Deliverable D5.1 - 5G Testbed Integration and EVI Deployment Guidelines

Editor:	Flávio de Oliveira Silva, UFU	
Deliverable nature:	Report (R)	
Dissemination level:	Public (PU)	
Date: planned | actual	30 June 2018	4 July 2018
Version | No. of pages	1.0	55
Keywords:	Testbed Integration, Multi-site Orchestration, 5G, NFV, MANO, NFVI, Service Orchestration, NS, VxF, VNF, EVI, Automotive Vertical, Smart City Vertical., End-to-End Experimentation, 5GINFIRE Platform	

Abstract

This document presents a report on the 5G testbed integration about the ongoing tasks inside the work package “5G Infrastructure Integration and Experimentation Enablement” (WP5).

It reports on the 5G testbed integration so far and actions performed as well as a generic guide and process that 5G testbed providers can follow to enable their 5G infrastructure for experimentation

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: WP5 5G Infrastructure Integration and Experimentation Enablement

Number and title of task:

- Task 5.1: Overall 5GINFIRE platform integration and coordination
- Task 5.2: Enabling 5G Automotive Vertical Testbed for VxF Experimentation and other EVIs
- Task 5.3: FIRE testbeds as deployment platform
- Task 5.4 - Integration and support of open call facilities

Document title: Deliverable D5.1 - 5G Testbed Integration and EVI Deployment Guidelines

Editor: Flávio de Oliveira Silva, Federal University of Uberlândia

Work-package leader: Aloizio P. Silva, University of Bristol

Executive summary

The deliverable 5.1 provides a Report on the 5G Testbed Integration so far and actions performed, as well as, a generic guide and process that 5G testbed providers can follow to enable their 5G infrastructure for experimentation. In addition, this report also includes specification and architecture of the new experiments and testbeds selected in the first open call as part of the 5GINFIRE ecosystem. It is therefore positioned as a Report on Fifth Generation of Mobile Networks (5G) Testbed Integration and Experimental Vertical Instances (EVI) Deployment Guidelines that drives further integration work in WP5. This document also presents a joint result from the implementation work in WP4 and WP3, based on the requirements and specification on WP2, and new functionalities and capabilities from WP7 towards a deployable platform for Virtual Network and Vertical Functions (VxFs).

The main outcome is a guideline of the 5GINFIRE platform for experimenters and integration of new facilities/testbeds. This provides a guide into the main 5GINFIRE platform components and the interaction with the infrastructure provided by stakeholders such as UNIVBRIS, TID, UC3M, UoP, ITAv, UTH, PSNC and TCD. D5.1 is a mature version of D5.3 that not only includes the specification of 5GINFIRE ecosystem but also describes the integration among all 5GINFIRE components and new infrastructures providing the main points for the realization of WP2, WP3, WP4, WP5 by itself and WP7. D5.1 further provides information on relevant functionalities, capabilities and experimentation being added to the 5GINFIRE ecosystem through the first open call.

The methodology driving the guideline starts with the main concept that underlines the 5GINFIRE platform, most important related to user functionalities and the emerging EVIs for 5G networks (5G Automotive and Smart City) followed by a new EVI for the health vertical industry. New experiments selected in the first open call are also described for automotive and smart city EVIs. The 5GINFIRE ecosystem aims at reusing existing FIRE testbeds from other FIRE projects for deploying federated experiments. To this end, we present the integration strategy between 5GINFIRE and three FIRE projects FIESTA-IoT, NITOS and FUTEBOL. The steps for the integration of new infrastructures are delineated highlighting the main requirements for integration and the functional and performance tests needed to guarantee that the new testbeds are properly integrated, they are operational and ready to receive experiments. Finally, a brief description of the 5GINFIRE platform from the experimenter point of view is presented including a concise description of the VxFs available for the experimenters. Overall, D5.1 provides the reports on 5G testbed integration, it briefly describes the next steps to the final and complete ecosystem integration resulting in the validation of the overall 5GINFIRE platform through real uses cases deployment (Car Overtaking and Smart City Safety) in indoor and outdoor environments.

List of authors

Company	Author	Contribution
UFU	Flávio de Oliveira Silva	Section 1, Section 3, Section 6
UNIVBRIS	Aloizio Pereira da Silva	Executive Summary, Section 1, Section 2, Section 6
UoP	Christos Tranoris	Section 5
IT-Aveiro	Susana Sargento, Miguel Luís	Section 2
NKUA	Sokratis Barmpounakis	Section 2.2
UMU	José Santa, Antonio F. Skarmeta	Section 2.2
BCOM	Michel Corriou, Riwal Kerherve	Section 5.1
NETICTECH	Piotr Pawałowski	Section 3.2
UTH	Nikos Makris, Thanasis Korakis	Section 3.3
TCD	Diarmuid Collins	Section 3.4
EGM	Mengxuan Zhao	Section 3.2
UC3M	Ivan Vidal	Section 4
PSNC	Damian Parniewicz	Section 2.3

List of Reviewers

Company	Reviewer
EGM	Mengxuan Zhao
UoP	Spyros Denazis

Table of Contents

Abbreviations

5G: Fifth Generation of Mobile Networks

5GINFIRE: Evolving FIRE into a 5G-Oriented Experimental Playground for Vertical Industries

EVI: Experimental Vertical Instance

FIRE: Future Internet Research and Experimentation

IoT: Internet of Things

MANO: Management and Orchestration

MEC: Mobile Edge Computing

N-PMIPv6: Network Mobility (NEMO) Enabled Proxy Mobile IPv6

NFV: Networking Function Virtualization

NS: Network Service

NSD: Network Service Descriptor

OBD: On-Board Diagnosis

OBU: On-Board Unit

OSM: Open Source MANO

PNF: Physical Network Function

R&D: Research and Development

RSPEC: Resource Specification

RSU: Roadside Unit

SDN: Software Defined Networking

SIP: Session Initiation Protocol

SBC: Single-Board Computer

SFA: Slice-Based Federation Architecture

UGW: Unified Gateway

UE: User Equipment

V2X: Vehicle-to-Everything

VIM: Virtualized Infrastructure Manager

VLAN: Virtual Local Area Network

VNF: Virtual Network Function

VxF: Virtual Network and Vertical Functions

1 Introduction

The 5GINFIRE project goal is to provide an 5G oriented experimental facility where different actors of the 5G ecosystem (startups, industry, SDOs, operators, academia, research institutes) can experimentally contribute to 5G development and deployment. The work package "5G Infrastructure Integration and Experimentation Enablement" (WP5) is responsible to put together all the efforts inside the project and demonstrate the enablement of the 5GINFIRE infrastructure to experimentation of Experimental Vertical Instances (EVIs), a composition of several virtual functions spanning all layers from application and services to networking with focus on 5G verticals.

1.1 Objective of this document

The objective of this document is to report on the 5G testbed integration efforts and actions performed. The document also presents a generic guide and process that 5G testbed providers can follow to enable their 5G infrastructure for experimentation and a guide about how experimenters can interact with the 5GINFIRE platform.

1.2 Structure of this document

This document is organized as follows: Section 2 details the status of the automotive and smart city Experimental Vertical Instances (EVI) and associated experiments that will be executed using these EVIs. It also presents a new EVI, belonging to the e-health vertical, that was accepted through the first 5GINFIRE open call. Section 3 shows the ongoing FIRE integration efforts and the new FIRE testbed providers that will be integrated into the 5GINGIRE ecosystem. Section 4 presents a high-level description of how new testbed providers can incorporate a new facility to the 5GINFIRE ecosystem. Section 5 shows how an experimenter can interact with the 5GINFIRE portal and presents a set of Virtual Network and Vertical Functions (VxF) already available. Finally, Section 6 concludes this document.

2 5GINFIRE Experimental Vertical Environments

This section presents the two central Experimental Vertical Instances (EVI) that are under deployment in the 5GINFIRE: the Automotive vertical at Instituto de Telecomunicações de Aveiro (IT-Aveiro) facility and the Smart City vertical at University of Bristol (UNIVBRIS) facility. These two EVIs designed and under deployment act as a blueprint and demonstrate the 5GINFIRE infrastructure integration and experimentation enablement.

The Car overtaking use case was deployed and tested at ITAv and smart city safety at UNIVBRIS. The VxFs pertinent to each uses case were deployed and instantiated through 5GINFIRE Portal enabling the validation of overall 5GINFIRE ecosystem/platform and achieving milestone MS5 (Initial deployment and iteration cycle completed after first open calls wave, experimental facilities operation is running).

These two uses cases serve as a template and motivation for the open calls. In this direction, this section also describes new experiments and new facilities selected from the first open call.

2.1 Automotive EVI Environment

The automotive EVI environment available in IT-Aveiro, Portugal, is depicted in Figure 1From the vehicular network perspective, it consists of several On-Board Units (OBUs), placed in vehicles, as well as Roadside Units (RSUs). An OBU can connect with other OBUs through IEEE 802.11p/WAVE links, and with RSUs via IEEE 802.11p/WAVE, IEEE 802.11g/Wi-Fi or cellular links supported by the C-RAN concept working under the 4G/LTE specifications, frequency band 7. For more details in the vehicular network communications one should refer to deliverable D2.1, Section 3.3.1.

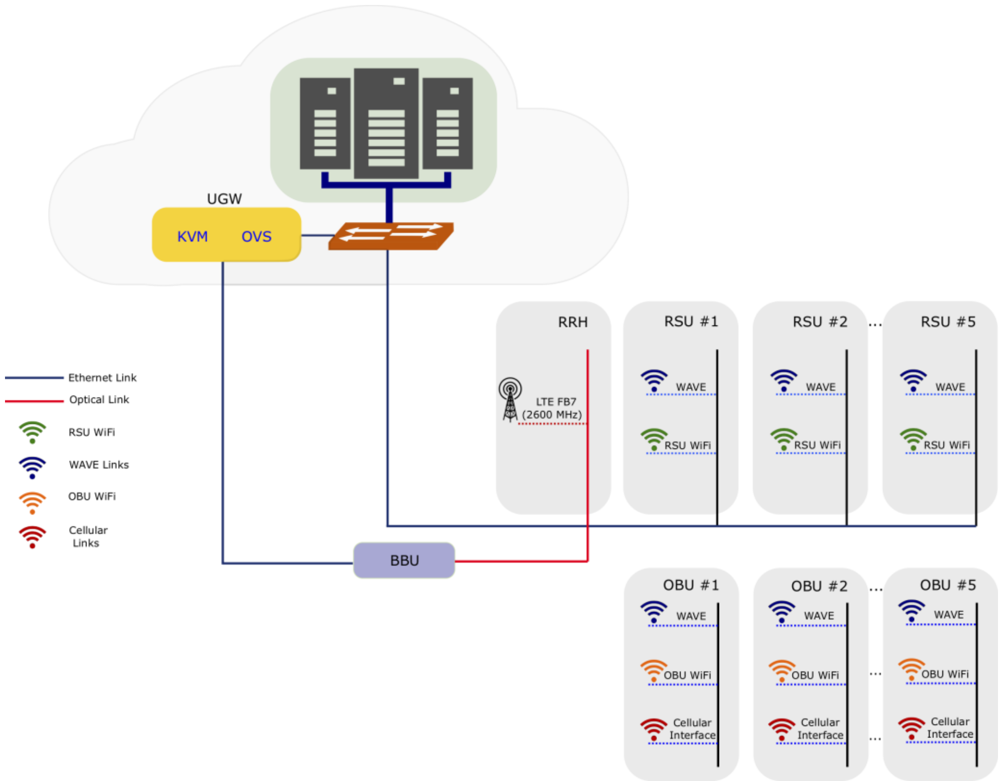

Figure 1 - IT-Aveiro 5G Automotive Testbed System Architecture.

RSUs are connected to the IT-Aveiro datacenter through Ethernet links. The C-RAN Remote Radio Head (RRH) is connected to the Base Band Unit (BBU) using optical fiber link, and from there to the Unifier Gateway (UGW) in charge to manage authentication, UEs sessions and end to end connectivity, which is available as a set of Virtual Network Functions (VNFs) deployed in the IT-Aveiro infrastructure datacenter on an OpenStack instance. The IT-Aveiro datacenter includes, among other components, the Virtualized Infrastructure Manager (VIM), directly connected to the multi-site orchestration facility managed by OSM MANO deployed in 5TONIC at UC3M, and the Network Function Virtualization Infrastructure (NFVI).

OBUs have access to the vehicular information such as velocity, GPS, and heading. This information can be used by the embedded in-Car Node Processor (in our case, a RaspberryPi) to take local decisions, but also be advertised to the other vehicles. Each vehicle has also access to information from the street and surroundings through embedded car video cameras and sensors (crossing roads and traffic lights, cars in the street, adverse conditions in the way, etc.) that will be transmitted using IEEE 802.11g/Wi-Fi to the OBUs. Vehicles may use this information to support a variety of use cases, e.g., assisted driving, autonomous driving, collision avoidance, accident detection, emergency messages dissemination, On-Board Diagnosis (OBD) for car self-repairing (when integrated with the OBU), etc.

To demonstrate the potential of the 5GINFIRE automotive testbed located in IT-Aveiro and to validate the workflow for new experimenters to deploy new VNFs, a VNF video transcoding camera-based car overtaking scenario was proposed. In this scenario, illustrated in Figure 2, each vehicle contains an OBU that provides the communication between vehicles and between each vehicle and the infrastructure. The OBU is also connected to an Android device, which can be a smartphone or a tablet, through Wi-Fi, providing visual information for the driver. The vehicle contains a video camera at its front side. This information will be used by the driver to take decisions on driving, and more specifically in overtaking situations. Regarding the communication interfaces, each OBU can communicate with the infrastructure (RSUs) using the IEEE 802.11p/WAVE or IEEE 802.11g/Wi-Fi or using the 4G LTE cellular technology (C-RAN). The VNF video transcoding is available on site and can be deployed at the edge of the infrastructure.

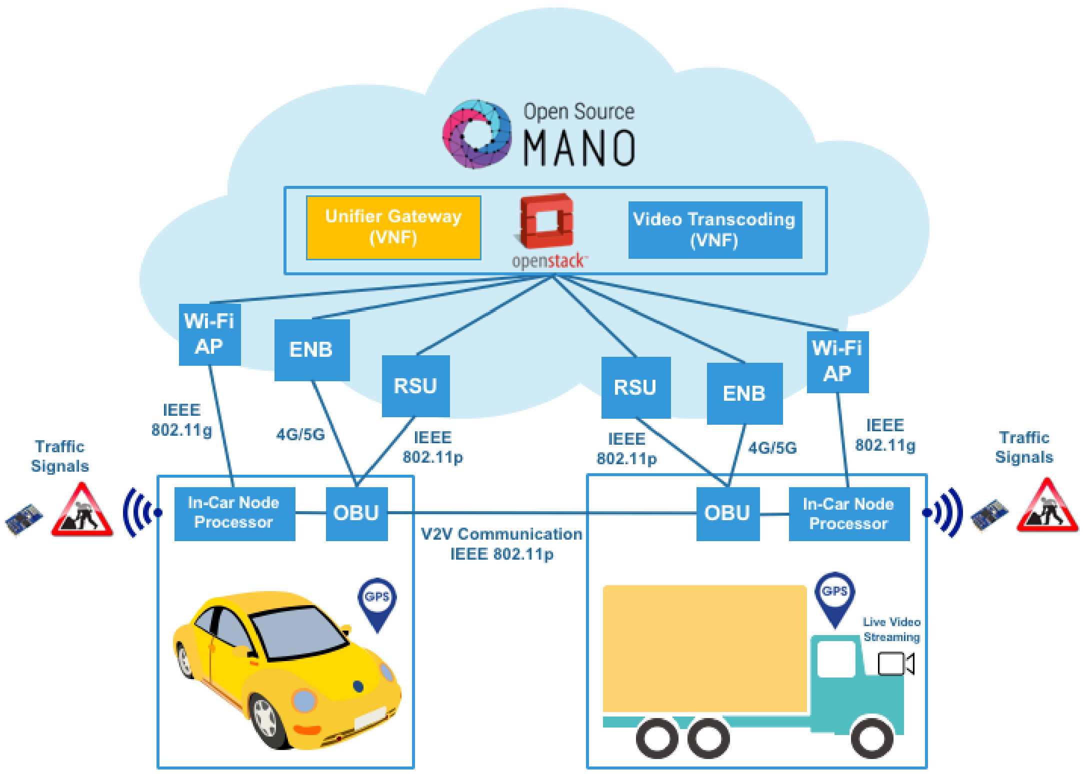

Figure 2 - 5GINFIRE automotive EVI: car overtaking scenario.

During the use case validation, we were able to stream a live video from the front vehicle to the RSU through IEEE 802.11p/WAVE, being transcoded by the video transcoding VNF located at the edge of the infrastructure, and then being transmitted again to the rear vehicle using the same communication technology. Another version of this use case, still waiting for validation, aims to cope with situations where the vehicles are not in range of RSUs, and the video is transmitted through 4G to the C-RAN at the edge of the infrastructure, where the VNF video transcoding is located. For this scenario to be evaluated, another VNF needs to be deployed and integrated with OSM for automatic deployment, which is the Unifier Gateway (UGW).

Summarizing, the automotive EVI environment will grant access to real OBUs, RSUs, additional sensing devices and video cameras, also having the possibility to create and deploy additional VNFs through the 5GINFIRE portal within the IT-Aveiro automotive testbed. Experimenters will have access to a controlled environment in the lab, with the possibility to evaluate and validate their own automotive VNFs components and services such as Vehicle-to-Everything (V2X) communication performance and metrics (e.g., latency vs overhead, throughput vs packet loss, etc.) and test their own automotive VNFs that make use of a diversity of contextual-aware information gathered from extra sensors (traffic signals) or from OBUs internal sensors available.

2.1.1 Overview of Automotive EVI Related Experiments

In the first phase of the first open call, three automotive EVI related experiments were accepted. The ***SURROGATES*** experiment, under the responsibility of the University of Murcia, will use the VIM platform to virtualize instances of OBUs at the edge of the 5GINFIRE automotive testbed to foster 5G vehicular services. This way, real OBUs will collect sensory information and send it, through IEEE 802.11p and cellular communication links, to virtual OBUs that will be responsible for the pre-processing, aggregation and pattern recognition tasks. This experiment will explore the automation process of the 5GINFIRE architecture when creating and managing the virtualized OBUs, while at the same time validating a number of essential components of the 5GINFIRE ecosystem, such as IT-Av OBUs, RSUs, VNF Unifier Gateway, VIM, OSM and 5TONIC VIM, OSM and computing nodes.

