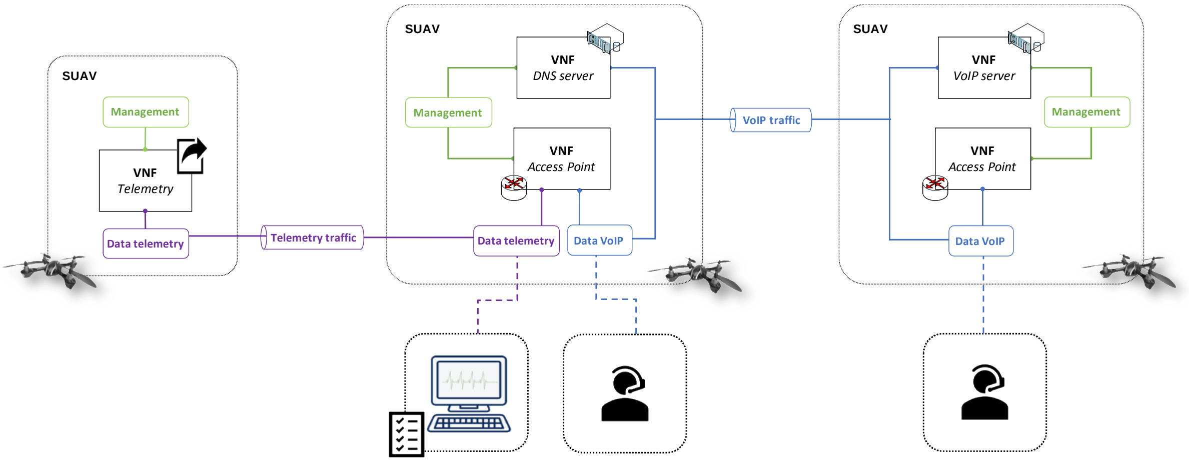


Figure 21: Validation scenario

On the other hand, the exchange of telemetry information (GPS positioning, information from on-boarded sensors, etc.) from an SUAV to a ground station is allowed by the lightweight VNF referred to as *Telemetry* in Figure 21. In this case, we emulated the transmission of telemetry data using the Iperf tool[14], which enables the generation of different data streams from a source to a destination. In our test scenario, telemetry information was emulated as a UDP stream with a bandwidth of 32 kbps.

With this, we instantiated the NS over our SUAV platform using the client application of OSM. This instantiation results in the execution of the aforementioned VNFs in virtual containers at the SUAVs, following the disposition indicated by Figure 21. For that end, the definition of each SUAV as an availability zone (i.e., a set of resources) in the OpenStack VIM allowed us to execute each VNF at the expected SUAV through the placement policies provided by OSM. The lightweight VNFs composing the IP telephony service communicate through a VXLAN that is established over the Wi-Fi ad-hoc network. Telemetry information is exchanged with the GCS using a different VXLAN (over the Wi-Fi ad-hoc network as well). Both VXLANs are dynamically created by the VIM, as instructed by the OSM stack. This way, and according to our system design, telemetry traffic is isolated from the VoIP traffic. Day-1 configuration of the lightweight VNFs (configuration of static IP addresses, activation of IP forwarding in the access points, and the start of DHCP, DNS and SIP services) was executed with Ansible playbooks, using our base charm layer *ansible-charm* presented in section 5.1.

Once the deployment and the configuration of the lightweight VNFs was completed, we connected a wireless VoIP phone ZyXEL Prestige 2000W to each of the access points offered by our NS, with the objective of carrying out a VoIP call. In addition, a commodity laptop was

[14] Iperf – The ultimate speed test tool for TCP, UDP and SCTP (last access: December 2018): https://iperf.fr

connected to the access point involved in the operations related with the exchange of telemetry information, emulating an equipment at the GCS.

