



**Software Defined Networks and Network Function
Virtualization Testbed within FIRE+**

Grant Agreement N° 687860

D3.3.

**Guidelines, rules and mechanisms governing the usage of the
SoftFIRE Testbed**

WP3

Experimentations Management

Version: 1.1

Due Date: July 31st 2016

Delivery Date: August 16th 2016

Type: Report (R)

Dissemination Level: PU – Public

Lead partner: TI

Authors: All Partners (See list below)

Internal reviewers: PMC

Disclaimer

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

The commercial use of any information contained in this document may require a license from the proprietor of that information.

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



SoftFIRE has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 687860.



Version Control:

Version	Date	Author	Author's Organization	Changes
1.0	05/08/2016	Bjoern Riemer	FOKUS	Extension to D3.2 Handbook for experimenters about the Programming LifeCycle
1.1	05/08/2016	Roberto Minerva	Telecom Italia	Comments and editing
1.2	12/08/2016	Susanne Kuehrer	EIT Digital	Approval at PMC
1.2	16/08/2016	Susanne Kuehrer	EIT Digital	Final editing
		Roberto Minerva	Telecom Italia	

Annexes:

Nº	File Name	Title
----	-----------	-------



Contributors:

Contributor	Partner
Massimiliano Calavaro	Ericsson
Giuseppe Carella	TUB
Davide Carmignani	Security Reply
Marius Corici	FOKUS
Peter Feil	DT
Gerry Foster	University of Surrey
Antonio Fusco	Ericsson
Kay Haensge	DT
Susanne Kuehrer	EIT Digital
Roberto Minerva	Telecom Italia
Simone Monticelli	Security Reply
Sergio Nuccio	Telecom Italia
Marco Persichini	Ericsson
Björn Riemer	FOKUS
Umberto Stravato	Ericsson
Lorenzo Tomasini	TUB
Daniel Nehls	TUB
Mario Ullio	Telecom Italia

Deliverable Title: Guidelines, rules and mechanisms governing the usage of the SoftFIRE Testbed

Deliverable Number D3.3

Keywords: NFV, SDN, 5G



Executive Summary:

SoftFIRE is an experimental federated Testbed aiming at nurturing an ecosystem of organizations willing to extend, consolidate and possibly industrialize solutions in the realm of NFV/SDN solutions with a specific reference to their adoption in 5G architectures.

In order to take advantage of SoftFIRE, two kinds of organizations could use the platform:

- Those selected by means of the Open Calls mechanisms
- Those interested in using the SoftFIRE infrastructure independently from the Open Calls and on the basis of precise purposes of the organization.

From a programming perspective the two kinds of requests will not differ too much and they will go through substantially the same guidelines and mechanisms. What will be different are the timeframe of the experimentation and a different level of project support.

This document extends Deliverable D3.2 (Handbook: Guidelines and Rules for on-demand access to the SoftFIRE Testbed) and presents a more detailed description of the SoftFIRE lifecycle for experimenters with specific focus on the SoftFIRE Software Portal and the OpenVPN experimenter access. It provides hints on the level of support that SoftFIRE will generally allocate to organizations participating to the open calls. It also describes limitations and constraints for the general use of the platform.

The focus of the document is in providing to programmers and users of the platform a view on mechanisms and possibilities offered by the Federated testbed and guide them through the expected lifecycle of a typical experiment.

In section 2, the document provides a description of the different components testbeds that form the SoftFIRE Federated platform. Section 3 provides a view of the tools that allow the SoftFIRE testbed to be accessible, manageable and integrated with the FRE infrastructure. Section 4 describes the intended lifecycle of an experiment: the design and the programming of the services and applications; the deployment of software components on the platform; the execution of the applications within the SoftFIRE Framework; the troubleshooting of the software; the security and monitoring of the experiment execution; and the closing and withdraw of the experiment. Section 5 presents some tools and mechanisms that can help the programmers during the experimentation phase and, finally section 6 describes some constraints and limitation in the usage of the SoftFIRE Federated testbed.

This document is an essential source of information for people that want to actually use the SoftFIRE Federated Testbed. The platform is under construction and its usage and set of tools and mechanisms will change in the course of the project life. This document is to be intended as a living document that the project will keep updated whenever substantial novelties will be added. In addition, the project is keen to receive comments and suggestions from practitioners and experimenters that have actually used the platform for developing their own solutions. These suggestions will be particularly useful to extend and consolidate the access and usage of SoftFIRE framework.



Table of Contents

Table of Contents.....	6
List of Figures.....	8
List of Tables.....	9
1 Introduction	10
2 Architecture of the Federated test bed	11
2.1 The Federation of testbeds	11
2.2 The component testbeds	12
2.2.1. Ericsson lab.....	12
2.2.2. The TUB/Fokus Testbed	16
2.2.3. TIMlab.....	18
2.2.4. University of Surrey (UoS).....	18
2.2.5. DT.....	26
2.2.6. Other Testbeds	28
2.3 The individual offerings of Testbed (services and virtual network functions)	28
3 The FIRE approach to accessing and using the Testbeds	29
3.1 Slice Based Federation Architecture.....	29
3.2 FITeagle	29
3.3 Open Baton.....	30
3.4 jFed Experimenter Client.....	31
3.5 OMF 6 – Federated Resource Control Protocol.....	32
3.6 OML	33
4 The experimenter’s Life Cycle.....	33
4.1 Experiment Lifecycle.....	33
4.2 Design and Programming	35
4.3 Access to the Federated test bed.....	36
4.4 Allocation of Resources.....	36
4.5 Running an experiment.....	38
4.6 Troubleshooting during the experiment.....	40
4.7 Security and Monitoring Tips.....	40



4.7.1.	Monitoring	40
4.7.2.	Security	40
4.8	Closing the experiment and feedbacks.....	41
5	SoftFIRE support to experimenters.....	42
5.1	Subscription	42
5.2	Case submission and followup.....	42
5.3	Workflow	43
5.4	Service Standards	44
6	SoftFIRE General Use.....	45
6.1	Constraints on the Experimenters Use	45
6.2	Additional support.....	46
7	References	47
8	List of Acronyms and Abbreviations	48