The ***VRU-Safe*** experiment, to be held by the University of Athens, will explore the IT-Av automotive testbed and NFV infrastructure to deploy and evaluate an effective solution for prediction and avoidance of vehicular accidents, exploiting a hybrid architecture combining Edge Computing and Cloud-RAN. The idea is to virtualize (in the edge/cloud) modules responsible for the Trajectory Computation and Hazard Identification, so they can identify and predict potential imminent road hazards involving Vulnerable Road Users (VRUs) and automotive OBUs. This experiment will evaluate the performance and reliability of 5GINFIRE infrastructure components, including the Cloud and Edge components, the OpenFlow/OpenDaylight SDN frameworks, and the provided VNFs.

Finally, the ***CAVICO*** project under the responsibility of ITTI, will try to push the boundaries of the QoE-aware adaptive streaming by introducing the context-aware mechanisms that can use no-reference, context-based QoE metric subsets to allow a single system to cope with many use case scenarios, e.g. to remotely operating automotive cars. This way, the experimenters will develop a video controller as a new VxF dedicated to optimize video streaming in the function of quality parameters read from network radio interfaces and video streams, under a 5G automotive scenario. This VxF will be a separate module or a functional extension of existing VNF video transcoding module running in the cloud edge computing controlled by OSM. The CAVICO project will evaluate the 5GINFIRE building blocks based on experimentations conducted during the project while assessing the flexibility and potential of existing VxFs and the CAVICO one.

More details regarding the aforementioned experiments are provided in the following sections.

2.1.2 Virtual OBUs and Hybrid Communications to Foster 5G Vehicular Services

In the SURROGATES experiment the ITAv OBU capabilities regarding network mobility will be exploited to offer 11p/4G hybrid communications as a base enabler to offload processing tasks from the same unit. This is carried out by virtualizing OBU instances at the edge of the 5GINFIRE cloud. A base monitoring system considering sensor data from the vehicle will be developed to send raw information to an OBU virtual instance. Here, pre-processing, aggregation and pattern recognition tasks will be carried out. Processed information will be available for global Cooperative Intelligent Transport System (C-ITS) services.

In the multi-homed scenario to consider, the OBU will select the most convenient interface while seamless handovers are performed. This idea is illustrated in Figure 3. The vehicular communication stack includes a set of IPv6 technologies to support network mobility and hybrid communications. A network mobility server, or a set of them, will be deployed on the IT-Av cloud domain. Network Mobility Basic Support (NEMO) or Proxy Mobile IP (PMIP) will be used to support IPv6 mobility. Security provision will be also considered including the establishment of security associations with IKEv2, and the creation of secure data channels with IPsec.

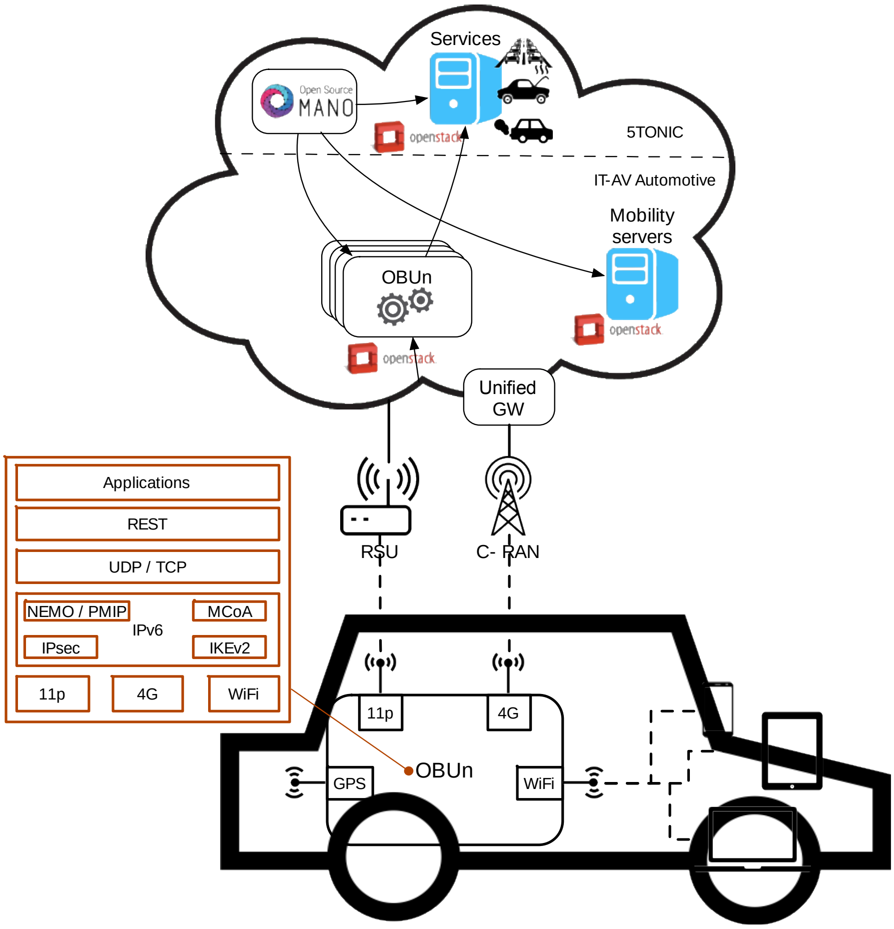

Figure 3 - SURROGATES Experiments.

The objective of the previous communication channel is offloading of data analytics tasks from OBUs, which should focus on actions of higher priority such as maintaining vehicle connectivity, managing communication flows, or applying security measures to data traffic. As it can be seen in Figure 3, the approach chosen in SURROGATES is to create virtualized images of OBUs near the edge of the network, where to process raw data sent from in-vehicle sensors. Hence, a monitoring middleware is considered for the OBU to continuously gather raw data that will be sent to the data processing functions delegated into the cloud-edge. Data analytics tasks envisaged, include pre-processing, aggregation and pattern recognition. Processed or cached information is accessed by vehicular services requiring a global view of the road. This solves the issue of accessing the physical OBU from multiple services and requiring monitoring data when the OBU is temporally offline.

Data to be collected from the OBUs consider navigation information, diagnosis and mechanical monitoring details of the car, and readings from other sensors connected to the OBU.

The NFV-based OBU approach is exploited through services providing the next functions:

- Tracking vehicles to further study driving patterns, above all in urban mobility scenarios.
- Detect malfunctions in the vehicle or provide the capability to predict future mechanical problems.

- Analyze pollution levels of vehicles from both an individual perspective and considering geographic areas.

Services are built on top of the 5GINFIRE network platform, analyzing the possibility of using the computing resources of the 5TONIC facility. OBU instances will be maintained near the edge of the network, at the IT-Av Automotive Testbed. Mobility servers are also deployed here, to improve routing performance.

For the OBU part, the units from the IT-Av automotive testbed are used, including the 802.11p and 4G communication channels. The Wi-Fi link is initially used for in-vehicle connectivity, given its poor performance under mobility conditions. The Unifier Gateway solution is used to reach the IT-Av cloud through the 4G link. Data to be extracted from on-board sensors are initially based on navigation parameters from the OBU GPS and a diagnosis interface (OBD-II). Moreover, additional sensors connected with the OBU are being considered to collect environmental data.

The virtualized OBU images are executed in the IT-Av Automotive cloud, using the OpenStack VIM platform, whereas the operation of the services requires the usage of the 5TONIC. The NFV orchestration is carried out using the global OSM module included also in the 5TONIC deployment.

2.1.3 Vulnerable Road Users Safety (VRU) using a hybrid Cloud RAN and Edge Computing model - VRU-Safe

The main concept of VRU-Safe is related to the Safety of the VRU, and the intelligent and effective prediction and avoidance of such accidents using technologies and concepts, which will be included in 5G, such as Cloud Ran, NFV, Software Defined Networking (SDN), as well as Multiple-Access Edge Computing (MEC). VRU-Safe's primary objective is the experimental evaluation of a network service with computing and networking capabilities, which operates in a distributed manner utilizing both MEC, as well as Cloud RAN capabilities, depending on the location of the involved OBUs and VRUs.

The proposed mechanism can identify Connected Vehicles and VRUs in potentially dangerous situations based on several contexts (Vehicle & VRU position, movement direction, accelerations, etc.), by predicting their trajectories and forwarding the respective notifications to both sides (Vehicle & VRU) to avoid a potential accident.

VRU-safe shall be evaluated in the context of the ITAv testbed, exploiting the testbed's infrastructure. More specifically, the primary system components and their relation to the testbed are described below:

- The Trajectory Computing Component (TCC), which resides at the edge of the network (both at the base station/AP and the Cloud RAN). TCC receives via the RSUs context information (signal strength, location and other sensor information, such as acceleration information, etc.) from the OBUs and the VRUs' smartphones and computes the trajectories of the moving entities. The information, which is transmitted by the OBUs and the VRUs to the RSU will be realized either via the IEEE 802.11 a/b/g/n Wi-Fi interface, or the 4G interface, depending on the testbed's guidelines.
- The Hazard Identification and Notification Service (HINS), which will receive the computed trajectories from TCC, and -based on specific, pre-defined events-, will

identify potential imminent collision hazards and generate the respective alerts, which will be forwarded back towards the VRUs and OBUs, via the RSUs.

- The VRU-Safe Controller (VRU-C) component, which will be responsible for managing the hybrid operation and control whether and which part of the back-end processing (trajectory computing, etc.) should take place at the C-RAN or the base station/AP's side, towards the minimization of the end-to-end delay.

2.1.4 Context-Aware Video Controller for autonomous transport and security monitoring - CAVICO

Normally "autonomous cars" are performing their tasks autonomously, but there are situations where a car controller becomes malfunctioned and, does not "know" how to perform the next step due to ambiguous environment state. Therefore, there must be a driver on board to take control of the car in such cases. However, from a business point of view, the need to assign human driver to such autonomous car seems a waste of resources. That is why such a human driver located in remote office could oversee the operation of multiple vehicles. The CAVICO project aims to provide such technology enabler for video. In such a scenario video monitoring would play a significant role to provide a valuable insight into the car's neighbourhood for a remote operator who would be able to assess situation (thanks to a video monitoring) and take a proper action where autonomous car could not (e.g. taking a remote control over a vehicle or sending to it instruction what to do next). The CAVICO context-aware video controller aims to be an important enabler for remote, mobile video security monitoring.

The main objectives of the CAVICO project are as follows:

- to adapt CAVICO context-aware video controller as a VxF for automotive EVI environment offered by ITAv in 5GINFIRE project,
- to test innovative QoE features of the CAVICO solution,
- to merge QoE and QoS parameters for controlling CAVICO video controller.

This will be achieved by testing campaign to devise, refine and test the best set of quality metrics for CAVICO video controller in mobile settings that address different use-cases. QoE parameters like blockiness, blur, flickering, freezing, etc. allow for an objective prediction of probable user feelings, with usage of automated scripts, without need of human interaction. With a combination of these and QoS parameters, including jitter, end-to-end delay, throughput and losses, as well as, a fact of correlation between QoS parameters and QoE metrics, CAVICO will be able to assess problems or requirements, calculate a synthetic overall quality indicator and then adjust video coding accordingly. QoS metrics will be monitored to sense any instant and averaged fluctuations. Such predictions will be mapped onto QoE metrics to support their fitness to real parameters whose estimations can be degraded by fast changing transmission channel. It will result in more precise control of the video coder. A set of QoS measures can be seen as a vector of weights that facilitate the evaluation of QoE measures.

The CAVICO video controller will be developed as a new VxF dedicated to optimizing video streaming regarding to quality parameters read from network radio interfaces and video streams. This VxF can be a separate software module or a functional extension of existing

VNF video transcoding module running in the cloud edge computing controlled by MANO. Figure 4 shows generic CAVICO solution layout that will be used in the tests.

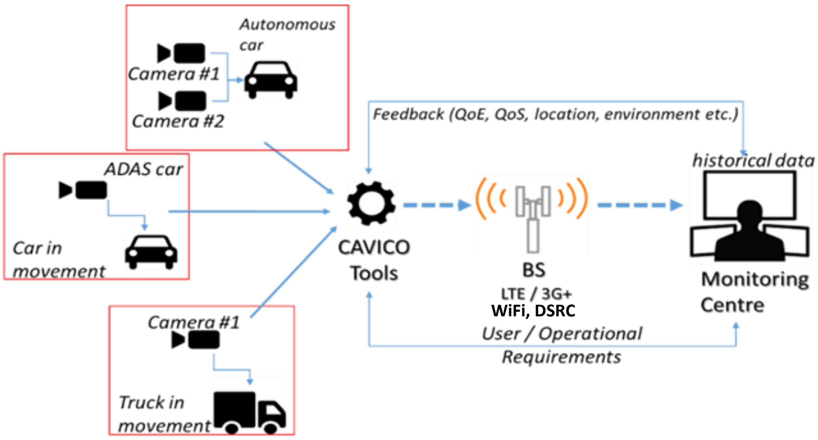

Figure 4: Generic CAVICO layout.

The developed VNF will merge information about QoS and QoE parameters as well as use-cases of autonomous car's operational mode based on Equation 1.

$$VNF_{quality}^{mode} = f^{mode}(QoS, QoE).$$

Equation 1

The CAVICO system will be deployed in IT-Av's testbed (See Figure 5). There will be a video camera with its controller installed on board of the vehicle. The camera controller will be a new In-Car Node Processor (ICNP). INCP will be implemented in Raspberry Pi board. It will control a GoPro camera or Raspberry Pi one. The camera controller will be then connected via on-board Wi-Fi network to OBU to have access to all three-access networks (Wi-Fi, WAVE, 3GPP). The CAVICO video controller as a VNF in the Unifier Gateway component will be a VNF running in OpenStack cloud infrastructure and controlled by OSM.

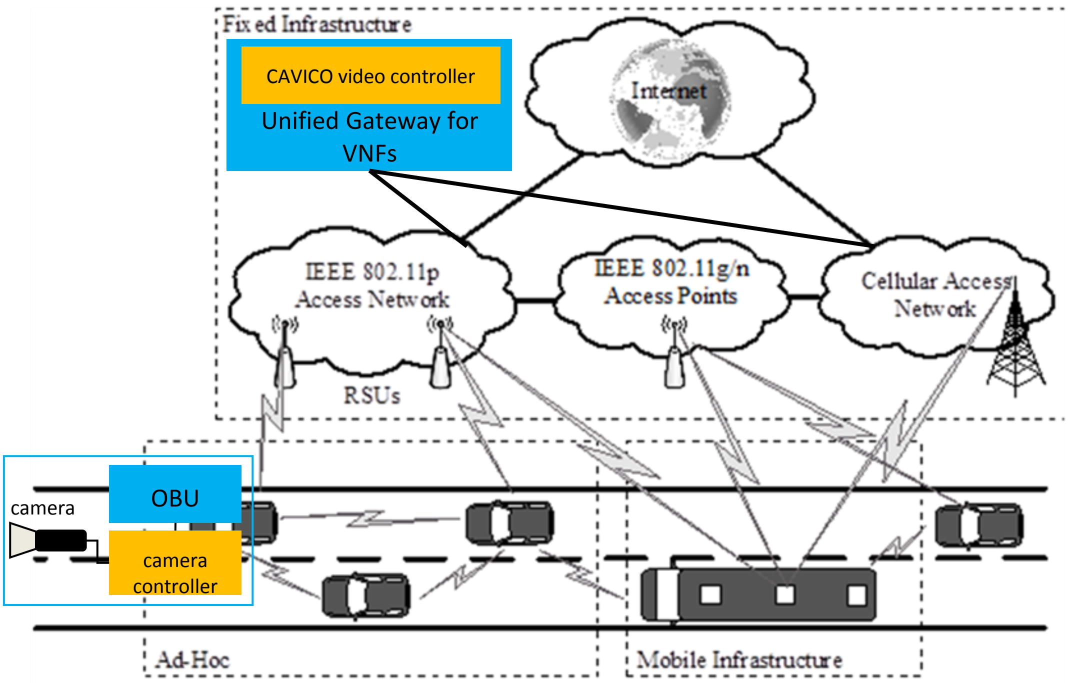

Figure 5 - CAVICO Experiment Overview at ITAV 5GINFIRE Facility.

We plan to use two OBUs. The smart switching among three networks will be supported by the OBU that uses AHP, Network Mobility (NEMO) Enabled Proxy Mobile IPv6 (N-PMIPv6) protocol (it handles multi-hop, hand-over and IPv4 seamless communication) and multi-homing (it handles simultaneous transmissions over many networks and the network selection based on QoS; it is managed to change the channel coding if the transmission is vulnerable to low quality wireless connections). The routing is powered by NSI that are broadcasted within VSAs.

2.2 Smart City Safety EVI Environment

Since its beginning, the debate on the smart city has waved between a vertical approach and a holistic approach to cities. The vertical approach looks at the smart city project with a focus on solving specific city problems such as traffic, security and pollution. Several projects on smart parking and pollution monitoring can be seen in that way. By contrast, the holistic approach looks at the city as a system of systems. In this direction, 5GINFIRE includes the smart city vertical as a platform that can be defined as a framework for sensing, for communications, for integration, and for intelligent decision making.

There is no fully smart city yet in existence. Taking this into account 5GINFIRE is providing an experimental ecosystem to test and experiment with smart city systems. University of Bristol has deployed through the 5GUK testbed[1] an environment that enables experimenting smart city use cases. The ultimate aim of a smart city is an entity that will be sensing to all environmental stimuli, with the ability to connect all these up and respond like a living

[1] http://www.bristol.ac.uk/engineering/research/smart/projects/uk-5g/

organism. The response will take the form of delivering up to the minute information and services that citizens need, based on the information received. By providing on-line information in real time, cities can optimize the use of resources for various operations e.g. parking, traffic management, security, lighting, all with the aim of improving efficiencies and reducing costs.

Figure 6 shows the Bristol 5G testbed system architecture. All the resources that enable experimenting with cloud, edge computing and network function virtualization is available at Millennium Square at Bristol city center. The Bristol datacenter is directly connected to the multi-site orchestration managed by OSM MANO deployed in 5TONIC at UC3M.