Figure 22a, Figure 22b illustrates the traffic exchanged during the VoIP call established by the VoIP terminals, including the voice traffic, as well as the DNS and the SIP signalling messages needed to execute the call. The call was made without errors and with appropriate sound quality. Figure 22c represents the control traffic generated and received periodically by the MANO system, which is exchanged in order to monitor and synchronize the information of the available/consumed resources at each compute node. This traffic was measured to get an insight on how control traffic operates in our platform once an NS is being executed, and to verify how this traffic may affect the provided service. As expected, the measurements show that this traffic is negligible (with an average of 20.65 kbps) compared with the traffic exchanged during the execution of an NS, e.g., during the VoIP call. Finally, Figure 22d shows the bandwidth consumption of a telemetry stream received at the ground control station (i.e., at the laptop), originated by the Telemetry VNF of one of the SUAVs.

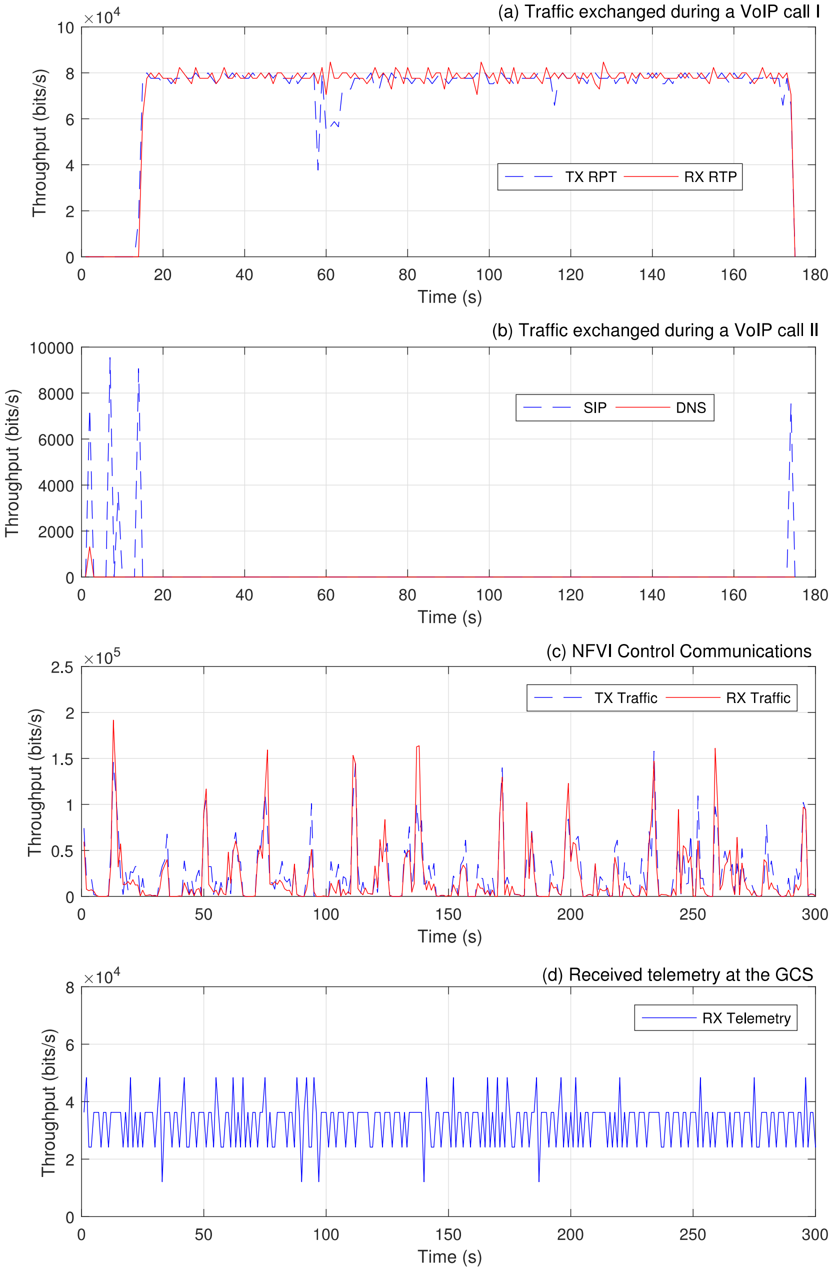


Figure 22 (a, b, c, d): Validation measurements

5.5 Function and service descriptor validation

With the onboarding of new Virtual Network Functions (VNFs) and Network Services (NSs) through the portal, an important step is to validate the new submissions to verify if they are ready to be deployed to the testbeds. In order to provide a testing mechanism, a validation pipeline was created using a Continuous Integration (CI) Server. The chosen tool was Jenkins.