List of Figures

Figure 1: The SoftFIRE structure for programmers	11
Figure 2: Ericsson Computing View	14
Figure 3: Ericsson OpenStack Components Organization	15
Figure 4: Ericsson Testbed Networking Layout.....	15
Figure 5: FOKUS Testbed overview	16
Figure 6: UoS Testbed Architecture	21
Figure 7: UoS Testbed Addressing.....	24
Figure 8: UoS Network Connectivity for ETE LTE Service.....	25
Figure 9: MultiRAT Testbed – Logical overview	27
Figure 10: SFA Architecture.....	29
Figure 11: 5G Orchestration Architecture.....	30
Figure 12: Open Baton platform	31
Figure 13: OMF.....	32
Figure 14: FRCP Sequence Diagram.....	33
Figure 15: Experiment Lifecycle	35
Figure 16: jFed Login Screen.....	36
Figure 17: jFed - New Experiment.....	37
Figure 18: 5G RSpec.....	37
Figure 19: jFed 5G Topology.....	38
Figure 20: jFed Terminate Topology.....	41
Figure 21: Redmine home for SoftFIRE.....	42
Figure 22: Issues tracking	42
Figure 23: Describing a New Issue.....	43
Figure 24: Process for tracking issues.....	44



List of Tables

Table 1: Ericsson Hardware Structure of the TestBed.....	13
Table 2. Fokus OpenStack Hardware.....	17
Table 3: UoS Test-bed RAN Capabilities	19
Table 4: Network Equipment.....	20
Table 5: Network Services Provided By UoS Testbed.....	22
Table 6: UoS User plane NS Slice User IP Addressing.....	23
Table 7: NS Slice Addressing.....	24
Table 8: UoS Support Contacts	26



1 Introduction

SoftFIRE is building a federated experimental platform aimed at the construction and experimentation of services and functionalities built on top of NFV and SDN technologies. The platform is a loose federation of already existing testbed owned and operated by distinct organizations for purposes of research and development.

SoftFIRE intends to offer the opportunity to use the federated environment in order to allow to the vaster ecosystem possible the creation of services as well as the functional extension of the platform itself.

SoftFIRE has three main claims: supporting interoperability, programming and security of the federated testbed. Supporting the programmability of the platform is then a major goal.

The objective of this document is to facilitate the usage of the testbed to third parties and to monitor and govern the access of resources to them during the programming phase and the execution phase.

In this document, rules and mechanisms that ease the access to functionalities of the federated testbed are presented and exposed to programmers in order to ease the access, the programming and the usage of SoftFIRE.

The approach used by the project is to work with different testbeds in order to figure out a set of common available functionalities to describe and offer externally. They allow a uniform access and govern of the federated testbed. The project also tried to implement a first minimal set of FIRE requirements to be fulfilled in order to allow the access to the platform. The tools that allow this are FITEagle (FITEagle) and Open Baton (OpenBaton). The programmers and experimenters should devote some time to familiarize with them. In addition the document presents the initial design and functions of the SoftFIRE Software Portal and the usage of the OpenVPN in order to access to the system and interact with its functions and images.

The different component testbeds do offer distinctive capabilities. Along the project development period they will progressively be made available to programmers. In such a way the platform will be enriched and made more suitable for complex developments related to NFV/SDN technologies and with a perspective to 5G.

This document also describes the individual testbeds. In such a way Experimenters could know the underlying infrastructure and understand how to take advantage of it. In perspective this could also be used in order to plan the interworking with other platforms.

The document describes also the approach used in order to be compliant with the FIRE framework and provides a description and hints on the life cycle of the experimentations on the federated testbed. This is clearly the most useful part of the document for programmers and coders.

The last parts of this document are devoted to the support provided by SoftFIRE to the experimenters and the limitation and constraints that do exist in order to use SoftFIRE.

This document is intended to be a living document and it will be extended and improved along the time while the project improves the federated testbed and acquires more knowledge on the experimenters' and programmers' needs.



2 Architecture of the Federated test bed

2.1 The Federation of testbeds

SoftFIRE federates five European testbeds owned by the partners of the project. These testbeds are:

- *RMED Cloud Lab* from Ericsson, located in Rome;
- FUSECO Playground from FOKUS Fraunhofer/TUB, located in Berlin;
- JoLNet from TIM, spread over several Italian cities;
- 5GIC from University of Surrey, located in Guildford, Surrey;
- Deutsche Telekom.

Secured links (IPsec) over the Internet interconnect the testbeds data and control plane.

Experimenters can access the available resources through a single access-point, i.e., the *FiTeagle* framework, Figure 1. FiTeagle provides primitives to authenticate users and to discover, reserve, acquire, monitor and finally release a set of arbitrary resources of the infrastructure. Once a user has been given the authorization to access the system, he can perform experiments on top of the architecture for a certain amount of time. FiTeagle ensures interoperability with other technologies by implementing the standard FIRE SFA interface.