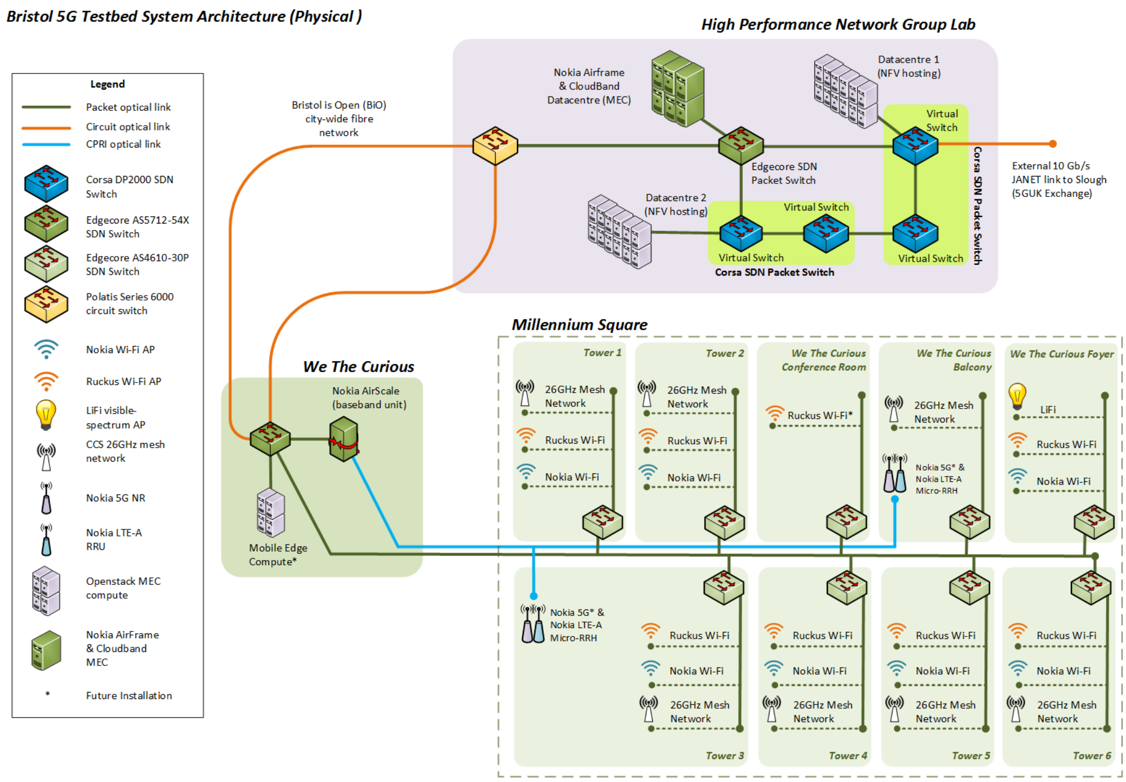

Figure 6 - Bristol 5G Testbed System Architecture.

The University of Bristol Smart Internet Lab 5G testbed is a multi-site solution connected through a 10km fiber ring with several active switching nodes. The core network is located at the High-Performance Network (HPN) research group laboratory at the University of Bristol with access technologies located in Millennium Square for outdoor coverage and "We The Curious" science museum for indoor coverage. Connectivity terminates via fiber optic at distinct points in the square with onward links through the use of fixed wireless access mmWave radios. To allow for future expansion, termination locations have been over engineered with enough installed fiber and power capacity to allow the next generation of 5G connectivity. Space and power for MEC has been provisioned at key locations around Millennium Square to allow the deployment of VNF and low latency real-time application processing close to the end user.

To demonstrate the capabilities and functionalities of Bristol testbed a smart city safety use case has been designed and deployed. The main goal of this use case is not just to provide a smart city but goes beyond and provide a safe environment for the citizen. In addition, combine the three folders: NFV, machine learning and mobile edge computing enabling lower latency and end-to-end communication. Figure 7 shows the smart city safety system architecture. Note that the 5GINFIRE ecosystem includes the multi-site orchestration as depicted in this architecture. However, we focus on Bristol facility where the use case has been deployed. The main building block of the system is the smart city safety demo that is composed of a Rasberry PI Model B, a 360-degree camera Ricoh Theta V, a battery which are attached to a bike helmet. A total of three bike helmets have been set.

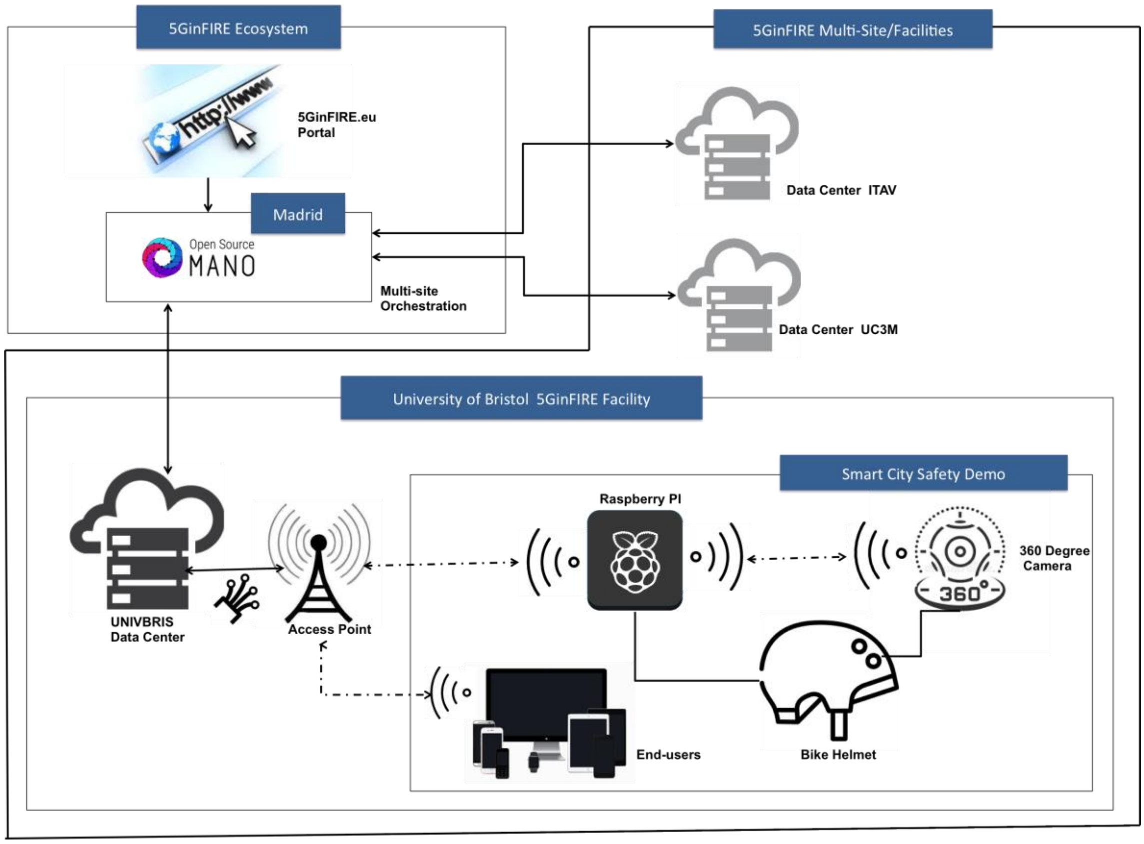

Figure 7 - Smart city safety system architecture.

As a proof of concept, the smart city safety use case looks for detecting suspicious activities along of the city. To this end, face detection and face recognition procedures has been implemented to enable identification of criminals in the city and automatically report their location to the security.

The smart city safety system works as follows. The 360-degree camera captures the live stream video in a 360-degree format. The live stream video is sent to the Raspberry PI via Wi-Fi 2.4Ghz. The Raspberry PI communicates with the infrastructure via Wi-Fi 5Ghz and sends the video to be processed at the cloud or at the edge of the network. Once the live stream video is in the datacenter (cloud or edge) two processing takes place. First the video transcoding, which is responsible for converting the live stream video from spherical to rectangular format. Second, the face detection and face recognition procedure that use the live stream video in a rectangular format to recognize people faces. The video transcoder function has been virtualized and then deployed in the 5GINFIRE ecosystem as a VNF Video Transcoder. This VNF video transcoder together with the face detection and recognition programs are located in virtual machine enabling the processing to be performed any point of the network.

2.2.1 Overview of Smart City EVI Related Experiments

Figure 8 shows the overall smart city safety use case experimentation workflow using the 5GINFIRE ecosystem. The experimentation starts by submitting the Network Service Descriptor (NSD) and Virtual Network Function Descriptor (VNFD) video transcoder trough the 5GINFIRE Portal and specifying Bristol Testbed as the VNF placement.

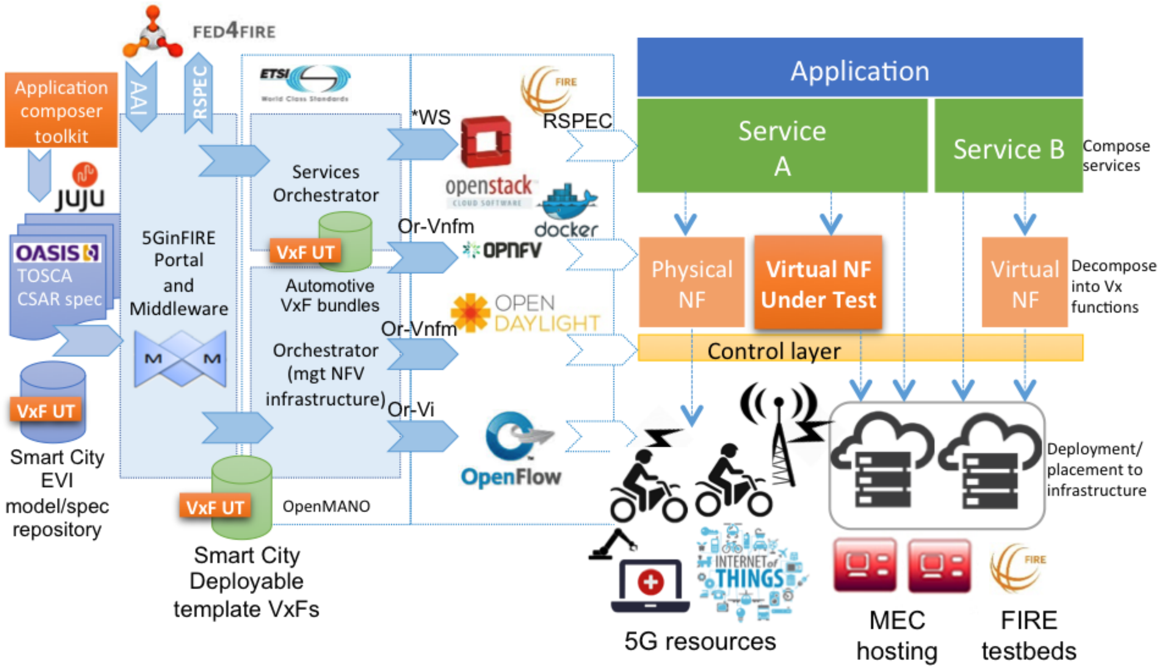

Figure 8 - Smart city safety use case experimentation workflow.

The experiment was performed in both indoor (at Smart Internet Laboratory) and outdoor (in Millennium Square at Bristol - UK). Figure 9 shows the smart city safety demonstration setup in the Millennium Square during the Layered Reality Weekend. Basically, these two experiments (indoor and outdoor) validate milestone MS5 where the experiment has been deployed and run on top of the 5GINFIRE ecosystem using OSM MANO orchestrator through 5GINFIRE Portal and the UNIVBRIS Facility.

Next sections describe the two experiments that have been selected from the open call to use the smart city EVI. The first experiment, named RobotView5G, focus on the development of a new VNF for Quality of Experience (QoE) for measurement of video stream. The second, SFCLola consists of Service Function Chaining (SFC) that operates across different datacenters. It looks for evaluating the latency among different VNFs.

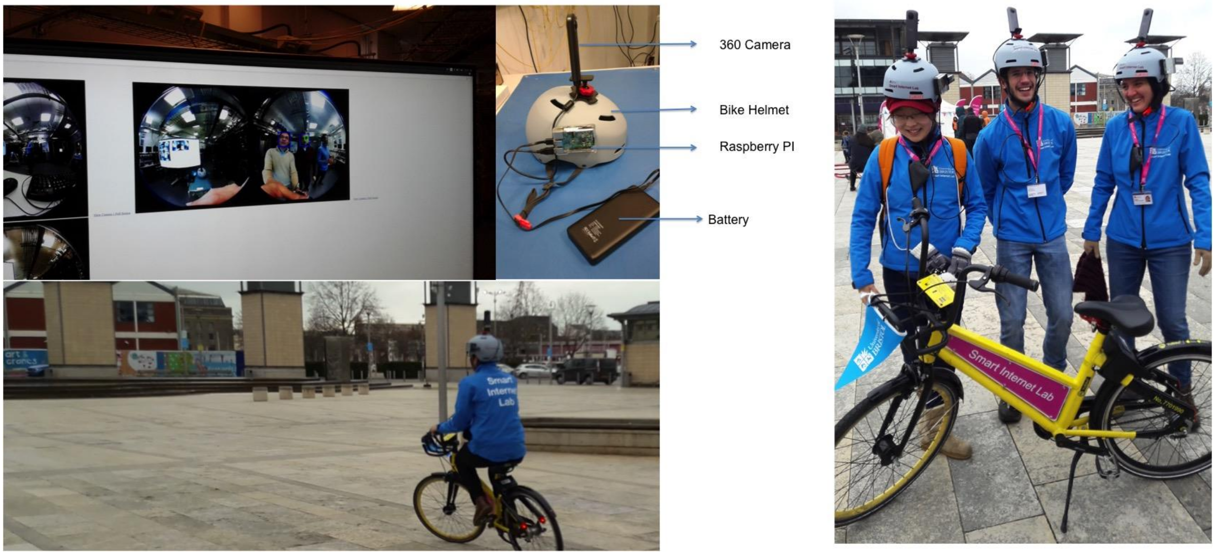

Figure 9 - Smart city safety demonstration setup.

2.2.1.1 5G Smart City Robotic Surveillance Platform - RobotView5G

NETICTECH is developing the Wireless Robotic Surveillance Platform “RobotView”, enabling real-time video surveillance using Wi-Fi and mobile networks from remote controlled robots, drones and other video-monitoring devices. The platform is designed for police, security and rescue forces to enable surveillance and operation in places hazardous to humans. In times of higher security risks, RobotView can save lives of officers, by allowing them to send a robot or drone and remotely watch the video from the installed cameras in real-time. RobotView is not bound to a specific solution, it is a vendor agnostic system, allowing to transmit video coming from cameras installed on robots, drones or vehicles using wireless networks. The system has adaptation mechanism to react to changing network parameters and maintain the needed video quality. This is done using automatic video quality analysis tools, that based on no-reference QoE measurement algorithms, provide in real-time a set of QoE parameters. The system is in the testing stage now and in the experiment we want to test our encoding and transmission parameters in order to tune them for usage in various video-monitoring scenarios. Figure 10 shows the general architecture of the RobotView platform.

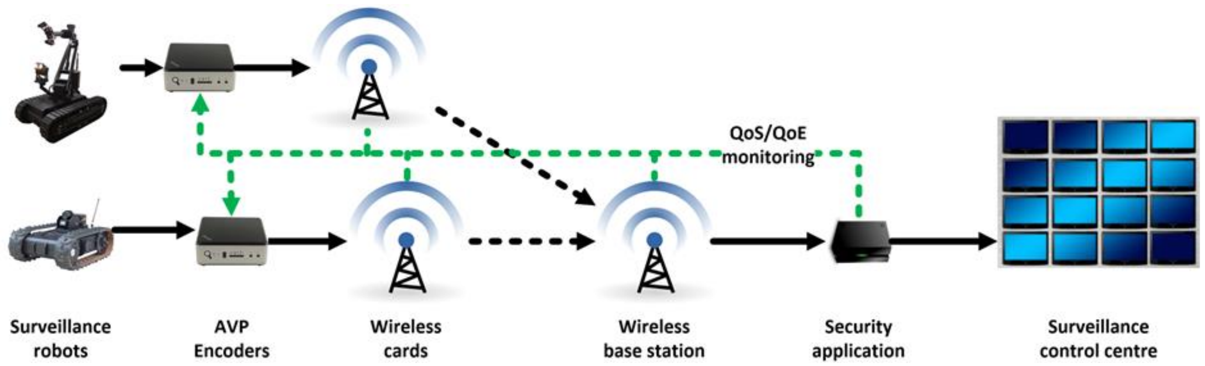

Figure 10 - General architecture of the RobotView platform.

The heart of the RobotView platform is the Adaptive Video Processor (AVP). The AVP can change the encoding and transmission parameters of video. The Controller of the AVP receives QoE-related information, the status of the network link, availability of local resources for video encoding and user input from the Security Application. Based on this data it decides which encoding and transmission profiles the AVP should use to enhance the QoE and thus the usability of the surveillance video in the changing conditions of wireless networks. In our current implementation, the central Security Application is providing QoE video analysis capacities, however taking advantage of the 5G technology, we want to distribute the QoE video analysis to the edge nodes.

In the experiment our goals are to deploy our no-reference QoE analysis tool as a VNF, so we can utilize the capacities of edge computing to perform QoE analysis close to the video source. We want to test how edge node processing can decrease the overall bandwidth usage for surveillance videos. We also want perform tests to find correlations between QoS and QoE, as well as define encoding and transmission profiles for various network conditions. Finally, we would like to add to the 5GINFIRE catalogue a new VNF providing real-time QoE measurement for video streams that will be available for other experimenters. Figure 11 shows the architecture of the RobotView development for our experiment.

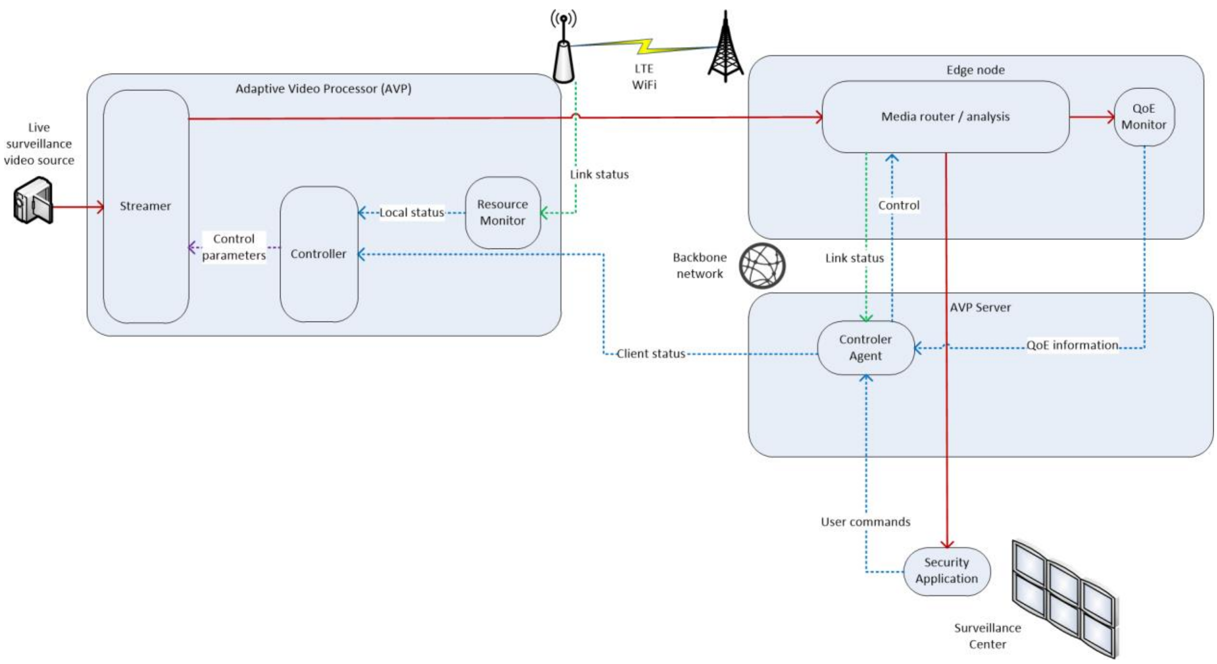

Figure 11 - RobotView architecture for the 5GINFIRE experiment.