Our CI server is configured at ci.5ginfire.com and it holds the validation pipeline. This pipeline is responsible for validating the descriptors and send information about the tests performed.

The process is illustrated by the following diagram (Figure 23):

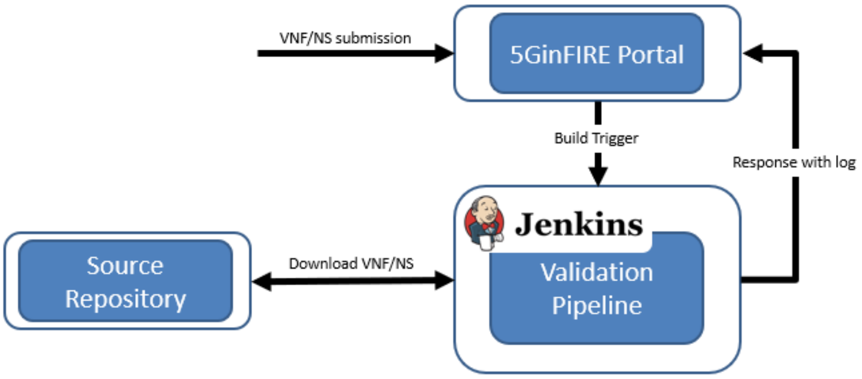


Figure 23: VNF and NS validation workflow

As the pipeline supports VNFs and NSs, it must detect which type of package was onboarded.

On one hand, if a new VNF is submitted, the pipeline is triggered and the following steps are performed:

1) **VNF Download**: The VNF metadata is retrieved and the package is downloaded. Its contents are extracted and the descriptor is identified.
2) **Tests Download**: OSM tests are being used, hence, in order to keep them always updated, they are fetched from their git repository every time the pipeline is triggered.
3) **VNF Validation**: Since some experimenters may submit descriptors that were built according to older versions and the OSM tests work only for the latest release, the first action taken is to upgrade the descriptors. Then, a syntactical test is applied, and its result is sent to the portal.

In terms of the validation of VNFs, we are planning on adding a direct verification with OSM by deploying the VNF to a testing site and collect some metrics.

On the other hand, if a NS is submitted, the pipeline is triggered and the VNFs that compose the NS are identified. Then, to each one of the VNFs, we are planning on adding an agent. The agent is responsible for collecting information about the connectivity of a VNF with the other VNFs and sending it to the collector. The collector holds the information of the VNFs such as its IP address. The NS is then deployed to a testing site and the agent starts sending information. If the connection between any VNFs is not possible, the agent will send a timeout and the pipeline will send a failure message with the logs. Oppositely, if all the VNFs have a connection between each other, the pipeline will send a pass message with the corresponding logs.

The described mechanism is presented on the picture below (Figure 24):

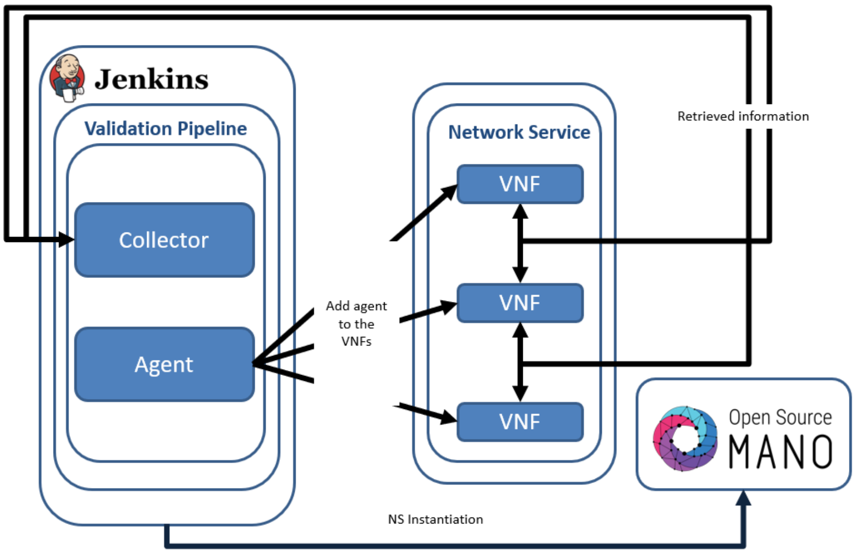


Figure 24: Network service validation architecture

5.6 Monitoring pipeline

Another important test to make is to verify if the testbeds are working properly. Hence, the creation of a testbed monitoring pipeline is a plausible solution since it offers an automated testing mechanism.