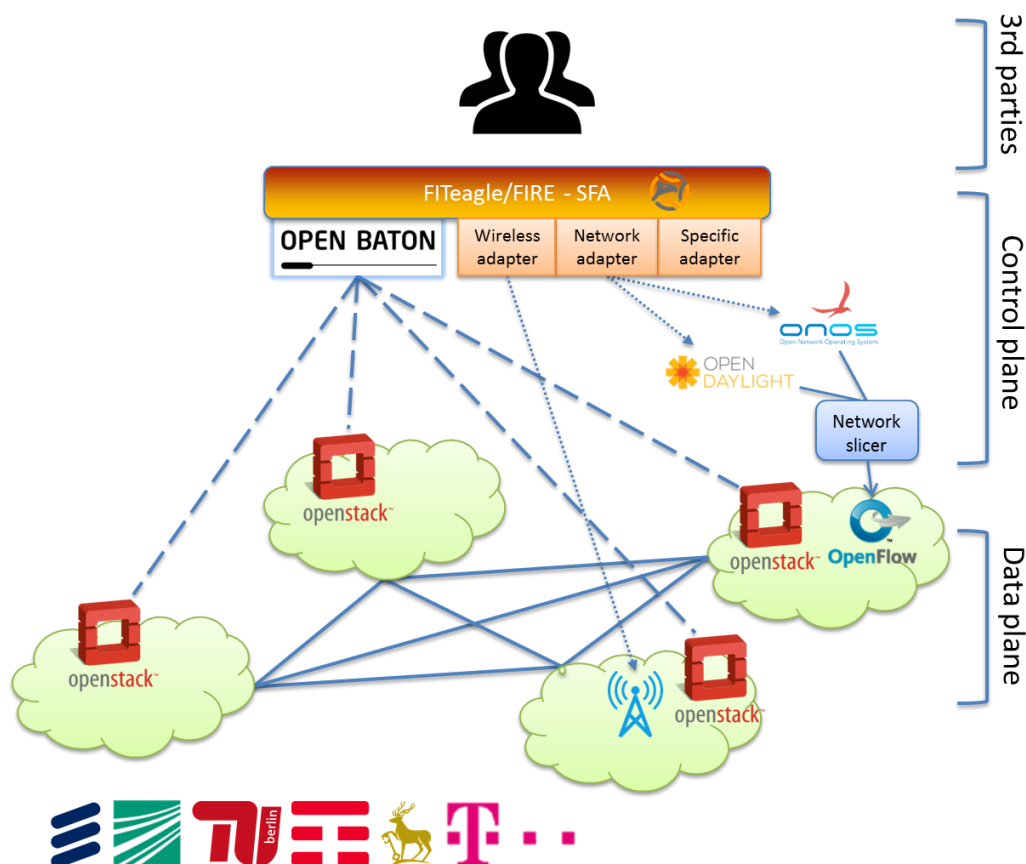


Figure 1: The SoftFIRE structure for programmers



FiTeagle interacts with the orchestrator that manages computing and networking resources of the testbeds, which is an instance of *OpenBaton* running on FUSECO. In fact, OpenBaton is a Network Function Virtualization Orchestrator (NFVO) that follows the ETSI NFV Management and Orchestration (MANO) specification and allows users to define and launch virtualized instances and to connect them through a set of virtualized networks. In addition, it provides auto scaling and fault management based on monitoring information coming from the monitoring system available at the NFVI level.

Each testbed provisions virtual resources by means of an *OpenStack* (OpenStack) cloud controller that controls the physical compute, storage, and networking resources dedicated to the project. This OpenStack controller is connected to the central OpenBaton orchestrator, which coordinates the instantiation of virtual machines (VMs) and containers over the testbeds. Exploiting the *Regions* and *AvailabilityZones* defined in OpenStack, experimenters can choose either the testbed or specific locations within the testbeds in which they want to instantiate VMs and so the architecture can simulate peculiar scenarios including the interaction of different domains inside one operator's network or among different operators. Though some testbeds own resources that could not be handled by means of the ETSI NFV framework (e.g., OpenFlow physical switches, wireless access points), FiTeagle is extended with specific *adapters* that manage those resources, which are then exported towards the experimenters.

2.2 The component testbeds

2.2.1. Ericsson lab

Ericsson SoftFIRE testbed is part of the Ericsson RMED CloudLab. Located in Rome, the Lab's scope is to provide Hands on and Competence Build-up, show specific and concrete "proof" points related to the cloud benefits, Customer Demo on specific products and demonstrate how issues and concerns can be managed mitigating the risks.

Main activities performed are:

- Standard Customer Demo
- Deep Dive on Customer specific request
- PoC on Customer premises
- Fully Customized PoC on Customer premises
- Validation and Certification on Customer specific stack / solution

The Ericsson Cloud Lab is a flexible environment where it is possible to combine different hardware configurations to support different delivery policies. The Lab is anyway adaptable to guarantee the Ericsson Platform requirements and commercial products.

Ericsson Testbed Architecture for SoftFIRE

Ericsson is sharing an experimental infrastructure (Ericsson SoftFIRE testbed) to be interconnected within the SoftFIRE project.

The scope of Ericsson testbed in SoftFIRE project is to provide an OpenStack Liberty in order to delivery an infrastructure as a service (IaaS) for creating and managing large groups of virtual private servers in a data center.



The architecture is based on Dell Hardware, as shown in the table below:

Table 1: Ericsson Hardware Structure of the TestBed

Description (single node configuration)	Qty
Dell PowerEdge R620	1
Intel Xeon E5-2660v2 2.2GHz, 25M Cache, 8.0GT/s QPI, Turbo, HT, 10C, 95W, Max Mem 1866MHz	1
Intel Xeon E5-2660v2 2.2GHz, 25M Cache, 8.0GT/s QPI, Turbo, HT, 10C, 95W, Max Mem 1866MHz, 2nd Proc	1
PowerEdge R620 Shipping - 4/8 Drive Chassis, EMEA1 (English/French/German/Spanish/Russian/Hebrew)	1
Chassis with up to 8 Hard Drives and 3 PCIe Slots, Low Profile PCI Cards Only	1
Bezel - 4/8 Drive Chassis	1
Performance Optimized	1
1866MT/s RDIMMs	1
16GB RDIMM, 1866MT/s, Standard Volt, Dual Rank, x4 Data Width	8
Heat Sink for PowerEdge R620	2
DIMM Blanks for Systems with 2 Processors	1
400GB, SSD SAS Value SLC 6Gbps, 2.5in Hard Drive (Hot-plug)	2
PERC H310 Integrated RAID Controller	1
Active Power Controller BIOS Setting	1
DVD ROM, SATA, Internal	1
Dual, Hot-plug, Redundant Power Supply (1+1), 750W	1
2M Rack Power Cord C13/C14 12A	2
Cable for Mini PERC Cards for Chassis with up to 8 Hard Drives	1
Intel X520 DP 10Gb DA/SFP+ Server Adapter, Low Profile	2
Intel Ethernet i350 QP 1Gb Network Daughter Card	1
PowerEdge R620 Motherboard, TPM	1
2/4-Post Static Rails	1
RAID 1 for H710p, H710, H310 Controllers	1
iDRAC7 Enterprise	1
10Gb LR SFP+ modules	2

The number of servers reserved for the project are three: one controller and two compute nodes Figure 2: Ericsson Computing View **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..**



From storage point of view, all servers are equipped with 2x 400 GB disk in mirroring for OS and 2 additional disks by 2TB each one, also in mirroring.