For our experiment we plan to use the Smart City Safety testbed and take advantage of the video cameras installed statically and those available on bikers, and the Raspberry Pi computers attached to the cameras. However, as the Raspberry Pi computers do not provide enough computational power for real-time video encoding, for our initial tests we will use a set of prepared video sequences. The sequences will be clips encoded using H.264 codec with various resolution, bitrate, fps and GOP type. We will stream them from the Raspberry Pi computers using wireless networks to the edge nodes. On the edge nodes the QoE analysis tool will provide QoE measurement. We will then correlate these with the QoS parameters of the wireless network. The videos will be streamed also to the security application, where security monitoring personnel will conduct a subjective QoE validation. We will also perform streaming of live video coming from the static and biker's cameras, that will be encoded by the cameras themselves. If possible, we can try to control the encoding mechanism of the camera from the AVP installed on the Raspberry Pi. This however, will only be possible if the cameras have the needed interface and if we have time left in the experiment for additional implementation. Finally, if we have more time left, we will travel to the testbed and install our AVP hardware device, to make further tests using footage coming from the cameras available in the testbed.

2.2.1.2 Service Function Chaining (SFC) orchestration application for low latency guarantees - SFCLola

The goal of SFCLola is to experimentally evaluate a latency-aware service chaining application that dynamically enforces service chains over geographically distributed sites by:

i) optimally selecting VxFs instance over the path that minimizes the offered end-to-end latency across cloud and network resource domains (e.g., service function path 1 in Figure 12);
ii) continuously monitoring that the Service Level Agreement (SLA) is not violated (i.e. the monitored latency is below a maximum latency provided in the request);
iii) if SLAs are violated or a change in user demand or context occurs, the chain is recomputed and updated (e.g. the chain update may consist of establishing a new path, i.e., service function path 2 in Figure 12).

Our reference scenario consists of a multi-site virtualized infrastructure where VxFs (referred as VFs in Figure 12) instances have already been deployed. More specifically, multiple instances of a VxF type have been deployed at different sites.

The latency-aware service chaining application exploits monitoring data that are collected from the underlying cloud and network infrastructure to derive network and computation latency. When a chain has to be established, the SFCLola application interacts with the Virtual Infrastructure Manager (VIM) to forward the appropriate forwarding instructions. SFC and traffic steering programming APIs are needed from the underlying infrastructure to support and implement this latter aspect.

The SFC application can be deployed in a Virtual Machine and thus be positioned in the NFV Architectural framework as a VNF, similar to virtualized SDN Controllers and SDN Applications, as described in the ETSI GS NFV-EVE 005 specification. Therefore, it can be in principle included in 5GINFIRE portfolio. Feasibility of inclusion in 5GINFIRE portfolio will be analysed in detail during the experiment, since the usage of SFCLola implies a set of prerequisites, especially concerning the acquisition of latency measurements from the infrastructure and VxF instances.

This experiment mainly focuses on the Bristol Smart City Safety Testbed since it offers a multisite OpenStack deployment. However, the Automotive testbed and 5TONIC will be also used for deploying a larger scale experimentation.

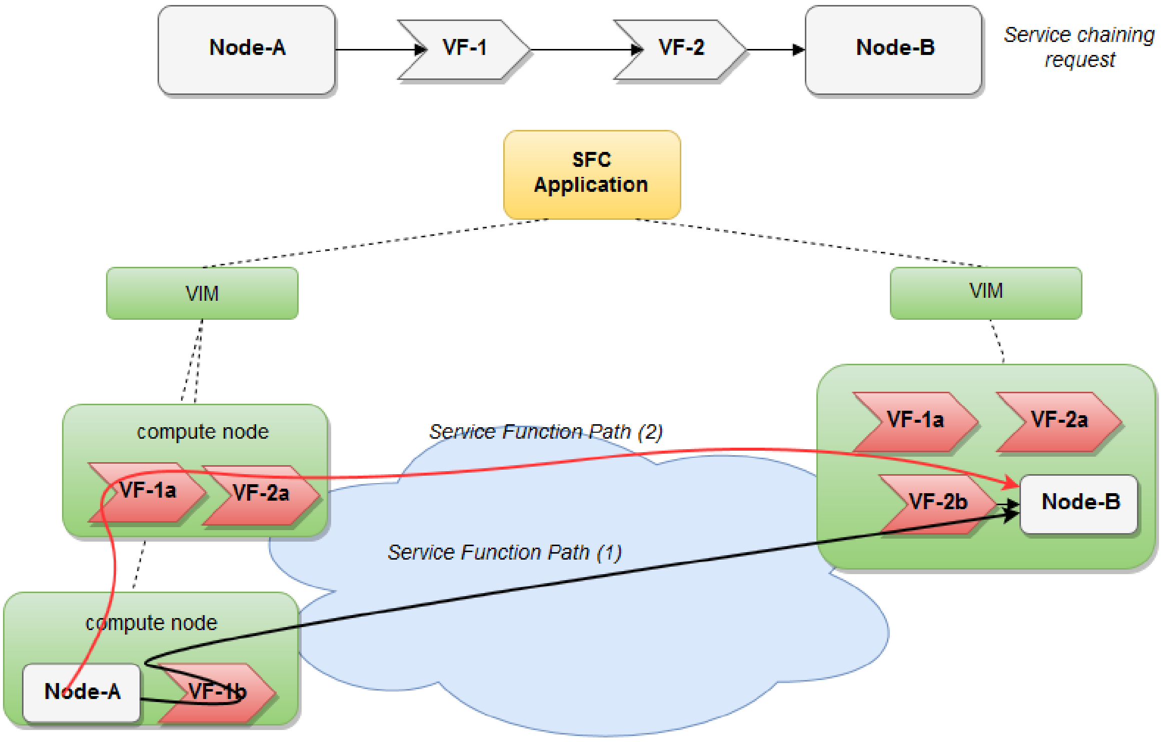

Figure 12 - Example of service chaining request handled by SFCLola.

The experiment aims at evaluating the capability of the SFCLola application to elaborate service chain requests, to compute the latency-optimized VF chains by correctly elaborating monitoring data on processing and network latency and instructing the underlying VIMs for setting-up VF chains.

The experiment methodology consists of the following main steps:

i) deploying our latency-aware SFC application as a VF that performs orchestration tasks on top of a VIM managing multiple compute nodes and gathering possible refactoring requirements to improve the interoperability and integration requirements of our implementation

ii) progressively building an experimental environment where different instances of VFs are deployed at different sites. Emulated VF implementations will be considered for deployment to increase the experiment scale.

iii) testing the SFC application to measure the effective latencies experienced by traffic flows in a smart city safety reference scenario in different load scenarios and the effectiveness of the SFC application in controlling and adjusting the service function chain path through optimal selection of VF instances

The latter point will be realized as follows. Service chaining requests will be dynamically generated through an ad-hoc script and provided as input to the SFC application REST APIs. The use of different request arrival rates (e.g., fixed/variable inter-arrival time intervals) are considered as well as different request profiles (e.g., bandwidth demand, traffic flow type, ordered set of VFs, source and destination nodes). Traffic flows for the received service chain requests will be generated using ad-hoc tools (e.g., iperf).

Monitoring data will be collected both for online and offline use. Monitoring data are used online by the SFC application to obtain both processing latency at VFs and network latency at the VF interconnection links to feed the optimization algorithm (see Figure 13). VNF instances are required to provide their processing time and nominal processing capacity. These metrics can be collected by OpenStack monitoring tools (e.g. Gnocchi/Telemetry) in addition while access to custom metrics is supported. Monitoring data are also used to detect SLA violations (increase of end-to-end latency) or faults and trigger possible recovery actions (i.e., updates of service chain paths). Different rates of monitoring data collection will be also considered to evaluate the trade-off between the orchestration decisions' timeliness and monitoring load on system performance.

Evaluation of SFC application performance is carried out by leveraging logging capabilities of the implemented software and by collecting monitoring data related to specified metrics. The following metrics will be collected and analyzed: execution time (time needed by the application/algorithm to elaborate the request and provide the result); acceptance ratio (the proportion of the total service requests that are accepted by the network); end-to-end latency (the overall latency of the identified path, from a given source to a given destination node, across the selected VFs); VF node performance profile (e.g., load, throughput) and load distribution.

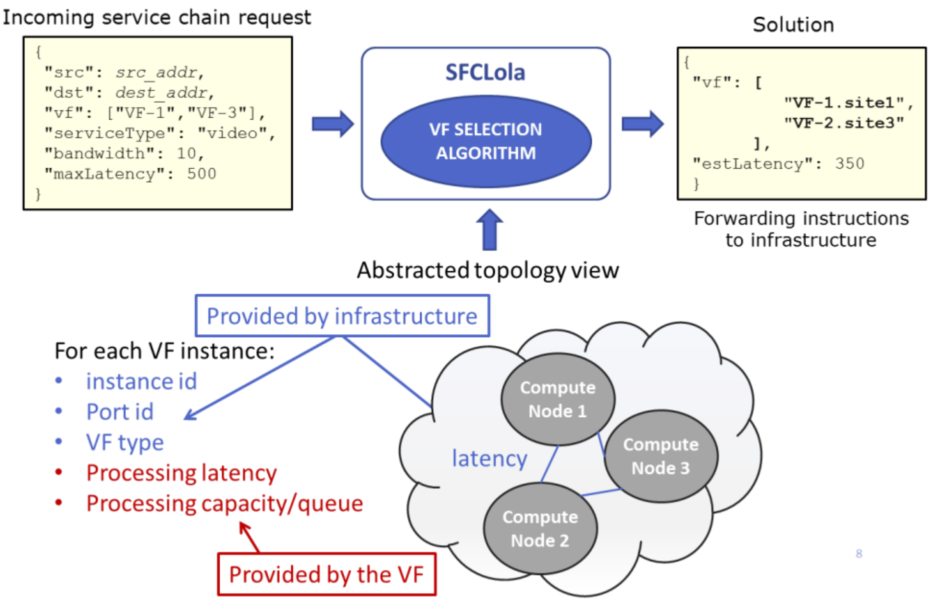

Figure 13 - SFCLola experiment workflow.

SFCLola requires cloud computing and networking testbed resources. Smart City Testbed is targeted since it offers an OpenStack environment managing geographically distributed compute nodes, therefore allowing to test our system in a realistic environment. Traffic steering capabilities exposed as APIs are required.

2.3 eHealth EVI Instance Overview

eHealth5G facility (located in Poznan Supercomputing and Networking Center, Poznan, Poland) is a new addition to the 5GINFIRE testbed offerings accepted through the 1st open call and it will be available for experimenters from December 2018. 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.

2.3.1 Testbed architecture

The eHealth Vertical Industry infrastructure that PSNC makes accessible to experimenters via the 5GINFIRE facilities consists of eHealth devices aggregated into 3 functional groups: the operating room lab, the physiological parameter sensors lab and the patient wellbeing sensor lab & living lab:

a) The operating room laboratory includes very specialized surgical equipment equipped with video cameras.
b) The physiological parameter sensors laboratory contains hi-tech devices used by a medical professional or patient supervised by a medical professional to gather data about the patient's physiological parameters.
c) The patient wellbeing sensor lab & living-lab are low cost devices for the private usage such as wearables and indoor elements.

The eHealth laboratories will be connected via access networks with the PSNC R&D testbed emulating a modern Communication Service Provider (CSP) network attached to Edge and Core Clouds (see Figure 14). The network infrastructure backbone is based on ADVA FSP 3000R7 DWDM optical systems (with 10G/100G client-side interfaces) for both Metro and Core networks. On top of the optical layer, layer 2/3 services are provided by Juniper MX480 nodes controlled by a Juniper NorthStar SDN WAN controller. Edge and Core NFV/Cloud computing is provided by the PSNC SDN laboratory composed of HP Proliant DL380 and IBM

System x3550 M3 rack servers, interconnected using a set of Pica8 and NoviFlow OpenFlow switches, and controlled by two OpenStack instances enabled for the 5GINFIRE OSM.

External access to the facility as well as communication with other 5GINFIRE facilities will be established using VPN gateway. Moreover, high-bandwidth data transmissions with other 5GINFIRE facilities will be available thanks to the GÉANT networking services.

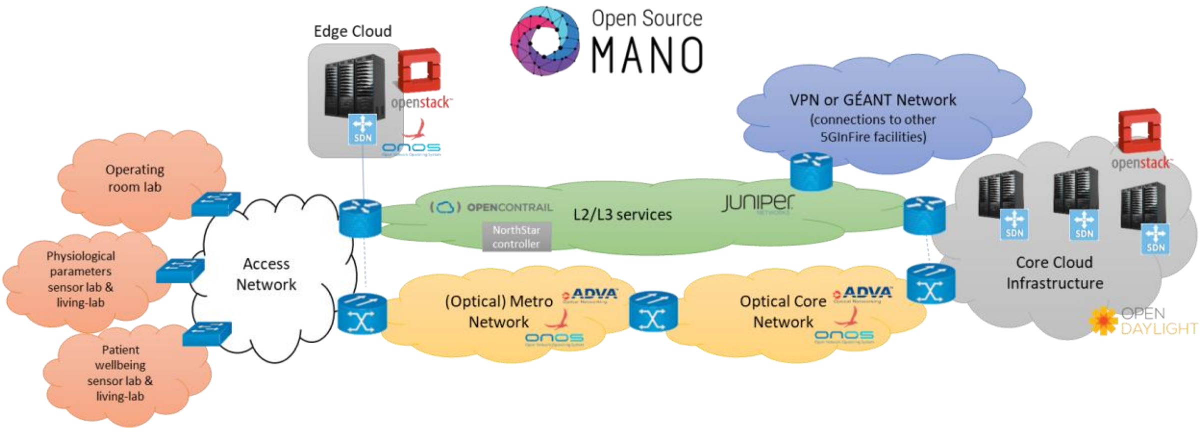

Figure 14 - Overview of eHealth5G infrastructure.

NFV infrastructure hardware details:

- 2x HP ProLiant DL380 Gen 9 compatible with Intel DPDK
- 4x IBM System x3550 M3
- 4x Pica8 P-5101 with OpenFlow 1.3/1.4
- 2x NoviSwitch 2128 - 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 analyzing (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)

2.3.2 eHealth equipment

The goal of eHealth EVI is to enable the 5GINFIRE experimenters to execute various eHealth experimental scenarios within the eHealth EVI, and for this reason, PSNC is providing a broad set of eHealth devices. Most of eHealth devices can be operated only by humans and are not network connected (doesn’t contain LTE or Wi-Fi modules inside) thus medical/sensor data must be manually downloaded from the devices and enabled for 5GINFIRE using some mobile or stationary nodes being part of the labs.

Most of the eHealth devices are part of PSNC's dedicated living lab space, so the sensors can be used by actual end users if a given experiment demands it. PSNC collaborates with a number of organizations and institutions within the city of Poznań, including hospitals as well as NGOs and municipal offices, which may be able to intermediate in contacting end users appropriate for the needs of a specific experiment.

The operating room lab contains cutting-edge endoscope and macroscopic camera, which are sources of medical video streams. This video can be delivered to processing nodes defined by the experimenter in both compressed or uncompressed form. Additionally, the operating room lab contains an H.323/SIP videoconferencing node, so the scenarios may include live audio-video interaction, e.g. for remote medical consultations. Table 1 shows the description of two equipment available in the operating room lab.

Table 1 - The Operating Room Lab Equipment's

EQUIPMENT VIEW	DESCRIPTION
	Storz Vitom Full HD - macroscopic surgical camera being part of high definition imaging systems for research and surgical imaging applications used for microsurgery and microdissection.
	Storz 3D endoscope - contains two Full HD cameras at the tip of the endoscope, representing the left and right eye, processes to a three-dimensional view.

The physiological parameters sensor lab includes a set of hi-tech devices that can be used by a medical professional or, a patient supervised by a medical professional, to gather data about the patient's physiological parameters. This equipment includes a stomatological camera, portable ECG monitor, electronic stethoscope, mobile ultrasound tool, digital podoscope and a baropodometric platform (see Table 2).

Table 2 - Physiological Parameters Sensor Lab Equipment's.

EQUIPMENT VIEW	DESCRIPTION
	VistaCam iX - a dental camera with interchangeable heads for: intraoral images, magnifying images 120x, visualizing caries and plaque, as well as light hardening
	Bittium Faros 360 - a lightweight, portable 3-channel ECG for cardiac monitoring.
	Littmann 3200 - an electronic stethoscope with the ability to listen remotely.
	GE Vscan - a handheld, pocket-sized ultrasound tool that provides real-time black-and-white anatomic and color-coded blood flow images at the touch of a button.

	Podoscan 2D FootCAD - an advanced digital podoscope for digital analysis of footprints and plantar loads. FreeMed - a digital baropodometric platform for static, dynamic and stabilometric analysis

The wellbeing sensor lab & living-lab includes devices meant for acquiring data about the environment of a patient (see Table 3). Those will include various wearable sensors as well as integrated and miniature indoor sensors, activity monitoring devices and cameras.

Table 3 - Wellbeing Sensor Lab & Living-lab Equipment's.

EQUIPMENT VIEW	DESCRIPTION
	CubeSensor - small, connected devices that help maintain a healthy and productive indoor environment by monitoring temperature, humidity, air quality, light, noise, pressure, and movement.
	Wearable sensors - health sensors integrated into various wearable accessories (garments, hats, wrist bands, eyeglasses, wristwatches, smartphones).
	Mother Sen.se - a family of small smart sensors that can be affixed to almost anything to detect and analyze the specific movements of every activity, as well as measure temperature and detect the presence of people or objects at a defined location.

2.3.3 Experimentation

Experimentation activities will be supported in the eHealth5G EVI through the 5GINFIRE portal. The experimenters will be able to:

- reserve eHealth device(s) for ensuring device availability and proper human staff support,
- reserve laboratory to conduct on-site experimentation with own hardware devices,
- create and deploy, from the 5GINFIRE portal, virtual functions at the eHealth5G Edge and Core Cloud, which will process data ingest from eHealth devices,
- deploy virtual machines at the eHealth5G Edge or Core Cloud infrastructure,

- monitor performance metrics of VxF execution and connectivity/traffic monitoring,
- request high-speed network connection (up to 1Gbps) to other 5GINFIRE facility allowing for combining resources and functionalities of different 5GINFIRE facilities,
- accessing experiment results stored in the Cloud.