The base tools are the same of the validation pipeline, that is, Jenkins is also being used and the monitoring pipeline is configured at ci.5ginfire.com.

We assume that a testbed is working properly if it can deploy a NS. Along these lines, we created a pipeline that is triggered at a fixed interval of time and tries to deploy a simple NS in each one of the testbeds, gathering information about whether the instantiation was successful or not.

The diagram below describes the pipeline interaction (Figure 25):

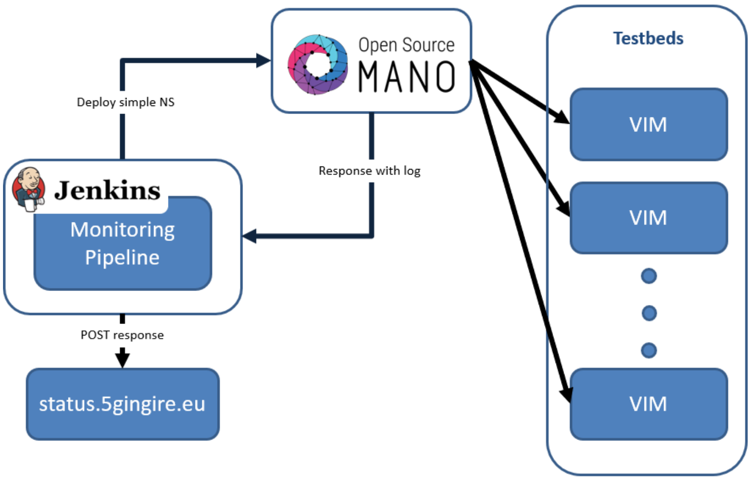


Figure 25: Testbeds monitoring workflow

The pipeline has three steps:

1) **Testbeds identification**: Using the OSM client, the list of the testbeds is fetched.

2) **NS deployment**: The simple NS is deployed to each one of the testbeds collected in the previous step.
3) **Data Parsing**: The OSM responses are parsed and the relevant information is extracted. It is also in this step that the POST message will be sent.

In future work, we still need to make better handling of errors, gather more information about the deployments and integrate the pipeline with status.5ginfire.com.

5.7 Telemetry: monitoring and performance measurements

5.7.1 Monitoring framework

Since the Release FOUR Lightweight build of OSM, new features have been added that permit monitoring and visualization of VNF metrics. This new service is based on OpenStack Telemetry mechanisms, for example, those offered by Ceilometer and Gnocchi. The new OSM MON component grabs metrics from the OpenStack telemetry modules, and puts them in the Kafka Bus, from where they can be read.

In particular, extensions have also been added to OSM installer to include an optional Kafka Exporter, which:

- Reads metrics from the Kafka Bus
- Exposes them in Prometheus, so they can be stored in its database
- Presents them in Grafana

This architecture is presented in Figure 26, directly extracted from the OSM Wiki. It is worth noting that the architecture has already changed for Release FIVE, so the current information in the OSM Performance Management web page has already been updated to the newest version.