In the controller 1 TB is used for Cinder and 1 TB is used for Glance; in each compute node 2 TB are reserved for Nova.

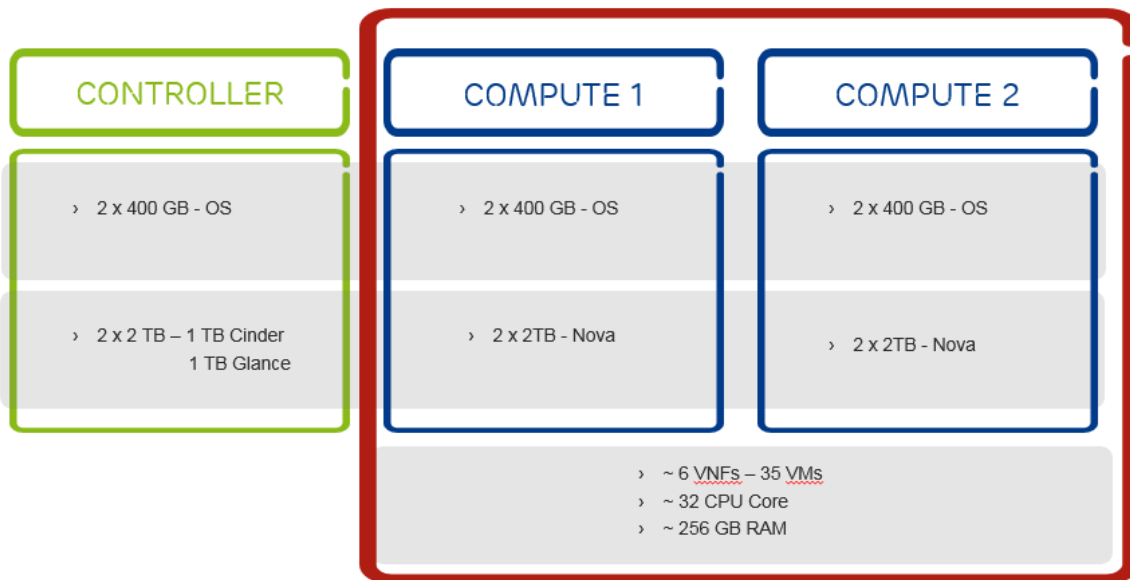


Figure 2: Ericsson Computing View

The installed OpenStack has a classical modular architecture where main components are (Figure 3: Ericsson OpenStack Components Organization):

Nova - provides virtual machines (VMs) upon demand.

Cinder - provides persistent block storage to guest VMs.

Glance - provides a catalogue and repository for virtual disk images.

Keystone - provides authentication and authorization for all the OpenStack services.

Horizon - provides a modular web-based user interface (UI) for OpenStack services.

Neutron - provides network connectivity-as-a-service between interface devices managed by OpenStack services.

Ceilometer - provides a single point of contact for billing systems.

As for Ceilometer, it is not integrated into the overall picture due to Zabbix takes over it.

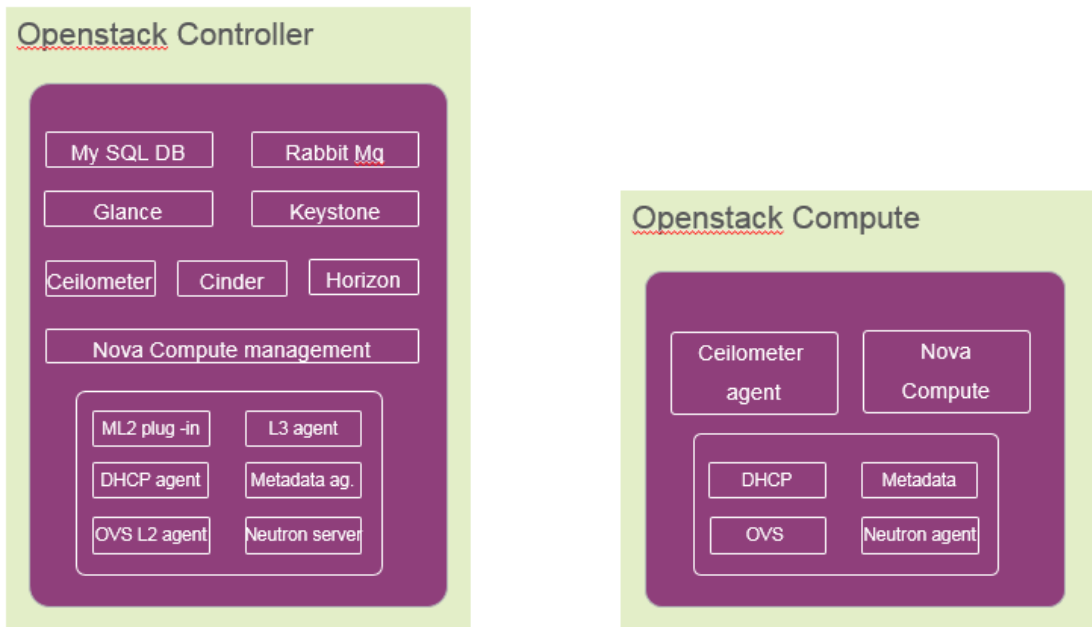


Figure 3: Ericsson OpenStack Components Organization

Network Design

The network design of the proposed architecture is composed by four networks (Figure 4: Ericsson Testbed Networking Layout):

- IDRAC network: is the console network;
- Tunnel network: used for internal communication between the servers;
- Management Network: used for management and API;
- Floating network: used for expose the VM to external