When the eHealth5G vertical resources will be successfully reserved, the experimenters will also be able to access them physically in PSNC premises in Poznan on specified dates, or PSNC could arrange one or more people who will use the sensors, follow experimenters' instructions and whose health and wellbeing information will be sent for further processing in the Cloud.

3 FIRE INTEGRATION

5GINFIRE, in a nutshell, aims at creating a platform for the experimentation of NFV based services oriented to 5G verticals using industry-leading and open source technologies.

Considering the number of assets already deployed on Future Internet Research and Experimentation (FIRE) testbeds, one of the goals of 5GINFIRE is to reuse these assets by enabling the deployment of the NFV network services atop of the FIRE based facilities.

This section presents the ongoing efforts regarding the integration with FIRE based testbeds. Initially, it shows the integration from a conceptual perspective, and then it presents concrete uses cases with FIESTA-IOT project facilities, NITOS facility at the University of Thessaly (UTH) in Greece and the IRIS testbed at Trinity College Dublin (TCD) in Ireland, a member of the FUTEBOL testbed.

3.1 Integration Between 5GINFIRE and FIRE Testbeds

The concept of an experiment to the 5GNFIRE ecosystem is a network service that is related to 5G vertical such as automotive, e-health or smart city. This network service, called EVI. An EVI has a set of Virtual Vertical Function (VVF), and VNF described as VxF. An NSD represents the complete network service, and the 5GINFIRE platform instantiates and manages the network services using a Management and Orchestration (MANO) entity. After the deployment, by the 5GINFIRE platform, the service is ready to be used by the experimenter.

In FIRE world an experiment encompasses the resource specification, reservation, and provisioning [1]. FED4FIRE compliant testbeds use a Resource Specification (RSPEC) that describes the physical resources such as switches and compute nodes. This description is then consumed by the Aggregate Manager (AM) entity that is responsible for resource reservation and provisioning.

To NFV world the MANO entity interacts with the Virtual Infrastructure Manage (VIM). A VIM, such as OpenStack, is responsible to the management of the virtualized resources such as compute, network and storage. OSM, the MANO entity adopted for the 5GINFIRE ecosystem, supports different VIMs. OSM R3, for example, supports several flavors of OpenStack, OpenVIM, and VMware [2].

To the FIRE world, the physical and virtual resources are managed by the AM, which controls the testbed resources. In this sense the AM acts as the VIM of a FIRE based testbed. In the fire world there are also several AM. For example, the FUTEBOL testbed uses the Cloud Based Testbed Manager (CBTM) [3] and the Open Testbed Control and Management

Framework for OpenStack (O2CMF) [4]. FIBRE in Brazil [5] uses the OFELIA Control Framework (OCF) [6].

One possible approach for the integration between 5GINFIRE and FIRE testbeds would be the exploitation of VxF available at the 5GINFIRE Portal. By using this approach, the 5GINFIRE would act as a catalogue of VxFs that would make available to the FIRE experimenters. In this case, the FIRE Testbed would use the 5GINFIRE Portal API (described in Section 4.1) to connect the portal, consume a VxF a convert this VxF into an RSPEC description. At this moment this integration is ready to be used, and it only requires an effort from the FIRE testbed provider.

The approach for the integration between 5GINFIRE and FIRE compliant testbeds would be the MANO entity, concretely regarding 5GINFIRE, OSM, orchestrate resources from a FIRE based Testbed. Figure 15 presents a mapping between the NFV architectural framework [7] considering 5GINFIRE and a FIRE based testbed. In this case, OSM would interact with the FIRE Testbed VIM (AM). The AM would then interact with the FIRE Testbed resources. The VIM exposes virtualized resource management interfaces and sends virtualized resource management notifications to OSM (using OR-VI and VI-VNFM interfaces), presented in Figure 15.

Using this first approach, the NSD that corresponds to the services and resources of an experiment would be orchestrated by OSM by using the interfaces with the FIRE testbed VIM. This deployment of the services in the FIRE tested would also require a previous configuration of the network connectivity between 5GINFIRE and the FIRE testbed as described in Section 0.

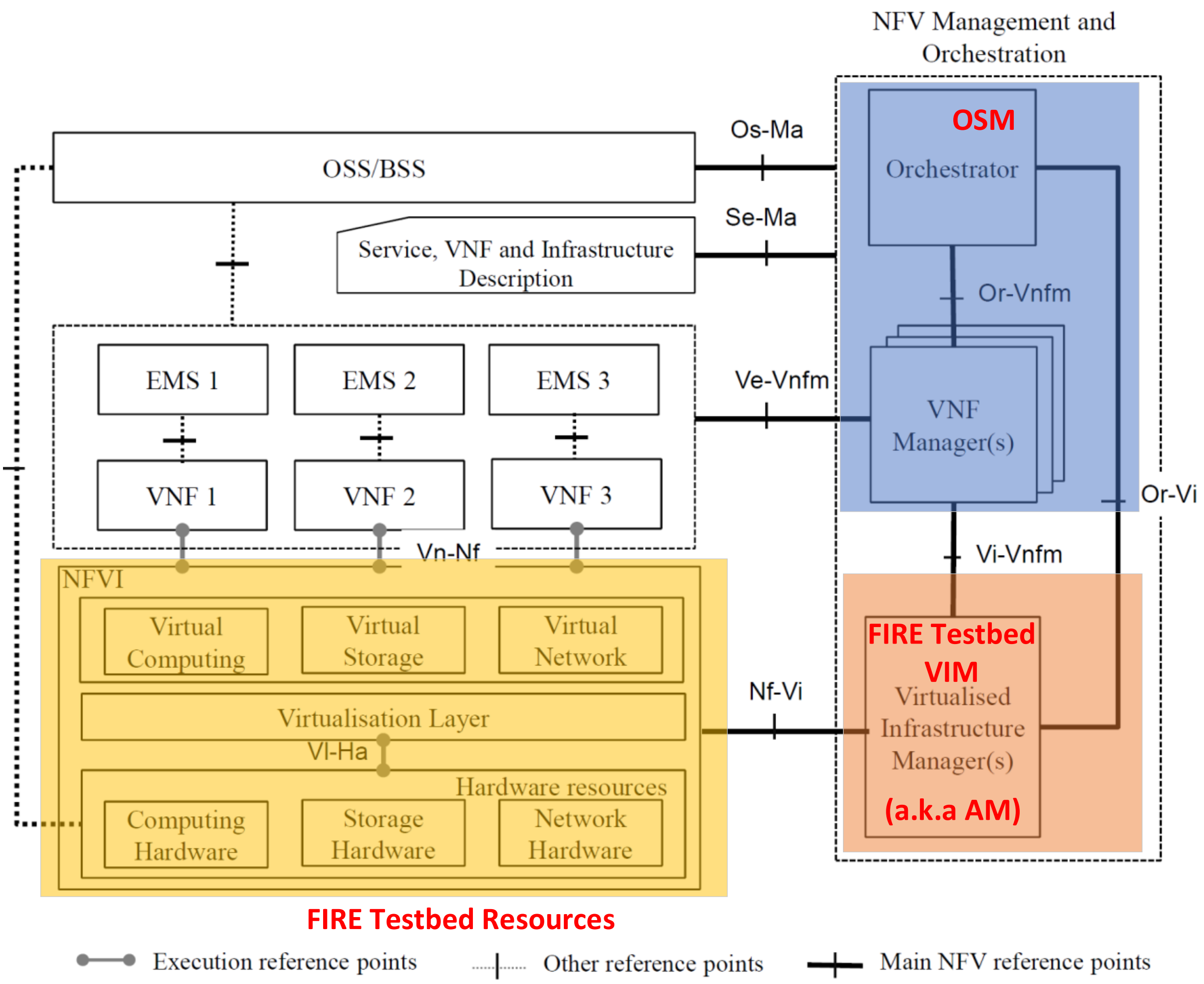

Figure 15 - Conceptual Integration Between 5GINFIRE and FIRE Testbeds.

Considering that each FIRE Testbed has its own entity responsible for the resource management, this first approach of integration requires an integration effort to each new FIRE based testbed. Currently this approach is being used to integrate 5GINFIRE with the FIESTA-IoT (Section 3.2), NITOS testbed (Section 3.3) and the FUTEBOL project facility located in Trinity College Dublin (TCD) (Section 3.4).

3.2 FIESTA-IoT Integration

Federated Interoperable Semantic IoT/cloud Testbeds and Applications (FIESTA-IoT) provides tools, techniques, processes and best practices enabling IoT testbed/platforms operators to interconnect their facilities in an interoperable way based upon cutting-edge semantics-based solutions. The main concepts of the FIESTA-IoT are:

- **Experimentation-as-a-Service (EaaS) paradigm for IoT experiments**: Instead of deploying yet another physical IoT infrastructure, it will enable experimenters to use a single EaaS application program interface (API) for executing experiments over multiple existing IoT testbeds. Experimenters will be therefore able to learn the EaaS API once, and accordingly use it to access data and Resources from any of the underlying testbeds.
- Testbeds participating in the federation will have to implement **the common standardized semantics and interfaces that are being defined within the FIESTA-IoT**

project. This will enable the FIESTA-IoT meta-platform to access their data, resources' and services' descriptions and other low-level capabilities.
FIESTA-IoT meta-platform will be a directory service where resources from multiple testbeds will be registered. In the same way, the observations produced by them will be also stored. This directory will enable the dynamic discovery and use of resources (e.g., sensors, services, etc.) from all the interconnected testbeds.

- Use of **semantic technologies to support the interoperability** between heterogeneous IoT platforms and testbeds. **FIESTA-IoT ontology** has been defined to rule the semantic annotation of the core concepts used within the FIESTA-IoT Meta-Platform

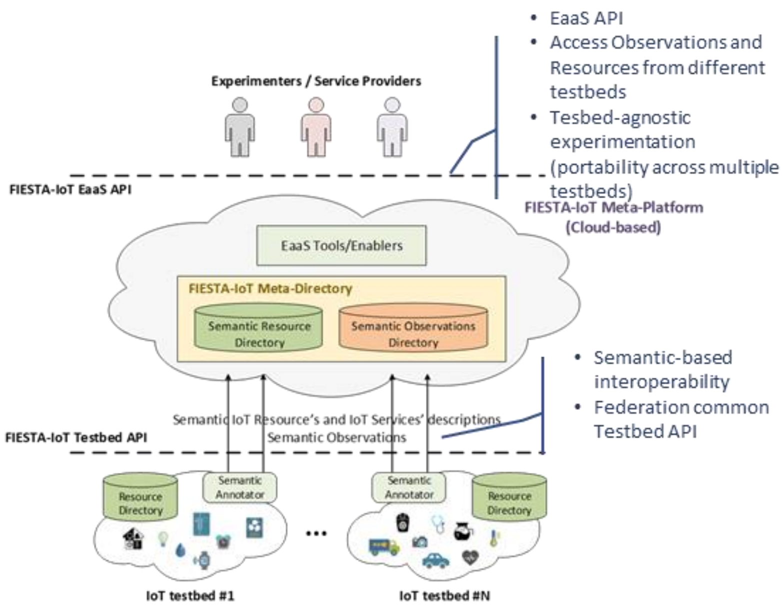

Figure 16 Fiesta-IoT overview.

As the FIESTA-IoT platform is already a federation of several testbeds and offers a set of REST APIs to access to the structured data, the integration to 5GINFIRE federation is proposed to be realized by deploying a VNF which can access the FIESTA-IoT data using the EaaS API upon the semantic query made by the 5GINFIRE experimenter. The following Figure 16 depicts the VNF connector deployed in the 5GINFIRE infrastructure to make the FIESTA-IoT data available for a 5GINFIRE experiment. For example, an experiment "network quality measurement" aims to draw a cartography of the network signal received in a given area, it can formulate a SPARQL query[2] to the VNF "FIESTA-IoT connector" which forwards this query to the FIESTA-IoT platform respecting the Experimentation-As-A-Service (EaaS) API specification, to search for the related measurement available in the FIESTA-IoT semantic data. The returned response from FIESTA-IoT platform is then forwarded towards the experiment by the connector so that the experimenter can proceed for the data analytics to complete the experiment.

[2] https://www.w3.org/TR/rdf-sparql-query/

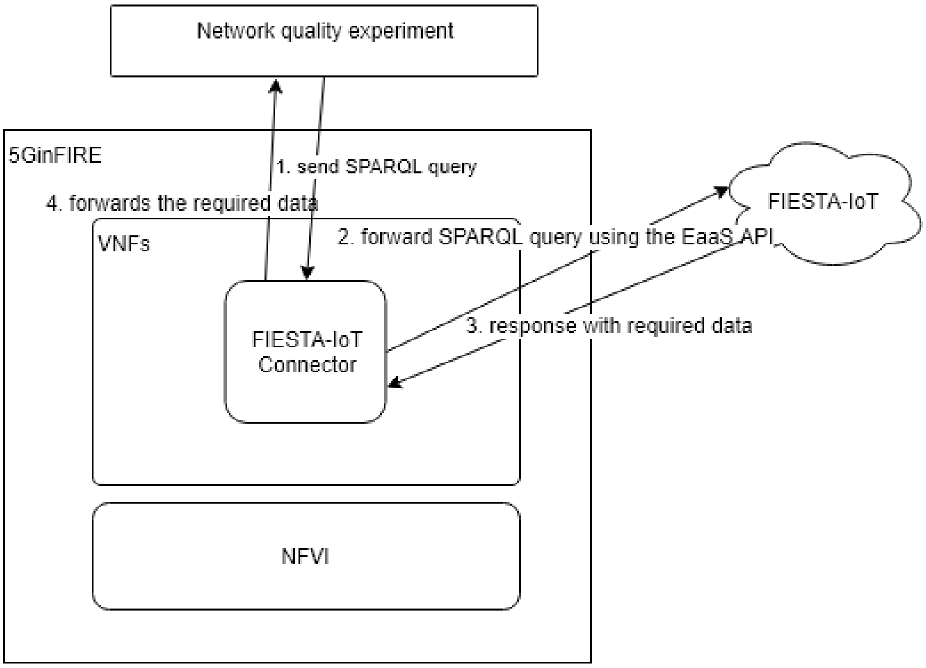

Figure 17 FIESTA-IoT integration with 5GINFIRE by “Fiesta-IoT” connector VNF.

3.3 5G Virtual Infrastructure provisioning over NITOS testbed

Virtual Infrastructure provisioning over NITOS testbed (5G-VINO) aspires to integrate the NITOS [8] testbed with the existing facilities of 5GINFIRE. This integration is not limited to the physical interconnection of the facility with the ones of 5GINFIRE but includes the provisioning of a kit of new NSDs and VNFs specific to the new technologies offered through NITOS. The wide experimentation environment created through the NITOS addition will render the experimental evaluation of multiple use case scenarios feasible, driven by the required use cases for 5G. These experiments include use cases that are currently high priority for 5G, such as MEC, with services being deployed at the edge of the wireless network and reducing the latency to access the content, experiments measuring the impact of different transport solutions for the functional splitting of the base station stack, end-to-end system virtualization and orchestration, as well as SFC, for the construction of chains of actions out of smaller elements that enable flexible service provisioning depending on the end-users' requests (for example chaining MME, HSS, S/P-GW services to form a virtual EPC [9] per infrastructure tenant).

The proposed integration will be developed around three strands:

1) Physical Interconnection of the facilities,

2) MANO integration of the offered experimentation capabilities, and

3) adding new services to the 5GINFIRE portfolio of offered services.

In the following subsections we analyse the testbed architecture and experimental components that are available in NITOS and will be integrated in the 5GINFIRE project.

3.3.1 NITOS testbed equipment

5G-VINO builds upon the existing efforts of 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 large single-site open experimental facilities in Europe, allowing users from around the globe to take advantage of highly programmable equipment (see Figure 18). The testbed is an integral part of larger federations of resources, such as OneLab [10] and Fed4FIRE [11], 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 in 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 [12] 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 [13] .
- Open Source LTE equipment, running over commodity Software Defined Radio (SDR) equipment, by the adoption of the OpenAirInterface (www.openairinterface.org) platform [14]. 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 SDR 5G testbed, consisting of 10 USRPs N210, 12 USRPs B210, 4 USRPs X310 and 4 ExMIMO2 FPGA boards. MAC and PHY algorithms can be executed over the SDR platforms, with very high accuracy.
- A millimeter wave testbed, operating in the V-band (60GHz), based on six nodes [15] . 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 [16] hardware switches, sliced using the FlowVisor [17] 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 [18] , 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.

Figure 18: NITOS testbed deployments; indoor/outdoor testbeds and COTS LTE macro-cell

For the integration with the 5GINFIRE orchestrator, NITOS will provide an interface to the VIM that is currently being used by the testbed. The adopted VIM by NITOS is OpenVIM [19], extended to support the following configurations:

- Each testbed node is a compute node for OpenVIM's datacenter configuration.
- Each node can use one of the two physical Ethernet interfaces for its control (management interface) by OpenVIM.
- New network configurations to support connection to the physical network interfaces of either LTE or Wi-Fi technology.
- Upon reception of a VNF configuration requiring the setup of an LTE network or a Wi-Fi AP, calls are made to the REST interface of the testbed's BSControl service [13] , that is in charge of configuring either the LTE infrastructure network (base stations and Core Network) or the Wi-Fi APs.
- Virtual Deployment Units (VDUs) are able to be provisioned regarding the LTE UEs with guarantees on the maximum achievable aggregate bandwidth on UL and DL per each client, Wi-Fi Access Point VDUs per each physical virtual access point provisioned through the Virtual Access Point functionality of the drivers used in the testbed, and Wi-Fi station VDUs.

A high level functional representation of this interaction across the different testbed components is shown in Figure 19. It is worth to mention, that currently Open Source MANO (OSM) is being used in NITOS for the orchestration of the testbed's resources. The extensions were developed using the OSM Release TWO, whereas all the functionality has been moved currently to OSM Release THREE. These extensions are also propagated to the OSM User Interface, where the end-user can list more network configurations, depending on the type of the network used.