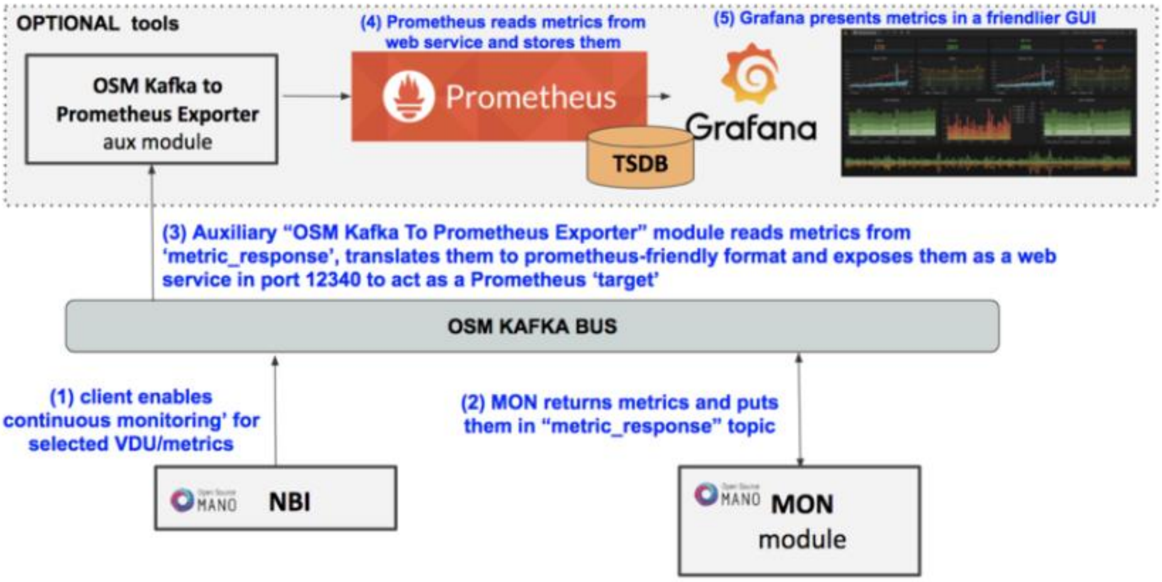
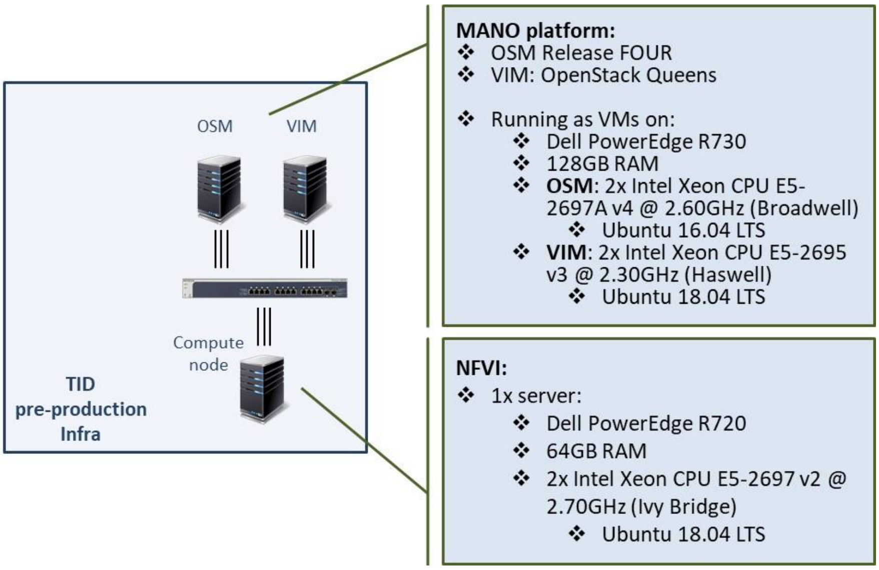

OPTIONAL tools
(4) Prometheus reads metrics from web service and stores them
(5) Grafana presents metrics in a friendlier GUI
OSM Kafka to Prometheus Exporter aux module
Prometheus
Grafana
TSDB
(3) Auxiliary "OSM Kafka To Prometheus Exporter" module reads metrics from 'metric_response', translates them to prometheus-friendly format and exposes them as a web service in port 12340 to act as a Prometheus 'target'
OSM KAFKA BUS
(1) client enables continuous monitoring' for selected VDU/metrics
(2) MON returns metrics and puts them in "metric_response" topic
Open Source MANO NBI
Open Source MANO MON module

Figure 26: OSM Performance Management architecture (Release FOUR)

In 5GinFIRE, as described in this document, the OSM versions in production have been Release TWO (first Open Call) and Release FOUR (ready for second Open Call). Therefore, the objective is to deploy these monitoring features together with the latter Release.