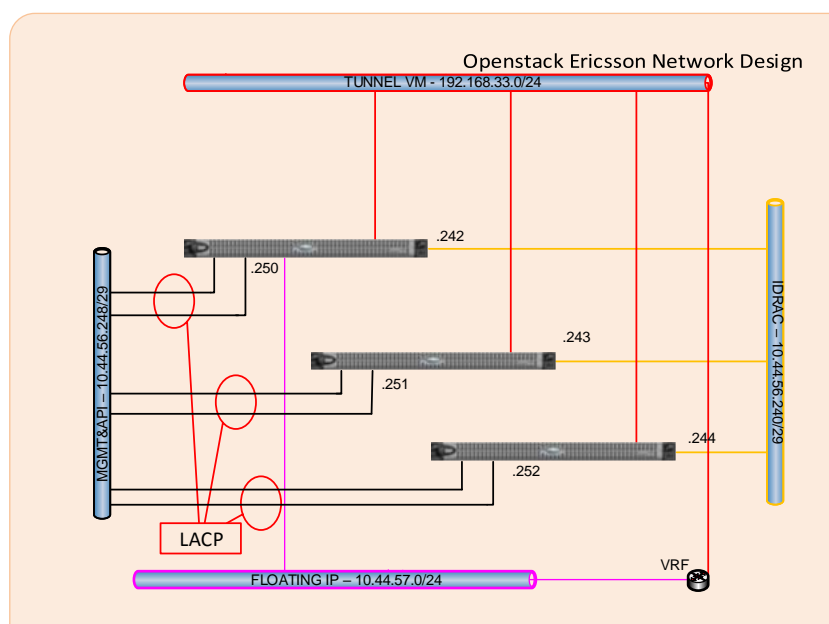


Figure 4: Ericsson Testbed Networking Layout



2.2.2. The TUB/Fokus Testbed

The SoftFIRE Testbed node located at Fraunhofer FOKUS in Berlin is realized as a slice of a much bigger Testbed that is used to benchmark virtual 5G core network functions, Figure 5: FOKUS Testbed overview. This includes a dedicated part (Tenant) of an OpenStack cluster to provide computing and storage resources. The connectivity to the distributed parts of the SoftFIRE Testbeds is realized by IPsec secured VPN links. The Fraunhofer FOKUS node is currently the center of the VPN network. It is realized as a pure virtual network that is feed as VLAN into the two Virtualization environments at FOKUS. The VPN encryption is handled by a virtual instance of an OpenBSD based Firewall running on a different VMware based virtualization cluster.

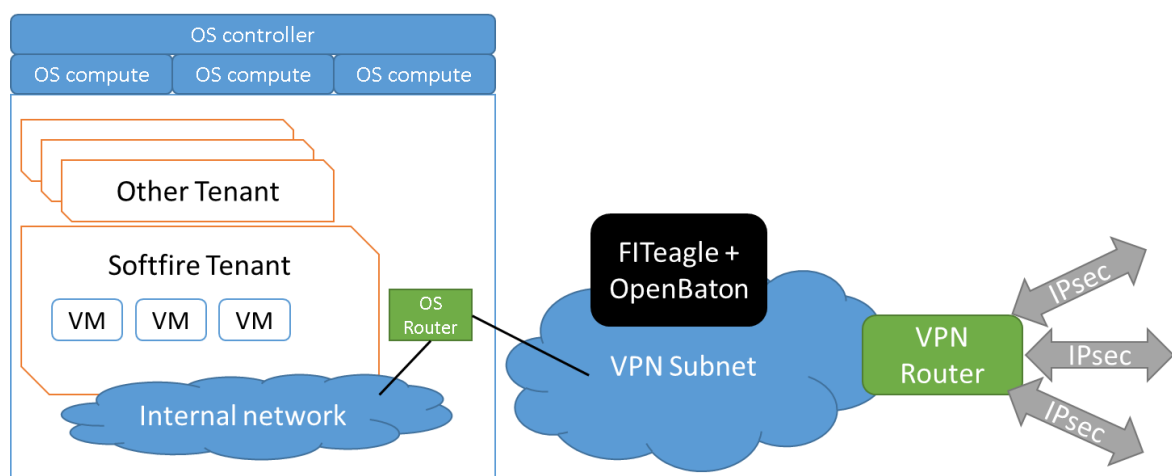


Figure 5: FOKUS Testbed overview

OpenStack Cluster

The compute resources are provided by an OpenStack Liberty cluster installation that will be shared between different Projects to reduce the administration overhead to keep the Openstack components up to date. However, the SoftFIRE project is guaranteed a fixed slice of compute and storage capacity. The setup is based on one combined controller and networking host and currently three compute nodes. The used servers are manufactured by Dell and are in the Blade form factor. Table 2 lists the details of the used server hardware. Storage Capacity is provided by a central Storage Array Network (SAN) manufactured by NetApp accessed via 10Gbit Ethernet. The Servers are connected to several redundant networks that are used for management and storage access. Direct access to these networks is not possible from within the VM instances. Connectivity for the SoftFIRE VPN is realized as an external provider network that is dedicated to the SoftFIRE project.

**Table 2. Fokus OpenStack Hardware**

	Type	RAM	CPU	Storage
Controller	Dell PowerEdge M630	128 GB	2x Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz (8cores)	2x 200GB SSD RAID
Compute 1	Dell PowerEdge M630	128 GB	2x Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz (8cores)	2x 200GB SSD RAID NAS: 860 GB
Compute 2	Dell PowerEdge M630	128 GB	2x Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz (8cores)	2x 200GB SSD RAID NAS: 860 GB
Compute 3	Dell PowerEdge M630	128 GB	2x Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz (8cores)	2x 200GB SSD RAID NAS: 860 GB

VPN Hub

Interconnection between FOKUS and the other SoftFIRE Testbeds is provided by a virtualized IPsec VPN server. An OpenBSD based firewall was chosen because of the great flexibility in supported VPN and firewall settings that are needed to interconnect the different testbeds. The networks are completely isolated from the internal network of the Institute. Incoming and outgoing network traffic is filtered based on whitelists that allow previously agreed protocols. The VPN Hub is capable of forwarding traffic between different SoftFIRE networks at different Partners. However, due to the limited Internet bandwidth at Fraunhofer FOKUS this feature is enabled with reduced bandwidth. In a later phase the VPN hub is moved to the TUB that provides a much faster Internet connectivity. The Second VPN Server located at TUB will provide OpenVPN services to registered Experimenters and IPsec interconnection to other Testbeds. Access to the OpenVPN server is protected by the same Certificates that are used to access the jFed client. This VPN server provides network access to registered experimenters so they can directly interact with their own NVFs.