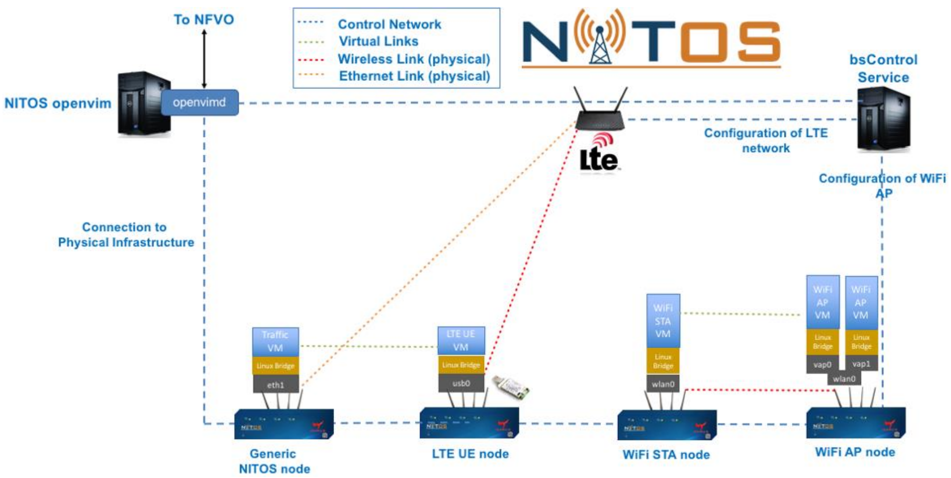

Figure 19: High Level interaction between the different components in NITOS through OpenVIM.

3.4 Wireless Network Slicing Functionality for 5G (WINS_5G) - FUTEBOL Integration

This Section presents the Wireless Network Slicing Functionality for 5G (WINS_5G). WINS_5G will extend 5GINFIRE capabilities by adding to the 5GINFIRE ecosystem the capability to use radio slicing and virtualization with a new facility located at Trinity College Dublin (TCD). WINS_5G also will bring to 5GINFIRE the opportunity to integrate with FUTEBOL project [20] and enable new experimentation use cases.

3.4.1 WINS_5G Overview

WINS_5G will add the capability to instantiate new NFV EVIs (*i*) to the radio access network supported by the radio slicing and virtualization tool called **Hy**pervisor for Software **D**efined **Ra**dios (HyDRA), developed by Trinity College Dublin researchers, (*ii*) which will be supported by the FUTEBOL Control Framework to support monitoring NFV experimentation scenarios in wireless, packet and optical networks. HyDRA as a VNF supported by Open Source MANO (OSM) will be available not only in the Iris Testbed, but also in other 5GINFIRE testbeds equipped with Universal Software Radio Peripherals (USRPs) N210s. These elements will enhance the 5GINFIRE ecosystem by offering the opportunity for experimenters to test and evaluate advanced 5G use case scenarios, such as massive eHealth communications in the Internet-of-Things, high-definition multimedia services in mobile broadband, and ultra-low latency communications for industry automation. WINS_5G is well aligned with the 5GINFIRE vision for 5G and general enough to support different Experimental Vertical Instances (EVIs).

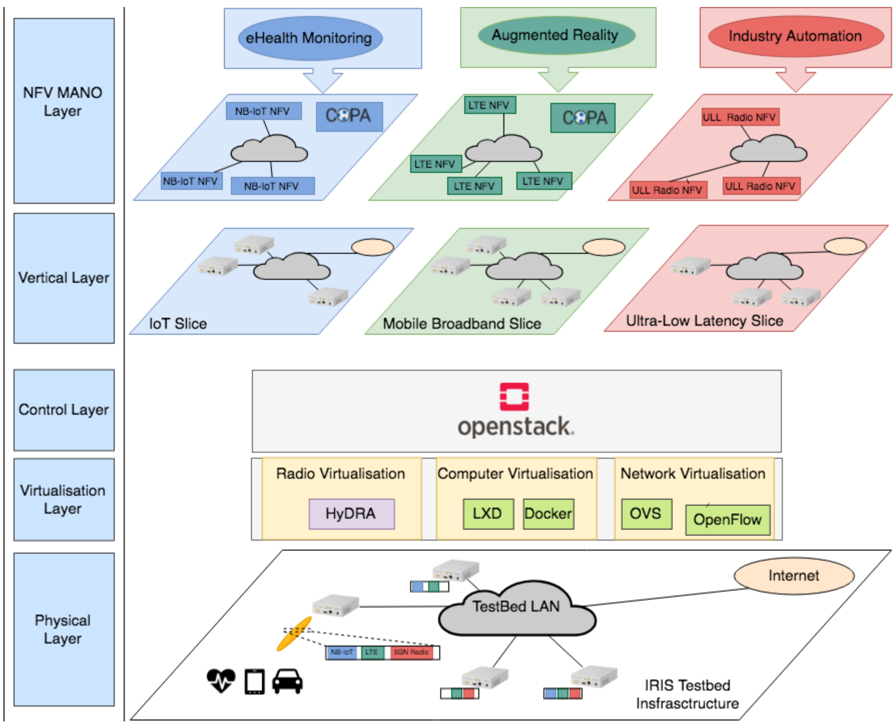

Figure 20 - WINS_5G overview.

3.4.2 The proposal

A high-level overview of WINS_5G architecture is depicted in Figure 20. The physical layer, at the bottom, represents the tangible resources including servers, switches, USRPSs, and so forth, at the Iris testbed. The virtualization layer in the middle is supported by FUTEBOL Control Framework technologies including the Container Orchestration and Provisioning Architecture (COPA). These interact with LinuX Containers (LXC), GNU Radio images, and HyDRA to support radio slicing, exposing the functionality of physical resources to applications and across different 5G verticals. The vertical resource layer at the top, supported by the OSM MANO software stack, will interact with the physical and virtualization layers to instantiate EVIs.

3.4.3 The Existing Iris Testbed

Currently, Iris - the Fed4FIRE+ federated reconfigurable radio testbed at Trinity College Dublin, pairs underlying flexible radio and computations resources (Physical Layer in Figure 20) with various hypervisors (virtualization layer in Figure 20) to realize several researches and testing configurations. This is directly supported by Iris's bespoke developed Cloud Based Testbed Manager (CBTM) framework, which operates on top of the hypervisor layer (Control Layer in Figure 20), allocating experimentation units across the Physical, Virtualization, and Control Layers, by supporting the creation of virtual machine images in physical rack servers. Currently, Iris's CBTM framework supports this operation by interacting directly with the *libvirt* virtualization API to manage the lifecycle of virtual machines. The CBTM is currently limited with the functionality it can offer to support instantiating and linking of NFV EVIs (Vertical Layer in Figure 20). To support EVIs, Iris will require an upgrade to the existing testbed virtualization layer. This upgrade will be completed in three steps.

3.4.4 Iris (FIRE testbed) Upgrade for 5GINFIRE Interconnection

First, to support interconnection with the 5GINFIRE multi-site OSM MANO orchestrator deployed in 5TONIC at UC3M, Iris will require an upgrade from the CBTM suite, to use OpenStack (Queens) to support resource orchestration. Towards this goal, we have installed and configured the following OpenStack services across two servers (controller and compute nodes) at the Iris testbed: Identity (Keystone), Image (Glance), Compute (NOVA), Networking (Neutron), Dashboard (Horizon), and Block Storage (Cider). The controller node will be connected to the 5TONIC via VPN over the GÉANT research and education network.

Second, Iris has dedicated USRP resources for allocation within the 5GINFIRE project. We employ the following hardware elements including: 4 x wall-fixed USRP N210s equipped with SBX daughterboards (frequencies between 400 MHz and 4.4 GHz with Rx/Tx up to 40 MHz); Two SMA Antennas Wi-Fi 2.4GHZ/5GHZ; and 1 GB Ethernet Connection to USRP. We envision these 4 USRP resources, complemented with HyDRA, the radio virtualization framework developed by Trinity College Dublin researchers (outlined further in the next section), will provide support for many EVI test scenarios from the 5GINFIRE community.

Third, building on existing functionality and expertise available at the Iris testbed, the WINS_5G extension intends to utilize some of the control framework tools developed by the FUTEBOL project to support richer experimentation scenarios and monitoring over EVIs in wireless, packet, and optical networks. The FUTEBOL H2020 project, which is an EU-Brazil collaboration that TCD coordinates, is creating a federated control framework to integrate wireless and optical testbeds in EU and Brazil for network researchers. To align with this project, tools available for integration with WINS_5G that fit within the ETSI NFV standard include COPA, which is an orchestration, provisioning, and monitoring (e.g., CPU, Memory, Signal Strength (RSSI), etc.) tool for containers. We currently envisage COPA virtual machine images being instantiated by 5GINFIRE EVIs that can support NFV container monitoring across the provisioned EVI network.

3.4.5 HYpervisor for Software Defined RAdios (HyDRA)

The open source HyDRA hypervisor licensed under the GPLv3 and developed by Trinity College Dublin researchers provides USRPs virtualization capabilities, enabling them to support multiple 5G verticals simultaneously. HyDRA, as shown in Figure 21, abstracts the RF front-end by adding a layer of indirection between virtualized radios, *i.e.*, the radio used by each vertical slice. Virtual Radios 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 virtual radio to adopt its own PHY and MAC technology, and configure its central frequency, bandwidth, and sampling rate on-the-fly.

Currently, HyDRA functionality is limited to radio level slicing. In the WINS_5G project, we will extend its current capabilities by integrating HyDRA as a VNF with the existing 5GINFIRE toolset portal and OSM, i.e. HyDRA as a MANO-compliant VNF. This extension, written for OSM as a Juju charm to support VNF configuration and management, will enable multiple slices to run on top of the same physical network and radio infrastructure so that they can simultaneously share the same physical network architecture. More specifically, each slice can adopt its own customized radio access technology, MAC layer and network layer protocols, without interrupting the operations and performance of other slices.

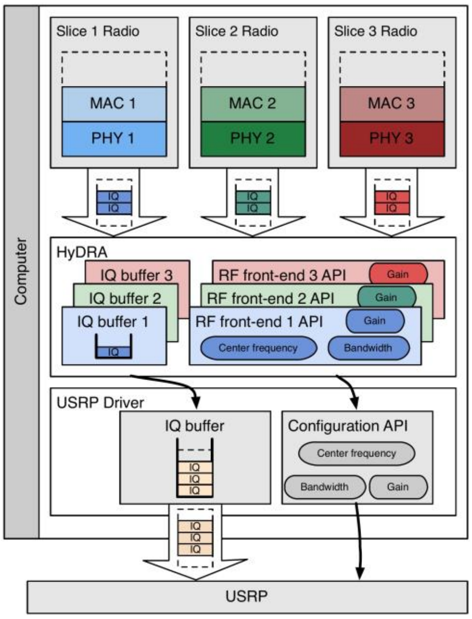

Figure 21 - HyDRA overview.

3.4.6 Iris Testbed 5GINFIRE Architecture

The combination of the upgrade to the Iris testbed to support OpenStack, HyDRA virtualization toolkit, COPA from the FUTEBOL control framework, and interconnection with the 5GINFIRE multi-site OSM MANO orchestrator at 5TONIC at UC3M, will enable the creation of slices that can be deployed on-demand and dynamically allocated regardless of the service vertical. Furthermore, sample EVI that support the instantiation of many different verticals for Ultra-Reliable Low Latency Communication (URLLC), enhanced Mobile BroadBand (eMBB), and massive Machine Type Communications (mMTC) across the Iris radio testbed, will be provided to 5GINFIRE experiments. A high-level view of the Iris 5GINFIRE testbed is available in Figure 22.

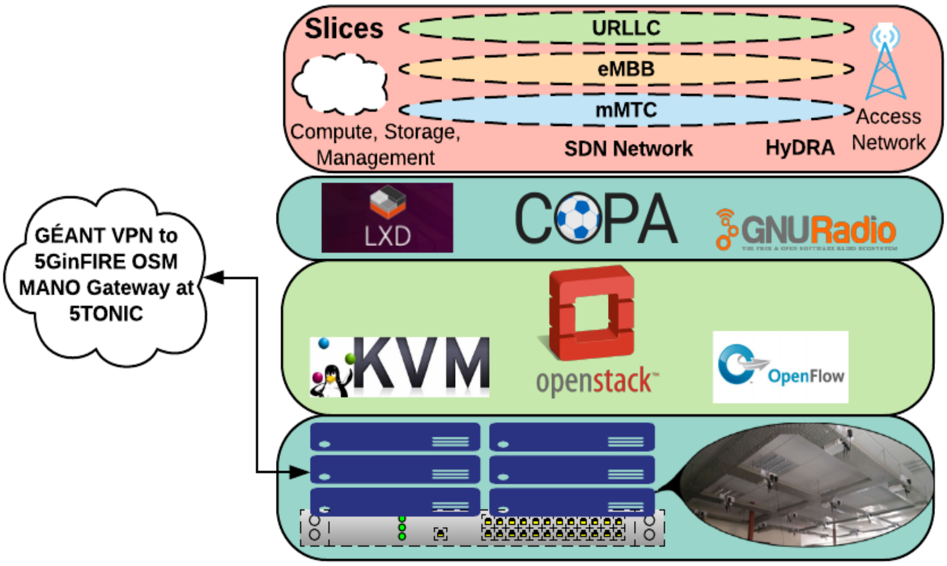

Figure 22 - Iris 5GINFIRE Integration. Testbed Providers Integration Guide.

In this section, we provide a general overview of the necessary steps to integrate the NFV infrastructure of a testbed provider into the 5GINFIRE MANO platform. The list of steps indicated in this section is not intended to be exhaustive, and additional procedures might be considered for each specific case. Additionally, the section provides a summary of the functional tests that will be carried out to validate a successful integration, verifying the availability of control and data-plane communications with the site of the new testbed provider (these functional tests are detailed in deliverable D4.1 [21]). Finally, we present a number of tests that will be executed to evaluate the performance of the inter-site communications among the sites conforming the 5GINFIRE MANO platform.

3.5 Integration of a new site

As described in deliverable D4.1 [21], the approach currently used in 5GINFIRE to support the exchange of control and data-plane information, among the sites that provide the MANO platform, considers the utilization of an overlay network architecture based on Virtual Private Networks (VPNs). Figure 23 illustrates the diverse communications encapsulated through the 5GINFIRE VPN that are needed to support the deployment of network services and VxFs:

- **Control-plane communications**, supporting: 1) the exchange of information between the OSM stack at 5TONIC and the VIMs that manage the experimental infrastructures (these communications are represented as a yellow line in Figure 23); and 2) the configuration of VxFs by the OSM stack (represented with a green line in Figure 23).

- **Data-plane communications**, enabling the exchange of information among VxFs deployed at different sites (purple line in Figure 23).

This way, control-plane information is distributed using a star topology centred around 5TONIC, which hosts the VPN server that supports the overlay network. At the time of writing, data-plane communications among sites are also established through 5TONIC, using the VPN server as a network relay. Anyway, the considered approach is flexible enough to support direct data communications between sites, as illustrated in Figure 23. In the shown example, site 1 offers a VPN service for data-plane communications, providing network access connectivity to site 2.

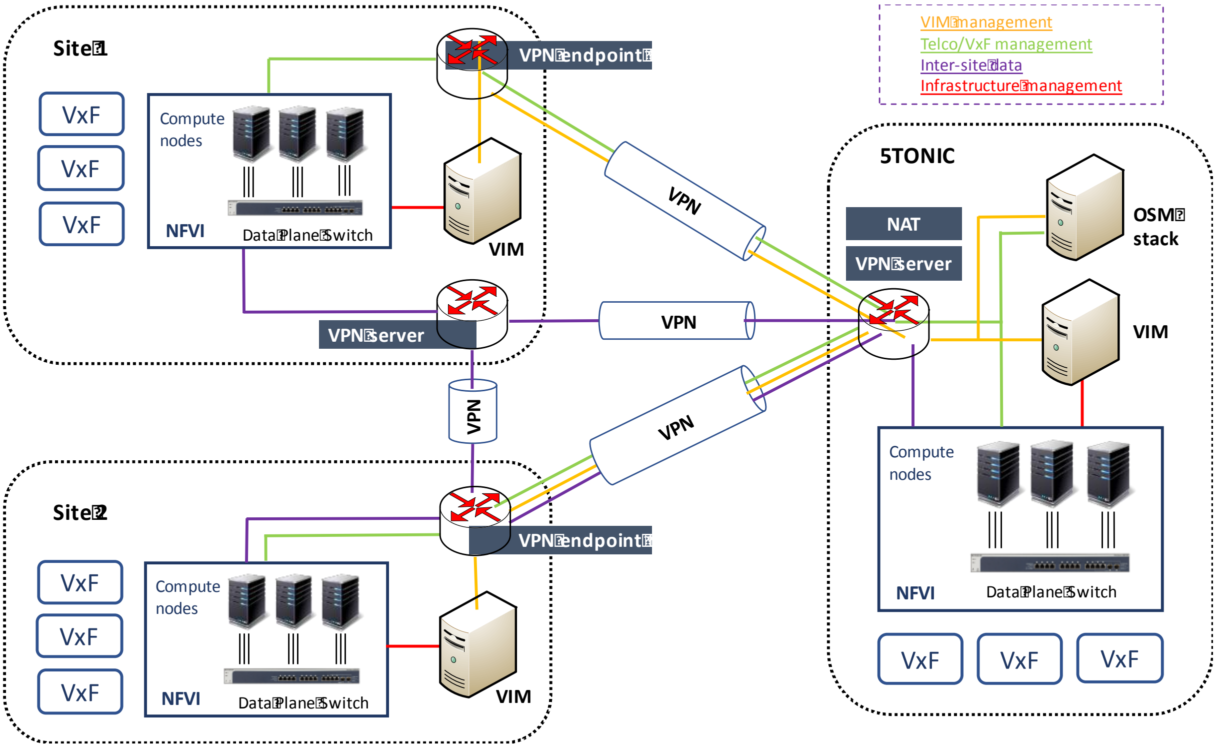

Figure 23: Examples of inter-site connectivity.

In the following, we provide a non-exhaustive list of the different steps that have been identified as necessary to support the integration of a new site to the 5GINFIRE MANO platform:

- **Utilization of a VIM solution, compliant with the OSM stack of the** 5GINFIRE **MANO platform**. At the time of writing, OSM release FOUR (the version of OSM that will be used by the platform in the short-term) supports OpenVIM, OpenStack, VMware vCloud Director and Amazon Web Services (AWS).

- **Obtaining an appropriate IP address space, along with VPN credentials, to connect to the VPN-based overlay network**. Requests for VPN credentials and IP address ranges should be placed through the issue management system of 5GINFIRE (see next section).
  With respect to the address space, deliverable D4.1 [21] contains the agreements that have been taken by 5GINFIRE. For convenience, these are summarized here: 1) testbed providers will use the private address space 10.154.0.0/16 for control and data plane communications; 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 current assignment of IP addresses to 5GINFIRE members is presented in Table 4.

- **Configuration of the site with the allocated IP address space**, such that the VIM and the VxFs are addressable using an IP address sub-space of 5GINFIRE.

- **Configuration of a VPN endpoint to connect the site to the overlay network of** 5GINFIRE**.** This will require the installation of a VPN client, and its configuration with the previously obtained credentials.