Initial tests were executed until October 2018 in a new pre-production environment deployed at TID premises, complementing the already available pre-production environment at 5TONIC. This infrastructure, which is not permanent, consisted of two servers, running the MANO platform on one side, and acting as NFVI on the other. Details can be found in Figure 27.



MANO platform:
- ❖ OSM Release FOUR
- ❖ VIM: OpenStack Queens
- ❖ Running as VMs on:
  - ❖ Dell PowerEdge R730
  - ❖ 128GB RAM
  - ❖ **OSM**: 2x Intel Xeon CPU E5-2697A v4 @ 2.60GHz (Broadwell)
    - ❖ Ubuntu 16.04 LTS
  - ❖ **VIM**: 2x Intel Xeon CPU E5-2695 v3 @ 2.30GHz (Haswell)
    - ❖ Ubuntu 18.04 LTS

NFVI:
- ❖ 1x server:
  - ❖ Dell PowerEdge R720
  - ❖ 64GB RAM
  - ❖ 2x Intel Xeon CPU E5-2697 v2 @ 2.70GHz (Ivy Bridge)
    - ❖ Ubuntu 18.04 LTS

Figure 27: TID pre-production infrastructure

Two were the main objectives of this deployment:

- Testing of the OSM monitoring framework
- Testing of new OpenStack versions

In particular, regarding OpenStack, "Queens" was the version selected for these tests since "Rocky", the following one, had just been released in August 2018, and could present maturity issues. The Ceilometer version was 10.0.2.

Results of the tests were very satisfactory, although irrelevant in themselves, since this was only a very simple pre-production environment and the objective of the testing was more focused on obtaining the skills to deploy and maintain the framework in the production network. Suffice to say that, based on this testing, a detailed list of metrics has been presented in Deliverable D6.1 [37] as candidates to be monitored.

5.7.2 Inter-site performance monitoring

In addition to the monitoring capabilities inherent to OSM, and considering the multi-site distribution of the 5GinFIRE infrastructure, the project also aims at developing a performance measurement solution, in order to evaluate the performance figures that can be achieved by inter-site communications, such as: average throughput among sites; end to end delay of inter-site communications; or delay jitter for network communications established among sites. These figures will provide valuable information to infrastructure providers and experimenters.

At the time of writing, the development of this performance measurement solution is in its final phase. A comprehensive description of this development and its current status is presented in D6.1 [37].