Orchestration and Management Services

The resources of the Federated SoftFIRE testbed are managed via FITeagle and OpenBaton toolkits. These services are the single contact point to the experimenters. To ensure good connectivity to all SoftFIRE testbeds these services are installed at Fraunhofer FOKUS as virtual services. For security reasons these services were installed on a different virtualization environment that cannot be accidentally modified by the SoftFIRE components. The second virtualization environment is realized by a VMware ESXi cluster. A Zabbix (Zabbix) proxy service is provided on the same instance to support the collection of KPI values.

To allow the Experimenters to upload custom Virtual Machine images that can be used in any of the distributed SoftFIRE testbeds an upload Portal is provided. Due to Bandwidth consumption this service is located in the TUB datacenter.



2.2.3. TIMlab

JoLNet (TIM Lab) is a geographical-distributed experimental network based on SDN/NFV paradigms. It exploits OpenFlow capable switches and COTS servers in order to offer flexibility and programmability, allowing experimenters to test novel network architectures along with the issues deriving from “real” operators’ networks. In fact, JoLNet consists of seven points of presence (PoP) located in Universities and Research Centers spread over the Italian country and logically connected as a full mesh. Each PoP includes two switches, namely CPE and node; the node connects to all the other nodes of the other PoPs. JoLNet exports an OpenFlow API and an OpenStack API that experimenters can use to program the behavior of the network and to instantiate virtual machine acting as network services and users for testing purposes. Leveraging a slicing mechanism, JoLNet hosts multiple isolated experiments at a time on top of the same infrastructure.

Researchers who want to start experimenting have to contact JoLNet technical support at supporto.jolnet@telecomitalia.it, submitting a request that includes the purposes of the research and technical details of the settlement. Experimenters must connect to Telecom Italia network from one of its venues or by means of a VPN connection. Each experiment will be given a network slice that consists of one or more VLAN tags and a tenant account of OpenStack with certain amount of virtual resources (vCPU, RAM, disk space). Experimenters can choose to use their OpenFlow network controller or to ask for one that will be provided by JoLNet itself and will be put on top of the given slice. In the former case researchers must inform the technical support about the address (IP:port) of the controller to properly configure the switches to connect to, in the latter the experimenters will use the APIs of the already provided controller (OpenDaylight (OpendayLight) or ONOS (ONOS)). Network isolation among experiments is guaranteed by means of the slices (i.e., the VLAN tags), hence each experimenter can use only the assigned VLAN tags, otherwise the system will reject the requests. Experimenters can manage the provided resources through the APIs available on the controllers (e.g., CLI, REST, GUI, ...) and using the different OpenStack regions, experimenters can launch VM instances on specific PoPs and attach them to the either node or the CPE exploiting the correspondence between OpenStack networks and VLAN tags of slices.

2.2.4. University of Surrey (UoS)

The UoS SoftFIRE testbed segment is part of the overall UoS 5GIC testbed. Located in the UK, the scope of the lab scope is to provide hands-on access to a 3GPP based campus RAN with indoor and outdoor coverage that is able to be interconnected with a variety of virtualized core slices, in order to develop Core Network 5G evolutions and demonstrate 5G Use Cases running over the resultant ETE cellular network. In this manner the testbed can be used to build industry core competence in 5G.

The network was initially built as a fixed ETE cellular system, but has now been evolved to provide a set of virtualized network capabilities that can be configured to connect with IP stubs towards the RAN to enable various network slices to be connected in circuit under the control of the federated SoftFIRE core.

It is envisaged that experimenters using the facilities of the UoS SoftFIRE testbed will be able to show specific and concrete “proof” points related to the 5G RAN and Core evolutions and demonstrate applications running over this infrastructure. These use case proof points can be used to support many types of use cases to highlight their benefits, explain to customers and



industry partners how they work and demonstrate how 5G targets may be met, and what the pros and cons are for each demonstration.

The Main activities performed are:

- Standard experimenter Demo
- Deep Dive on experimenter specific request
- PoC whilst connected to experimenters equipment or remote site
- Validation and Certification on Customer specific solution

The UoS testbed is a flexible environment where it is possible to combine different RAN hardware configurations and RAN and Core software configurations to support different delivery designs and policies. There is also scope to evolve the testbed in cooperation with a given experimenter and/or customer.

UoS Testbed Architecture for SoftFIRE

The 5GIC UoS testbed is sharing a segment of the testbed with the SoftFIRE Federated Testbed (UoS SoftFIRE testbed segment), which is interconnected within the SoftFIRE project.









The scope of UoS SoftFIRE testbed segment in the SoftFIRE project is to provide the following component parts:

- 1) OpenStack Liberty system access to infrastructure that can be used to instantiate core slices for experimentation with the local RAN components.
- 2) Access to a segment of the 5GIC Testbed outdoor campus based LTE-A RAN
- 3) Access to the 5GIC in-building LTE-A RAN
- 4) Access to the 5GIC in-building Wi-Fi system for multi-access 5G use cases

The RAN, LTE-A and Wi-Fi capabilities provide by the UoS Testbed are illustrated in the following **Fehler! Verweisquelle konnte nicht gefunden werden..**

Table 3: UoS Test-bed RAN Capabilities

Radio Access Network(s)

Site Type	# off Sites	# off Cells	Access Type	Band	BW (MHz)	Mode	Handover	Mobile	CA	UL Mbps Typ(Max)	DL Mbps Typ(Max)	Notes
Outdoor 2xSector	14	28	LTE-A	B38	20	TDD	Yes	CAT6	Yes	10(18)	100(220)	
Outdoor Omni	5	5	LTE-A	B38	20	TDD	Yes	CAT6	Yes	10(18)	100(220)	
Indoor Lampsites	6	6	LTE-A	B41	20	TDD	Yes	CAT6	No	7(18)	75(100)	3xLS/per floor, @ ground & 2nd, 1st has none
Indoor AP	6	6	Wi-Fi	2.4GHz	30	N/A	none	N/A	N/A	400	400	3xAP/per floor, @ ground & 2nd, 1st has none

In order to deliver the infrastructure as a service (IaaS) capabilities for creating and managing as a data-centre, several server resources are provided. The number of servers reserved for the project is five: one controller, one compute node, one image/package resource server, one license server and one breakout server.