- **Configuration of the appropriate VIM networks**, to support inter-site control and data plane communications with the VxFs deployed at the site. As an example, these networks can be pre-created as "*provider networks"* using OpenStack.

- **Do any necessary local configurations to support the routing of control and data plane information** across the local network segments of the external entity, i.e. between the VPN endpoint and the VIM and VxFs of the site.

Table 4: Current allocation of IP addresses to sites

Site	IP address range
5TONIC	10.4.0.0/16
ITAv	10.154.0.0/20
UNIVBRIS	10.154.16.0/20
UFU	10.154.32.0/20

3.6 Functional tests

In addition, deliverable D4.1 [21] provides detailed information about the functional tests that have been used to validate inter-site control and data-plane communications. After the integration of a new site, these tests will be carried out to verify that these types of communications are available with the site, this way certifying the appropriate integration of the site into the 5GINFIRE MANO platform. The descriptions of these functional tests, along with their fundamental objectives, are collected in Table 5.

Table 5: Functional tests to validate inter-site communications

Test ID	Description	Objective
F1	Deployment of a reference Network Service (NS) in the new site, not requiring day-1 configuration of VxFs. The reference NS is composed by two interconnected VxFs, which do not require configuration through the VNF Configuration and Abstraction (VCA) module of the OSM stack.	Validate the inter-site control-plane communications that are necessary to deploy a NS

Test ID	Description	Objective
F2	Deployment of a reference NS in the new site, requiring day-1 configuration of VxFs. In this case, the NS consists of two interconnected VxFs that require Juju configuration via the VCA module. In this test, both VxFs are deployed at the new site.	Validate the inter-site control-plane communications that are necessary to deploy a NS and configure its constituent VxFs.
F3	Deployment of a reference NS using the multi-site capabilities of the 5GINFIRE MANO platform. In this test, the NS is the same as in test F2, however one VxF of the NS service is deployed at the new site, while the other one is deployed at 5TONIC.	Validate the inter-site control and data-plane communications that are necessary to: deploy a NS, assigning each VxF to a different site; configure the VxFs; and support their data communications.

3.7 Performance tests

To conclude this section, we present an overview of the tests and experiments that have been identified to evaluate the performance that can be achieved by existing inter-site communications in 5GINFIRE. We consider two categories for performance measurements:

1) Network performance (i.e., throughput & delay).

2) The influence of remote operations in lifecycle events (i.e., deployment time of a NS).

Table 6 shows the performance tests that have been identified by the 5GINFIRE consortium, and that will be carried out for every site integrated in the MANO platform of the project.

Table 6: Summary of performance tests

Test ID	Objective	Target parameters	Methodology
P1	Obtaining performance figures of inter-site communications.	Available throughput, Round Trip Time (RTT) and Jitter, considering inter-site back-to-back communications between every site.	Measurements will be taken every hour, using a Virtual Machine (VM) deployed at every site.
P2	Evaluate how the number of VNFs affects the deployment of a NS.	Deployment time of a reference NS.	For each site, deploy a reference NS with an increasing number of VNFs

Test ID	Objective	Target parameters	Methodology
	Compare the performance of remote deployments against reference values.		(e.g., 1, 2, 4, 8…). For each deployment of the NS, measure the average deployment time.
P3	Evaluate how deployment time is affected by existing deployments. Compare the performance of remote deployments against reference values.	Deployment time of a reference NS.	For each site, consecutively deploy a reference NS a number of times (e.g., up to 4). For each deployment of the NS, measure the average deployment time

The measurements of test P1 will be repeated daily, and the results collected to provide performance statistics of inter-site communications. As an example, Figure 24 and Figure 25 represent the throughput that is available for inter-site communications between 5TONIC and ITAv, obtained during a set of preliminary experiments with the VM that has been developed to execute Test P1.

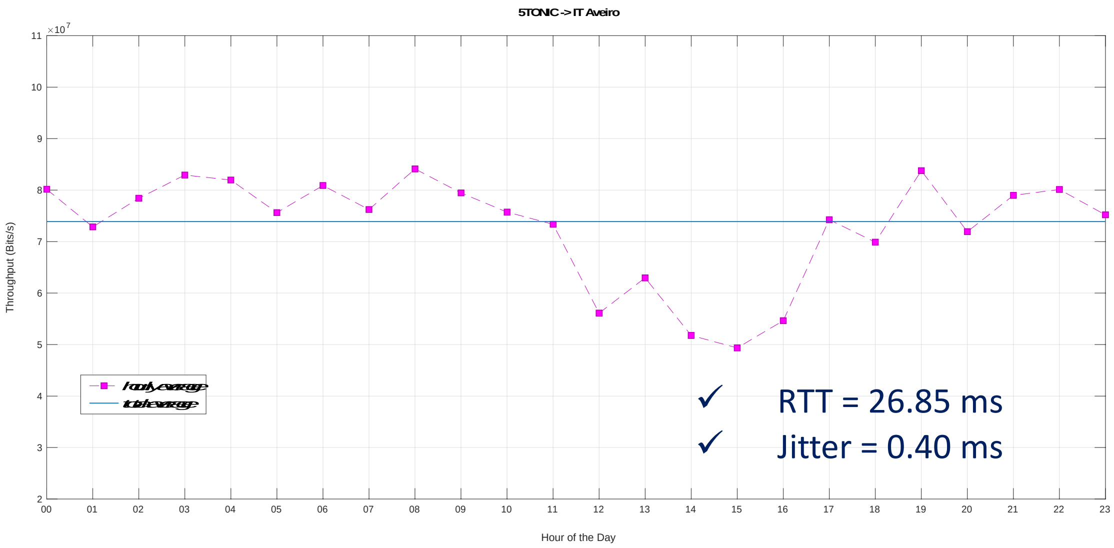

Figure 24: Performance figures 5TONIC → ITAv.

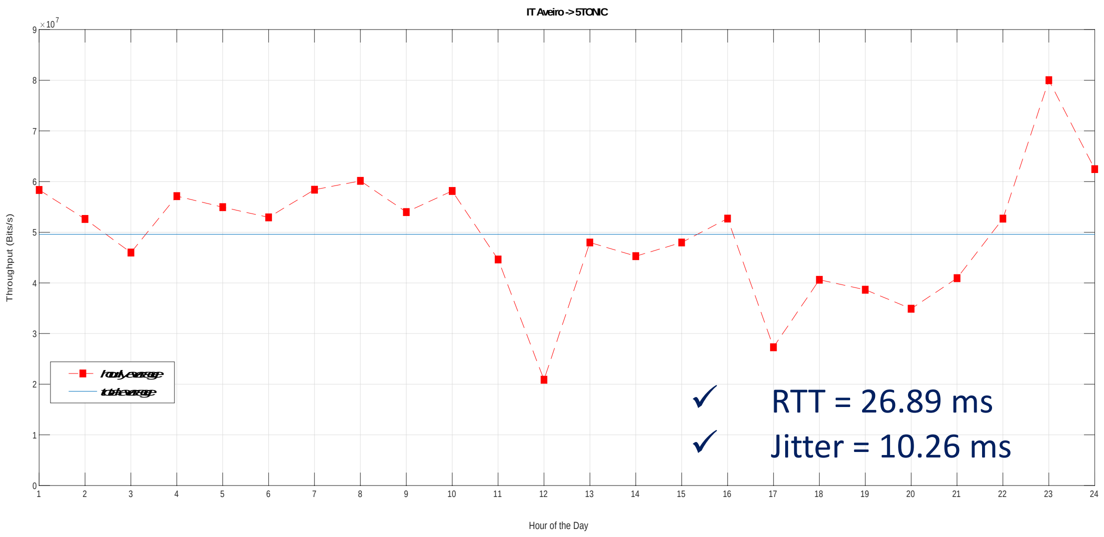

Figure 25: Performance figures ITAv → 5TONIC.

Figure 26a shows the set of results obtained with the execution of test P2 at 5TONIC, i.e., using a datacenter at the same location as the OSM stack. The graph illustrates the average deployment time of a reference NS (available at https://github.com/5GinFIRE/mano) as the number of its constituent VxFs increases from 1 to 4. As it can be observed from the figure, the deployment of a NS with a single VxF requires approximately 55 seconds. Increasing the number of VxFs introduces higher deployment times, with an extra delay of less than 25 seconds per each additional VxF.

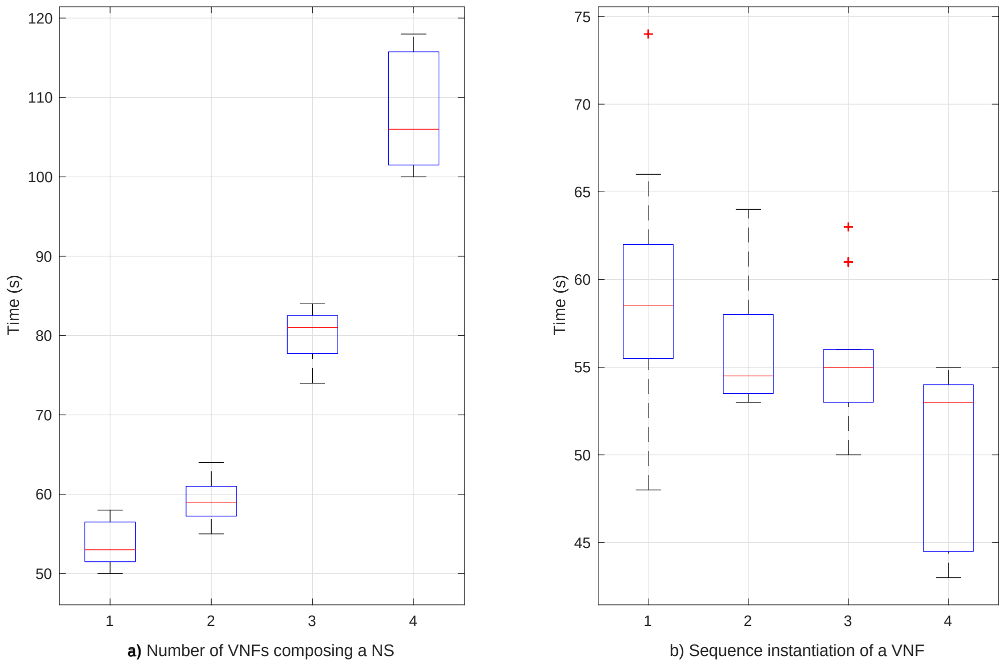

Figure 26: Performance figures of deployment time.

On the other hand, Figure 26b shows the set of results obtained after carrying out test P3 using a datacenter at 5TONIC. According to the information provided in Table 6, the experiment consisted in the successive deployment of a reference NS. The NS was composed of a single VxF (available at https://github.com/5GinFIRE/mano) and was deployed four consecutive times. The picture shows the average time required to do the first, second, third and fourth deployment of the NS. As it can be seen, the first deployment completes in approximately 55 seconds (as in the previous test), being in general lower for subsequent deployments.

The results collected in Figure 26 will be used as a reference to evaluate the performance that can be achieved in remote deployments at every other site integrated into the 5GINFIRE MANO platform.

4 5GINFIRE Experimenters Guide

The requirements and the supported actors by the 5GINFIRE portal were presented in D2.1 Section 4. For clarity we also present them here with less details:

- Experimenter: can upload Experiments in terms of NSDs and request the deployment of an experiment over the 5GINFIRE infrastructure
- VxF Developer: can upload VxF archives
- Testbed provider: can register a target infrastructure
- Services administrator: responsible for the portal management

The portal, as well as the underlying 5GINFIRE services like the MANO stack, needs to support certain functions of the 5GINFIRE experimentation workflow presented in D2.1 Section 4.3.

The experimenter, the 5GINFIRE operations and the 5GINFIRE testbed providers that interact during an experimentation life-cycle. At the simplest case, users signed-up to the platform via the portal, will be approved by 5GINFIRE Operations. To perform an experiment on top of the 5GINFIRE infrastructure at its simplest form the user needs to create an experiment, e.g. some experiment metadata, scheduling, purpose, etc., and select available VNFs or deploy new ones. Then he needs to compose the experimentation solution. This can be done as an OSM NSD. The user will provide an OSM-supported YAML description of the network service, potentially aided by a graphical composer. As soon as everything is in place for an experiment description, the experimenter selects the testbed facility based on resource availability after the experiment is submitted for validation.

5GINFIRE prepares a process for validating an experiment in terms of various rules such as scheduling, resource availability, etc. The procedure is supported by 5GINFIRE Wiki for documentation but also an issue management system based on Bugzilla. The validation process is closely performed together with the target testbed providers. This process is iterative in various cycles involving the experimenter by either asking questions or modifying any experiment details and parameters.

As soon as an experiment is approved, it is scheduled by 5GINFIRE operations for deployment. Through the portal or OSM the 5GINFIRE operations create a deployment (i.e. uploading descriptors etc.) and OSM will later orchestrate it (trigger the services instantiation). We expect that there will be a close collaboration during the management of the orchestration/deployment with the testbed providers. After deployment, the resources are available and accessible to the experimenter.

At the end of the experiment schedule, the resources of the experiment are released, and access is revoked. We expect though that any available results of the experiment will be available to the experimenter.

Currently to deploy an EVI, is accomplished in terms of NSD through the portal. For example, at Figure 27 presents all the necessary fields to define a new deployment. The experimenter can select also the target infrastructure for all or for each individual constituent VxF. The experimenter can submit this specific deployment by clicking the request deployment button on the bottom of the interface. Then this deployment request is sent to the service administrator to be processed or to be rejected.

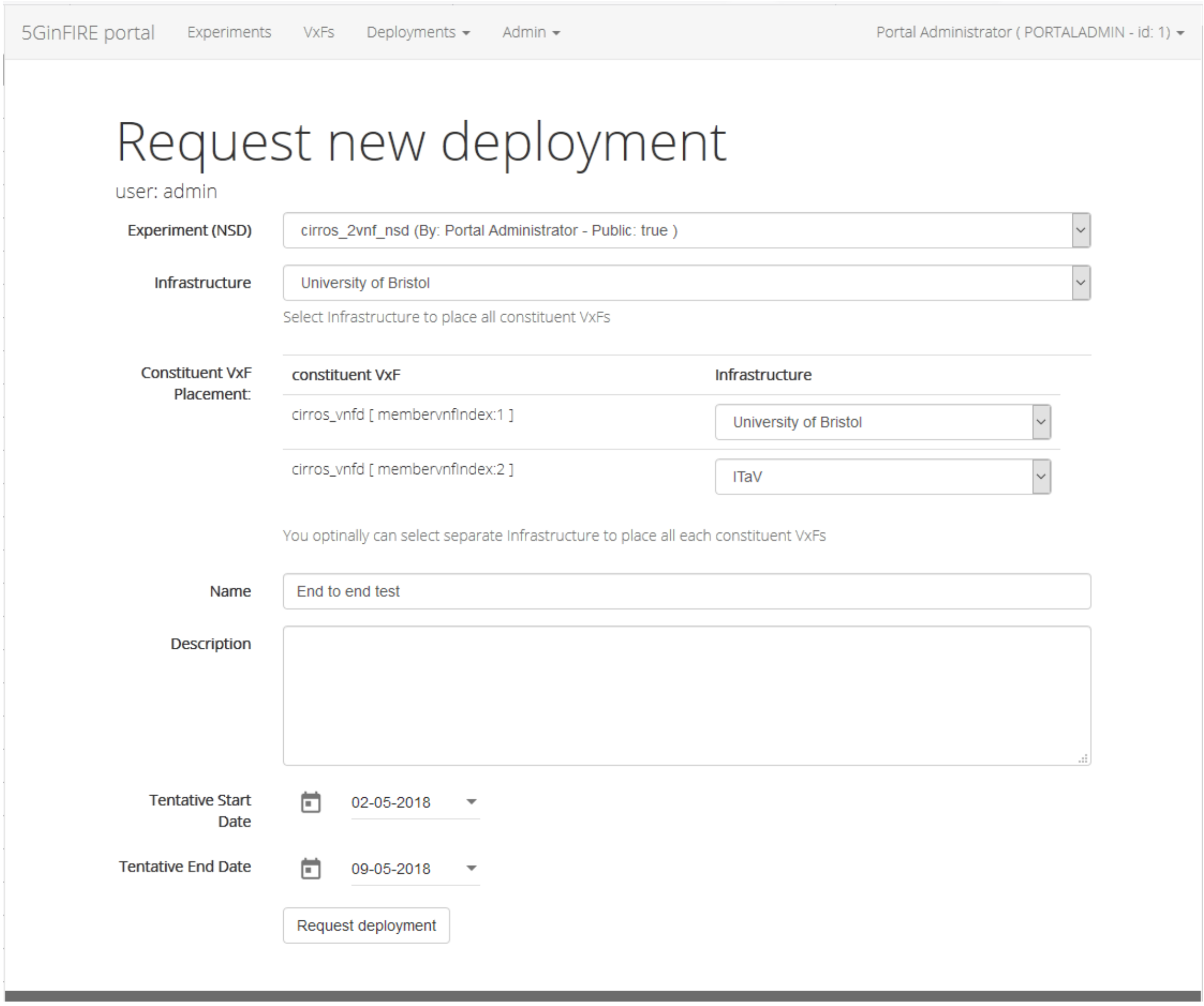

Figure 27 - Request Form to Define a New Experiment Deployment.

As a result of this procedure, and assuming that the experiment is approved, the experiment description is provided to the administrator of the 5GINFIRE MANO platform (OSM MANO - based on Version TWO for now), NS and VxF descriptors and packages provided by VxF developers and experimenters must conform to the formats supported by OSM (NFV descriptors supported by OSM are based on YANG).

After verifying that the images of the VMs that will be needed to deploy the VNFs are available at the deployment sites provided by 5GINFIRE testbed providers, the administrator of the MANO platform can then deploy the NS. This is done using the OSM Launchpad, which is the graphical user interface that can be utilized to interact with the run-time system of OSM. The NS consists of two interconnected VNFs and, the administrator of the MANO platform can choose a specific datacenter for the deployment of each of the VNFs. Besides this, the Launchpad also includes a Dashboard that provides real-time information about the deployed VNFs and NSs. More comprehensive information on the OSM Launchpad can be found in D3.1 as well as the 5GINFIRE Wiki.

4.1 Exposed API to the experimenters

The 5GINFIRE Portal architecture was conceived as two decoupled components: the web frontend and the API backend. The Web front end communicates with the backend via a RESTful API, as presented in Figure 28.

The portal backend API is used by the front end, but it is also available for use by 3rd party applications. A complete reference of this API can be found in Portal API [22] page available 5GINFIRE Wiki [23].