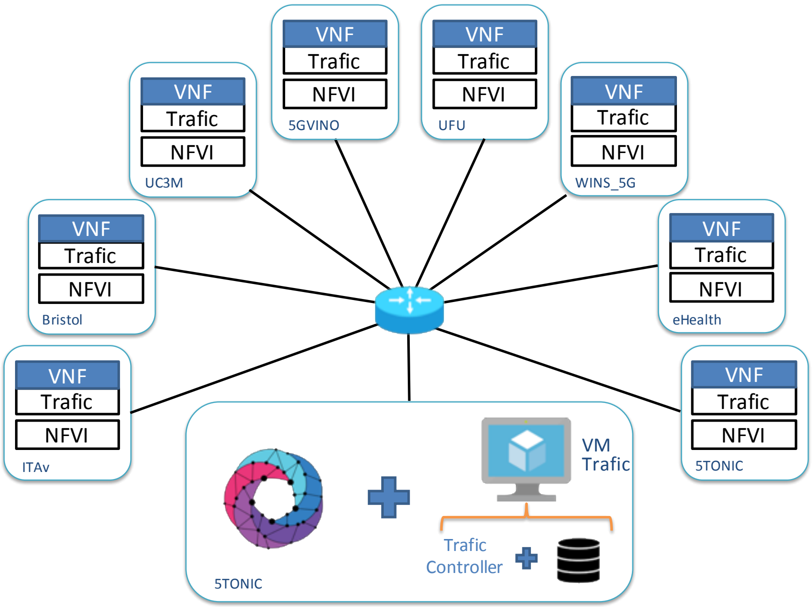


Figure 28: Architectural design of the performance measurement solution

For completeness, Figure 28 presents a high-level overview of the solution. The development is based on *Trafic* [38], an open-source traffic scheduler. In the design of the solution, a controller entity coordinates the execution of performance tests among sites. These tests are carried out by specific-purpose VNFs using of the *Trafic* tool. Each of these VNFs is deployed at a 5GinFIRE site. After the execution of a performance test, the results of the test are collected and stored into a data storage for further analysis. The specific purpose VNFs conform a NS that will be deployed using the 5GinFIRE MANO platform. In addition, *Trafic* provides the flexibility to configure different performance tests (e.g., sending TCP flows at the maximum available bandwidth, sending UDP flows, exchanging real-time interactive audio and video, etc.). The required configurations to run these tests can be predetermined when the VNFs are deployed, using the Juju charm support of OSM. Finally, the controller entity and the data storage unit will be deployed as static components at 5TONIC, using virtual machines.

6 Conclusion

The WP4 5GINFIRE team addressed the original orchestration challenges of a distributed, multi-domain, multi-tenant experimental environment based on the most advanced NFV technologies and deployed a MANO platform that has evolved with the experience gained by the contribution of the experiments and additional infrastructures incorporated using the project open calls.

The WP4 5GINFIRE team has continued addressing the orchestration challenges in the project, updating the original MANO platform, consolidating the validation mechanisms, and enhancing the aspects related to a multi-site and multi-user environment. The original documentation set has been extended, including reference materials for experimenters and experimental facility administrators. In other words, the MANO platform has grown, demonstrating its ability to incorporate new experimental sites, and providing a high-value experience for the deployment of NFV-enabled network testbeds.

At the same time, the team has consolidated the path for the MANO platform evolution and actively contributed to the upstream project of reference, building solid trust links with the open source community and, at the same time, providing strong advances in several research challenges in network orchestration.

References

[1] Diego R. López, et al., “Operational MANO Platform”, 5GinFIRE Deliverable D4.1, January 2018.

[2] Riwal Kerherve et al., “Evolving FIRE into a 5G-Oriented Experimental Playground for Vertical Industries”, 5GinFIRE Deliverable D2.1, May 2017.

[3] Diogo Gomes, et al., “5GinFIRE Experimental Infrastructure Architecture and 5G Automotive Use Case (update)”, 5GinFIRE Deliverable D2.2, September 2018.

[4] Open Source MANO. ETSI-hosted project (last access on Dec. 2018): https://osm.etsi.org

[5] 5TONIC: an open research and innovation laboratory focusing on 5G technologies (last access on Dec 2018): https://www.5tonic.org

[6] Adrian Hoban et al., “OSM Release FOUR Technical Overview”, ETSI OSM Community White Paper, May 2018.

[7] ETSI GS NFV-SOL 005, “Network Functions Virtualisation (NFV); Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point”, February 2018, http://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/02.04.01_60/gs_nfv-sol005v020401p.pdf

[8] ETSI GS NFV 002 V1.2.1, “Network Functions Virtualisation (NFV); Architectural Framework”, version 1.2.1, Dec. 2014.

[9] Adrian Hoban et al., “OSM Release TWO, A Technical Overview”, ETSI OSM Community White Paper, April 2017.

[10] OpenStack Ocata (last access on Dec. 2018): https://releases.openstack.org/ocata/

[11] http://5GINFIRE-5g.eu/ © 5GINFIRE consortium 2017

[12] Kist, M., Hypersvisor for Software Defined Radios (HyDRA), 2018, GitHub repository, https://github.com/maiconkist/gr-hydra

[13] Fdida, S., Friedman, T., & Parmentelat, T. (2010). “OneLab: An open federated facility for experimentally driven future internet research”, *New Network Architectures* (pp. 141-152). Springer Berlin Heidelberg.

[14] Fed4FIRE project (last access on Dec. 2018): www.fed4fire.eu

[15] Hiertz, G. R., Denteneer, D., Max, S., Taori, R., Cardona, J., Berlemann, L., & Walke, B. (2010). “IEEE 802.11 s: the WLAN mesh standard”. *IEEE Wireless Communications*, *17*(1).

[16] Makris, Nikos, et al. "Enabling open access to LTE network components; the NITOS testbed paradigm." *2015 1st IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2015.