The following network hardware is provided, as shown in Table 4 below:

Table 4: Network Equipment

Description (single node configuration)	Qty
Dell PowerEdge R920 (deployed as SoftFIRE “uos-compute” server) Intel Xeon based processors based on E7-4800 v2, 32GB RAM 1600MT/S 4 x 500 GB HDD, 90 Core on platform, up to 24 cores available for SoftCORE, platform dedicated to SoftFIRE 8 Ethernet ports	1
Dell PowerEdge R430 (deployed as SoftFIRE “uos-admin” server) Intel Xeon based processors based on E5-2609 1.9GHz, 16GB RAM 1600MT/S 2 x 500 GB HDD, 16 Core on platform, dedicated to SoftFIRE 2 Ethernet ports	1
Dell PowerEdge R430 (deployed as SoftFIRE “uos-resource” server) Intel Xeon based processors based on E5-2609 1.9GHz, 16GB RAM 1600MT/S 2 x 500 GB HDD, 16 Core on platform, dedicated to SoftFIRE 2 Ethernet Ports	1
Dell PowerEdge R430 (deployed as SoftFIRE “uos-licence” server) Intel Xeon based processors based on E5-2609 1.9GHz, 16GB RAM 1600MT/S 2 x 500 GB HDD, 16 Core on platform, dedicated to SoftFIRE 2 Ethernet Ports	1
Dell PowerEdge R430 (deployed as SoftFIRE “uos-web-bo” server) Intel Xeon based processors based on E5-2609 1.9GHz, 16GB RAM 1600MT/S 2 x 500 GB HDD, 16 Core on platform, dedicated to SoftFIRE 2 Ethernet Ports	1
Cisco Nexus 3048 SDN Switches	3
Indoor, Wi-Fi Access Points Wi-Fi 802.11ac Access Points from Aruba	6
Indoor, LTE-A, TDD, picoRemote Radio Heads (picoRRH) Huawei C-RAN pRRH Lampsites	6
Indoor, LTE-A, TDD, Remote Radio Heads (RRH) Huawei 3256 C-RAN, RRU	33
C-RAN Baseband Units BBU's (LTE-A eNB) Huawei 33900 C-RAN BBU <ul style="list-style-type: none">- 14 outdoor base sites with 2 x sector- 5 outdoor baes sites with 1 x sector- 6 indoor base sites, omni (1 sector)	16

The network infrastructure is illustrated in the following diagram, Figure 6: UoS Testbed Architecture

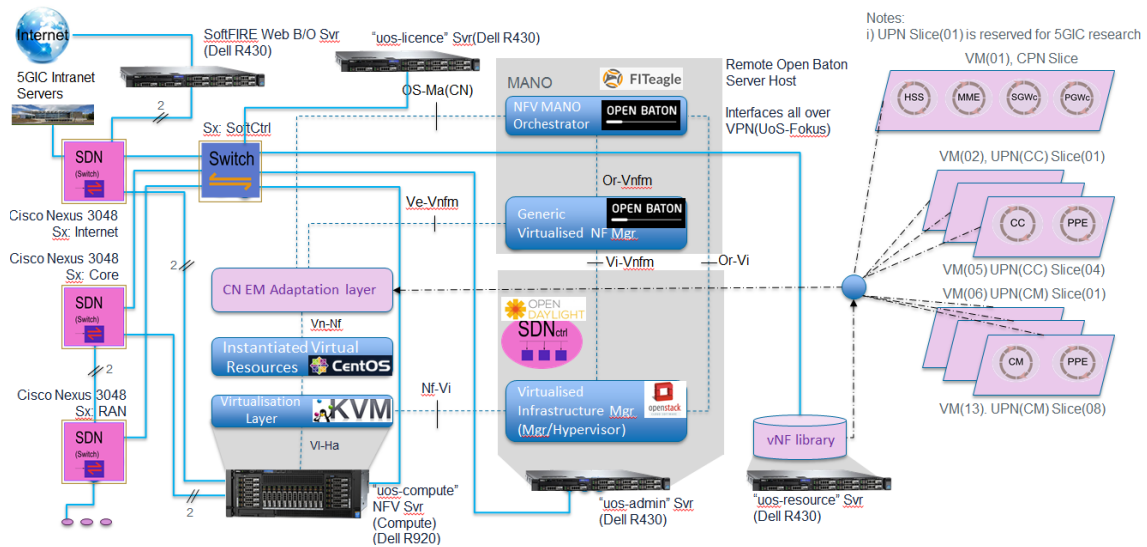


Figure 6: UoS Testbed Architecture

The installed OpenStack has a classical modular architecture where main components are:

- Nova** is the interface between the OpenStack uos-admin server instance and the uos-compute server Virtual Machines (VMs) for instantiation of VMs and images and/or packages instantiated on each VM.
- Glance** is the interface between the OpenStack uos-admin server instance and the uos-resource server for access to the UoS catalog of software images and packages
- Keystone** is the interface between the OpenStack uos-admin server instance and the OpenBaton Orchestrator located at TUB/Fraunhofer for overall federated testbed control on authenticated and authorized basis for all OpenStack services.
- Neutron** – is the interface between the OpenStack uos-admin server instance and the OpenStack Open Virtual Switch (OVS) at the uos-compute server that enables the OpenStack instance to configure mappings between virtual OpenStack internal IP addresses and externally presented IP addresses in order to connect up the testbed Virtual Network Functions (vNF).