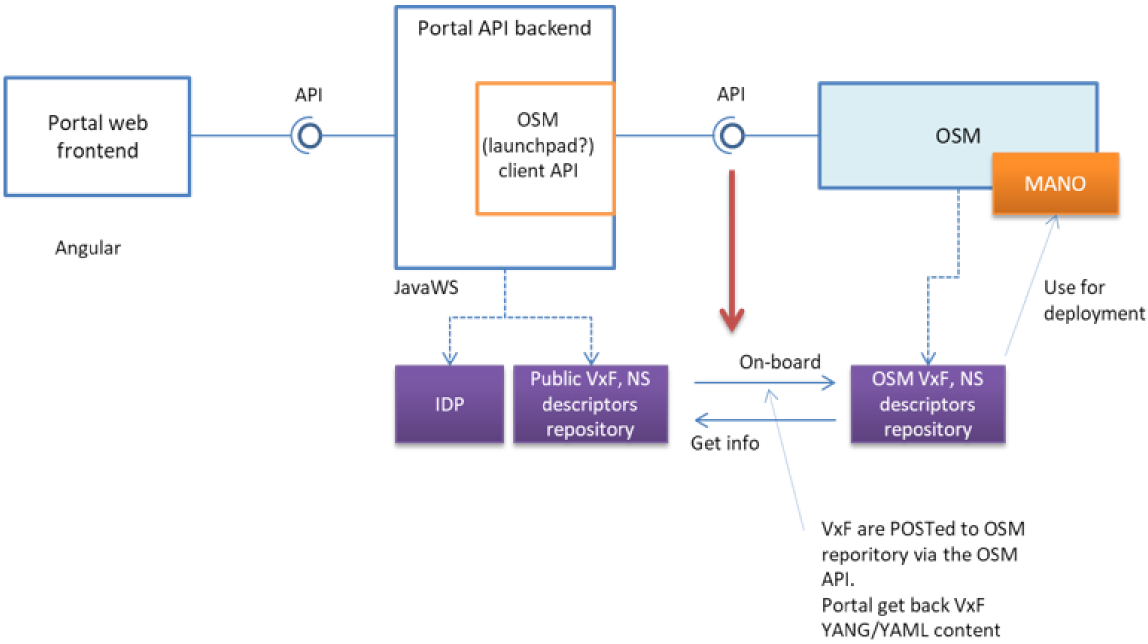

Figure 28 - 5GINFIRE Portal architecture.

4.2 Exposed VxFs to the experimenters

This section describes some VxFs that are available to the experimenters. When a VxF is published in the 5GINGIRE portal, if allowed by the publisher, it can be accessed by using the portal or the API described in Section 4.1.

These VxFs are used by the Automotive and Smart City EVIs. The experimenters can use them to compose new services and applications.

4.2.1 5G-In-A-Box for Experimentation

This Section describes the b<>com *Unifier Gateway, a.k.a., 5G-In-A-Box,* a pre-5G Mobile Edge private connectivity enabler.

4.2.1.1 Overview

b<>com *Unifier Gateway* is an SDN based private network framework enabling end to end broadband, IoT and WebRTC critical communications to be carried out in a full secure manner in small to medium size buildings or industrial sites.

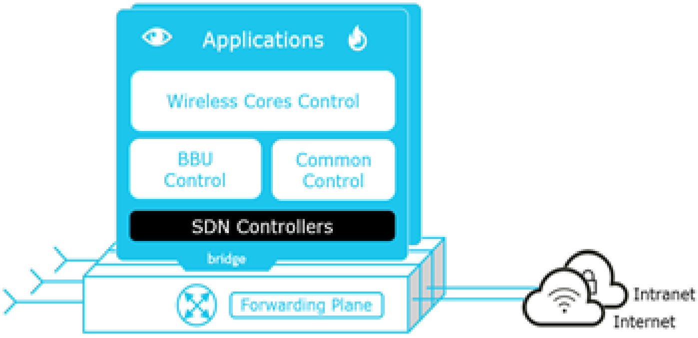

Figure 29: b<>com Unifier Gateway Overview.

- Key features:
  - Security: Unified SIM based WLAN/LTE authentication approach (EAP-AKA, EPS-AKA), patented and distributed SDN based firewalling solution, pre-5G slice-based architecture ensuring isolation between services
  - Service continuity: Unified DHCP that enables seamless mobility between connected RAN for all services profiles
  - Onboarding and Provisioning: Zero touch installation and provisioning (Heat, Ansible playbooks)
  - Inexpensive: Compatible to Commercial Off-The-Shelf Wi-Fi Access Points and eNodeBs, Instantiation on standard IT infrastructure

- Specifications:
  - SDN (Openflow v1.3, OpenDaylight controller)
  - OpenStack Pike
  - Preloaded with Full LTE EPC (MME, S/P-GW, HSS), 3GPP Rel10 compliant
  - WLAN 802.1x protocols
  - EAP-AKA, EPS-AKA SIM based authentication mechanisms

4.2.1.2 PNF UGW

b<>com Unifier Gateway is deployed in Bristol Smart City Safety Testbed as a PNF (Physical Network Function), with a dedicated hardware to provide "5G-in-a-box" solution.

Figure 30: "5G-in-a-box" solution.

b<>com *Unifier Gateway* has deployed according to the architecture detailed in the figure just below.

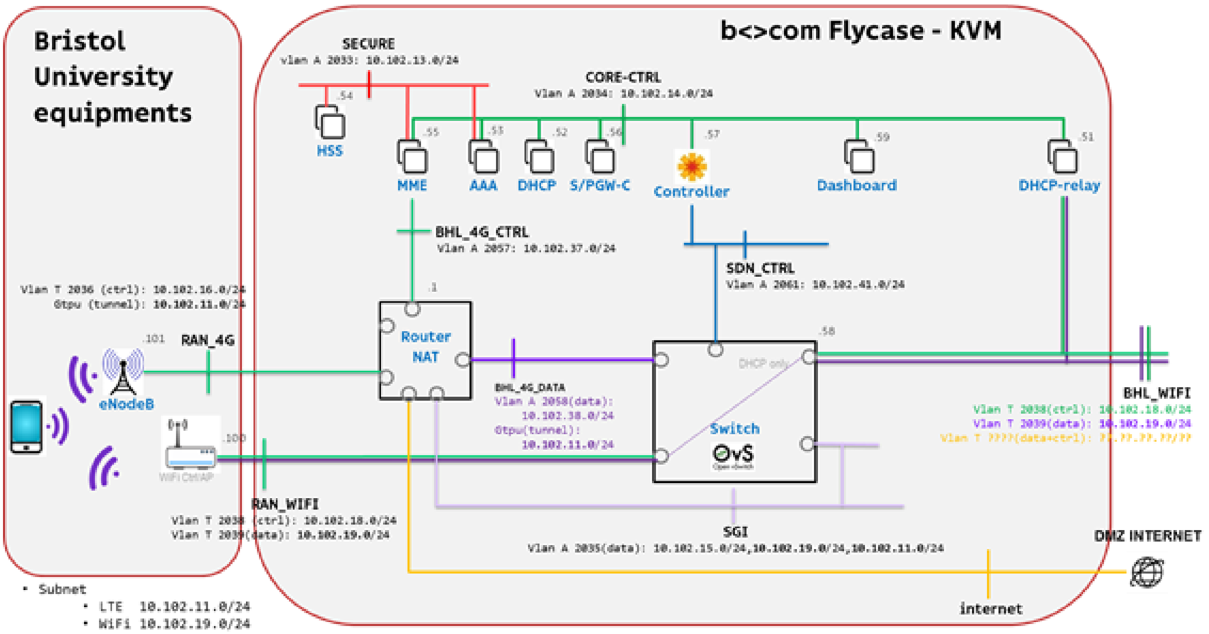

Figure 31: Bristol PNF UGW Architecture.

4.2.1.3 VNF UGW

b<>com *Unifier Gateway* is deployed in IT-AV AUTOMOTIVE TESTBED as a VNF hosted on an edge cloud operated by IT-Aveiro. It follows the architecture detailed in the Figure 32.

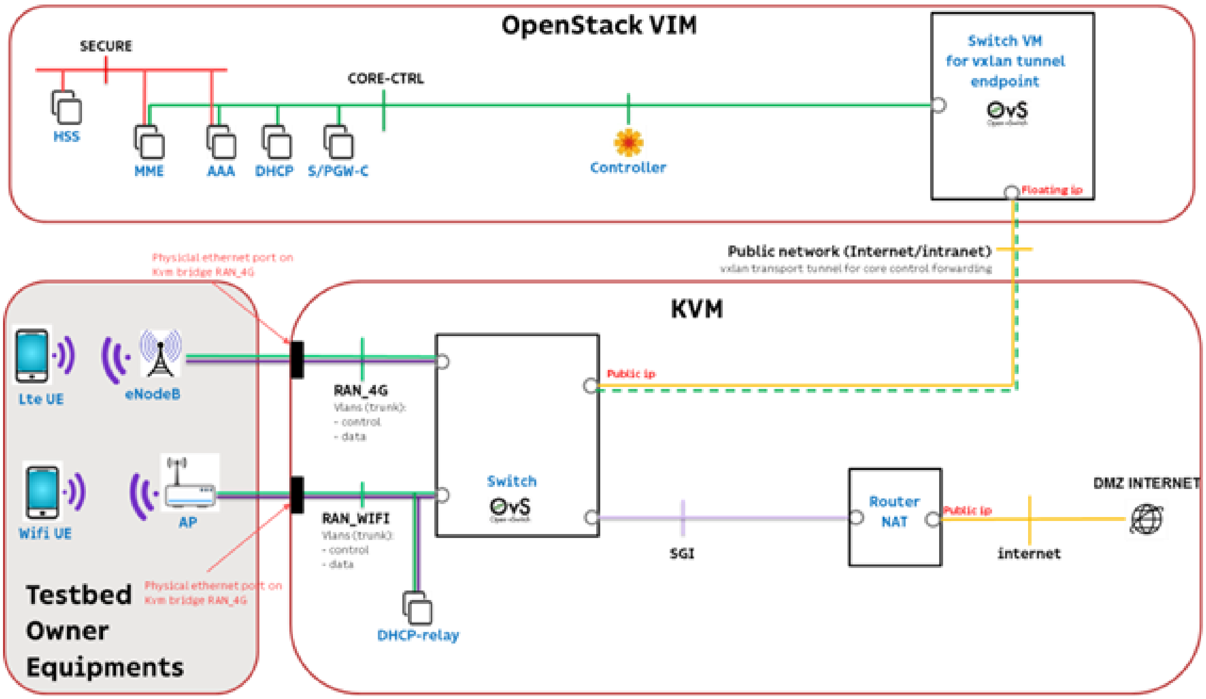

Figure 32: VNF UGW Architecture.

4.2.2 FFMPEG Transcoder VNF

This VNF can be used to convert video formats and configure video characteristics such as sample rates, frame rates, resolutions, etc. Its main purpose is to add flexibility to implementations that require video streaming and to facilitate the integration of different video capture devices, display screens, media players and/or streaming servers. The encoding tool used is FFmpeg, a video and audio converter that provides support for a wide variety of formats, encoders and containers.

The current implementation is used to convert videos compressed following the popular H.264 standard into a M-JPEG compressed video. The latter is a better-suited compression format for applications that require video streaming or multimedia PC applications, such as IP cameras, webcams, and generic surveillance applications. Besides being compatible with most media players, it is natively supported by most Web browsers, including Google Chrome, Mozilla and Microsoft Edge, providing more options when choosing the devices that will display the video to application users. Furthermore, FFmpeg is fully integrated with FFserver, a framework that essentially acts as a HTTP server, providing support to host the video data recorded, as well as to deliver the video images via HTTP.

4.2.3 OpenCV Transcoder VNF

This VNF is derived from OpenCV (Open Source Computer Vision), which is a library of programming functions mainly aimed at real-time computer vision. Basically, two functionalities are deployed in this VNF: video transcoder and face recognition system (face detection and face recognition). In the current implementation (Smart City Safety use case) the virtual function video transcoder is used to transform a 360-degree live stream video from spherical to rectangular format. After this transformation, the virtual function face detection and recognition are applied to detect and recognize people faces. The VNF video transcoder enable to transcoder the 360-degree live stream video in different formats facilitating the visualization from different mobile devices.

4.2.4 Residential Gateway VNF

The VNF descriptor of a residential gateway function. This function includes a management interface and two data interfaces, supporting the exchange of information between a WAN and a residential environment. Routing functionalities are provided through a Linux Ubuntu 16.04.2 LTS virtual machine. The function provides a DHCP service to support the auto-configuration of end-user equipment connecting from the residential side. The configuration of the VNF is supported through an Ansible playbook which is encapsulated as a Juju charm in the VNF package.

VxF Image is also available in GitHub: https://github.com/5GinFIRE/mano/blob/master/descriptor-packages/vnfd/residentialGW_vnfd/icons/Residential.png.

4.2.5 Router 5TONIC VNF

The VNF descriptor of a routing function. Similarly, to the Residential Gateway VNF, this function includes a management interface and two data interfaces, and supports data forwarding between them. Routing functionalities are provided through a Linux Ubuntu 16.04.2 LTS virtual machine. The configuration of the VNF, including the IP addresses of the data interfaces and its routing table is supported through an Ansible playbook, which is encapsulated as a Juju charm in the VNF package.

The VxF Image is available in GitHub:
https://github.com/5GinFIRE/mano/blob/master/descriptor-packages/vnfd/router_5tonic_vnfd/icons/Router.png.

4.2.6 Video Server VNF

The VNF descriptor of an HTTP server function. This function can maintain several video files that can then be requested on demand by interested consumers through the HTTP protocol (for instance, using GStreamer or a VLC client). The VNF has a management and a data interface and is configured via an Ansible playbook that activates the HTTP server function and configures the IP addressing information for the appropriate operation of the VNF.

VxF Image is also available in GitHub:
https://github.com/5GinFIRE/mano/blob/master/descriptor-packages/vnfd/videoServer_vnfd/icons/videoServer.png

5 Conclusion

This document presented an overall status of the current tasks inside the WP5 whose primary goal is to demonstrate the experimentation enablement of the 5GINFIRE ecosystem.

This document detailed the status of the automotive and smart city EVIs. It also presented a set of new experiments, proposed in the first open call, that will use each one of these EVIs. Moreover, a new EVI, in eHealth, resultant from the first open call is presented. This new EVI will enable the exploitation of new experiments in a new 5G vertical.

Regarding the goal of integration between 5GINFIRE and the FIRE world, the document detailed the ongoing efforts to integrate with FIESTA-IoT, NITOS, and FUTEBOL FIRE based facilities.

The testbed providers integration guide presented how new providers can join the 5GINFIRE ecosystem, and an experimenters guide shows how to start a new experiment in the 5GINFIRE facility.

This document is then an entry point to new providers, experiments and the several actors from the 5G ecosystem to get into the 5GINFIRE.

6 References

[1] M. Smirnov *et al.*, “Detailed Specifications Regarding Experiment Workflow Tools and Lifecycle Management for the Second Cycle (D5.2).” 25-Apr-2014.

[2] A. Israel *et al.*, “OSM Release Three: A Technical Overview.” Oct-2017.

[3] A. Hammad *et al.*, “FUTEBOL Control Framework Design: Deliverable D4.1,” 2016.

[4] I. de A. Ceravolo *et al.*, “O2CMF: Experiment-as-a-Service for Agile Fed4Fire Deployment of Programmable NFV,” in *Optical Fiber Communication Conference (2018), paper Tu3D.13*, 2018, p. Tu3D.13.

[5] FIBRE, “Future Internet Brazilian Environment for Experimentation (FIBRE),” 2017. [Online]. Available: http://fibre.org.br/. [Accessed: 19-Dec-2017].

[6] FIBRE, “FIBRE: Ofelia Control Framework,” 01-Jun-2017. [Online]. Available: https://wiki.rnp.br/display/fibre/8.+OCF+-+Ofelia+Control+Framework. [Accessed: 07-Jun-2018].

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

[8] “Network Implementation Testbed Laboratory (NITOS),” May-2018. [Online]. Available: https://nitlab.inf.uth.gr/NITlab/nitos. [Accessed: 04-Jun-2018].

[9] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC),” *IEEE Netw.*, vol. 28, no. 6, pp. 18-26, Nov. 2014.

[10] S. Fdida, T. Friedman, and T. Parmentelat, “OneLab: An Open Federated Facility for Experimentally Driven Future Internet Research,” in *New Network Architectures*, Springer, Berlin, Heidelberg, 2010, pp. 141-152.

[11] B. Vermeulen, W. Van de Meerssche, and T. Walcarius, “jFed Toolkit, Fed4FIRE, Federation,” in *GENI Engineering Conference*, 2014, vol. 19.

[12] G. R. Hiertz *et al.*, “IEEE 802.11s: The WLAN Mesh Standard,” *IEEE Wirel. Commun.*, vol. 17, no. 1, pp. 104-111, Feb. 2010.

[13] N. Makris, C. Zarafetas, S. Kechagias, T. Korakis, I. Seskar, and L. Tassiulas, “Enabling open access to LTE network components; the NITOS testbed paradigm,” in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 2015, pp. 1-6.

[14] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, “OpenAirInterface: A Flexible Platform for 5G Research,” *SIGCOMM Comput Commun Rev*, vol. 44, no. 5, pp. 33-38, Oct. 2014.

[15] Blu Wireless Technology, “Blu Wireless Technology WiGig Products,” *Blu Wireless*, May-2018. [Online]. Available: http://www.bluwirelesstechnology.com/. [Accessed: 14-May-2018].

[16] N. McKeown *et al.*, “OpenFlow: Enabling Innovation in Campus Networks,” *SIGCOMM Comput Commun Rev*, vol. 38, no. 2, pp. 69-74, Mar. 2008.

[17] R. Sherwood *et al.*, “Flowvisor: A network virtualization layer,” *OpenFlow Switch Consort. Tech Rep*, vol. 1, p. 132, 2009.

[18] T. Korakis, “NITOS IoT Testbed and Applications.” 17-Jul-2017.

[19] R. Mijumbi, J. Serrat, J. l Gorricho, S. Latre, M. Charalambides, and D. Lopez, “Management and orchestration challenges in network functions virtualization,” *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 98-105, Jan. 2016.

[20] FUTEBOL, “Federated Union of Telecommunications Research Facilities for an EU-Brazil Open Laboratory (FUTEBOL),” 2018. .

[21] D. R. López *et al.*, “Operational MANO Platform (D4.1).” Jun-2017.

[22] “Portal API: 5GinFIRE Wiki,” Mar-2018. [Online]. Available: http://wiki.5ginfire.eu/portalarchitecturedesign/portal-api. [Accessed: 07-Jun-2018].

[23] 5GINFIRE, “5GinFIRE Wiki,” 2018. [Online]. Available: http://wiki.5ginfire.eu/. [Accessed: 05-Jun-2018].