[17] Nikaein, Navid, et al. "OpenAirInterface: A flexible platform for 5G research." *ACM SIGCOMM Computer Communication Review* 44.5 (2014): 33-38.

[18] Blu Wireless Technology WiGig products (last access on Dec. 2018): http://www.bluwirelesstechnology.com/

[19] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). "OpenFlow: enabling innovation in campus networks", *ACM SIGCOMM Computer Communication Review*, *38*(2), 69-74

[20] Sherwood, R., Gibb, G., Yap, K. K., Appenzeller, G., Casado, M., McKeown, N., & Parulkar, G. (2009). "Flowvisor: A network virtualization layer", *OpenFlow Switch Consortium, Tech. Rep*, *1*, 132.

[21] NITOS IoT Testbed and Applications (last access on Dec. 2018): http://dtveda.e-ce.uth.gr/wp-content/uploads/2017/07/DTVEDA2017_Korakis_NITOSIoTTestbedandApplications.pdf

[22] ETSI TS 127 007 V10.3.0 (2011-04) Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; AT command set for User Equipment (UE) (3GPP TS 27.007 version 10.3.0 Release 10)

[23] USRP Bus Series - B210 (last access on Dec. 2018): https://www.ettus.com/product/details/UB210-KIT

[24] Gomez-Miguelez, I., Garcia-Saavedra, A., Sutton, P. D., Serrano, P., Cano, C., & Leith, D. J. (2016, October). srsLTE: an open-source platform for LTE evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization* (pp. 25-32). ACM.

[25] M. Peuster, H. Karl, and S. v. Rossem: MeDICINE: Rapid Prototyping of Production-Ready Network Services in Multi-PoP Environments. IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, pp. 148-153. doi: 10.1109/NFV-SDN.2016.7919490. (2016)

[26] Public Private Partnership in Horizon 2020, "Creating a Smart Ubiquitous Network for the Future Internet", Advanced 5G Network Infrastructure for the Future Internet, November 2013.

[27] Juju, operate big software at scale on any cloud (last access on Dec. 2018): https://jujucharms.com

[28] Ansible, Automation for Everyone, Red Hat (last access on Dec. 2018): https://www.ansible.com

[29] Nogales, B., Sanchez-Aguero, V., Vidal, I., & Valera, F. "Adaptable and Automated Small UAV Deployments via Virtualization". *Sensors*, vol. 18, no 12, p. 4116. Nov, 2018.

[30] Nogales, B., Sanchez-Aguero, V., Vidal, I., Valera, F., & Garcia-Reinoso, J. "A NFV system to support configurable and automated multi-UAV service deployments". In *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications* ACM, pp. 39-44. Jun, 2018.

[31] Perkins, C.; Belding-Royer, E.; Das, S. "Ad hoc On-Demand Distance Vector (AODV) Routing"; RFC 3561; *Internet Engineering Task Force (IETF)*, 2003.

[32] Clausen, T.; Jacquet, P. "Optimized Link State Routing Protocol (OLSR)"; RFC 3626; *Internet Engineering Task Force (IETF)*, 2003.

[33] Sanchez-Aguero, V., Nogales, B., Valera, F., & Vidal, I. "Investigating the deployability of VoIP services over wireless interconnected Micro Aerial Vehicles". *Internet Technology Letters*, vol. 1, no 5, p. e40. Mar, 2018.

[34] Mahalingam, M.; Dutt, D.; Duda, K.; Agarwal, P.; Kreeger, L.; Sridhar, T.; Bursell, M.; Wright, C. "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks". RFC 7348; *Internet Engineering Task Force (IETF),* 2014.

[35] Kamailio. The Open Source SIP Server (last access on Dec. 2018): https://www.kamailio.org

[36] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Sparks, R.; Handley, M.; Schooler, E. "SIP: Session Initiation Protocol"; RFC 3261; *Internet Engineering Task Force (IETF)*, 2002.

[37] Michel Corriou, et al., "Infrastructure Operation Report", Deliverable D6.1, December 2018.

[38] Trafic: A traffic mix generator based on iperf3 (last access on Dec. 2018): https://github.com/mami-project/trafic