Network Service Slices Provided by the UoS Testbed

The UoS testbed provides several LTE-A Core network software Images and/or packaged components for experimenters to use, as in Table 5:

**Table 5: Network Services Provided By UoS Testbed**

Network Service	Max # of instances on the UoS Testbed	vNF's included in Network Service	Description
CPN	1	HSS, MME, integrated(SGWc, PGWc)	<p>This Network Service (NS) slice is instantiated as soon as any EPC is required to be instantiated on the UoS testbed for Control Plane connectivity via the UoS LTE-A RAN.</p> <p>There is only ever one instance of the CPN NS for the whole UoS SoftFIRE network Segment. All experimenters instantiated on the UoS testbed share this slice for LTE-A EMM connectivity.</p> <p>The PLMN-id used is 235 91 which is a Vodafone UK test id with label "5GIC"</p>
UPN(CC)	4	CC, integrated(SGWu, PGWu)	<p>This Network Service is instantiated per experimenter for LTE User Plane service and extended 5G Context Awareness Association to a group of cells known as a "Cluster" via the newly proposed 5G node called a Cluster Controller (CC).</p> <p>This slice provides best performance for Internet access.</p> <p>Each authorized experimenter is enabled for instantiation of one of these NS, network slices on the UoS testbed.</p>
UPN(CM)	8	CM, integrated(SGWu, PGWu) = a.k.a. "PPE"	<p>This Network Service is instantiated per experimenter for LTE User Plane service and extended 5G Context Awareness Association to a Cluster Member within a "Cluster" via the newly proposed 5G node called a Cluster Member (CC).</p> <p>This slice provides best performance for Intranet access via one of the Cluster Members.</p> <p>Access to a local server instance that the experimenter may wish to deploy at the edge of the LTE-A network for optimum latency performance is enabled with instantiation of this slice. Otherwise, the experimenter would need to deploy their test network server application north of the PGWu, which is less optimal for test server access from the User Equipment (UE, Mobile).</p> <p>Each authorized Experimenter is enabled for instantiation of up to two of these NS, network slices on the UoS testbed.</p>



Network Design

The network design of the proposed architecture is composed with 2 key subnets networks:

- RAN Sub-net, 10.5.20.x
- Core Sub-net, 10.5.20.21

A VPN is provided to connect the OpenStack Keystone interface from the local UoS OpenStack instance to the TUB/Fraunhofer located OpenBaton orchestrator, which enables control of the whole Federated Testbeds VNF infrastructure.

User Addressing

The User Address space is divided into 4 ranges on the RAN Sub-net. One range of 25 IP addresses is provided for each experimenter tenant. The slices are named as follows:

- CC(X) Control Plane Slice X
- UPN(A) User Plane Slice(A) to (D)
- UPN(A) User Plane Slice(A1), (A2), to (D1), (D2)

Each experimenter operates the following IP address ranges for their devices camping on the network, as follows in **Fehler! Verweisquelle konnte nicht gefunden werden.:**

Table 6: UoS User plane NS Slice User IP Addressing

NS Slice	APN	IP start Address	IP end Address
CC(A), CM(A1), CM(A2)	aSoftFIRE	10.5.20.101	10.5.20.125
CC(B), CM(B1), CM(B2)	bSoftFIRE	10.5.20.126	10.5.20.150
CC(C), CM(C1), CM(C2)	cSoftFIRE	10.5.20.151	10.5.20.175
CC(D), CM(D1), CM(D2)	dSoftFIRE	10.5.20.176	10.5.20.200
Expansion planning ...	N/A	10.5.20.201 to 254 = Reserved	N/A

An illustration of the high level IP addressing of the UoS testbed is provided in the following Figure 7: UoS Testbed Addressing.

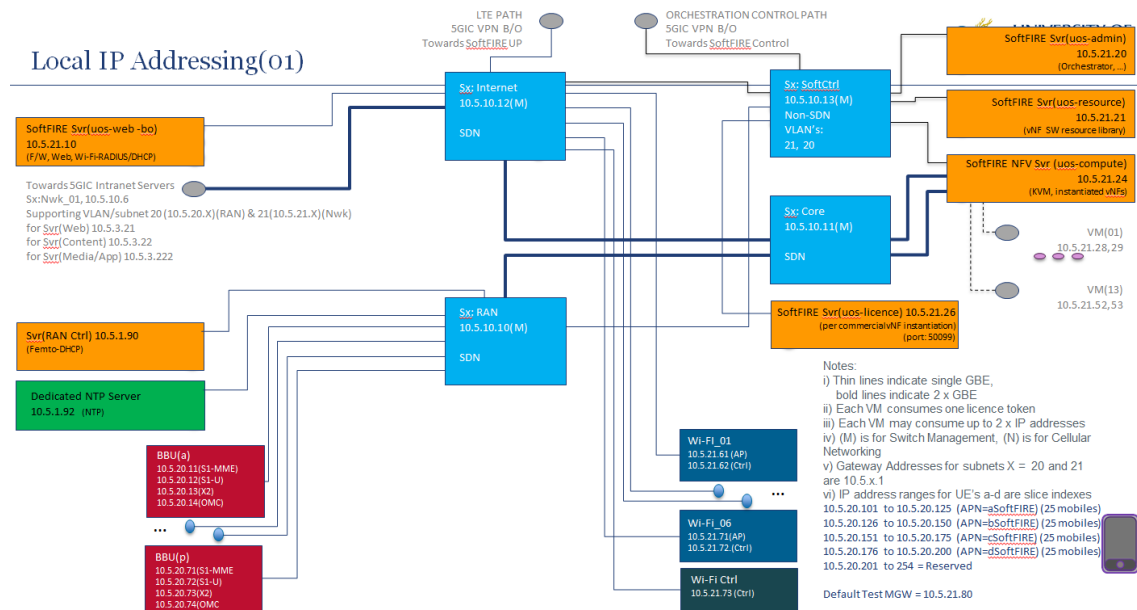


Figure 7: UoS Testbed Addressing

The RAN addressing is fixed, and IP address stubs are provided for connecting the vNF user plane NS, slice instances across the range 10.5.21.28,29 to 10.5.21.52.53 as follows in Fehler!

Verweisquelle konnte nicht gefunden werden.:

Table 7: NS Slice Addressing

NS Slice	vNF	IP Addresses	Description
CPN(X)		10.5.21.28	Single Control Plane NS
CC(A)	CC PPE	10.5.21.30 10.5.21.31	
CM(A1)	CM PPE	10.5.21.32 10.5.21.33	
CM(A2)	CM PPE	10.5.21.34 10.5.21.35	
CC(B)	CC PPE	10.5.21.36 10.5.21.37	
CC(B1)	CM PPE	10.5.21.38 10.5.21.39	
CC(B2)	CM PPE	10.5.21.40 10.5.21.41	
...			
CM(D)	CC PPE	10.5.21.48 10.5.21.49	
CM(D1)	CM PPE	10.5.21.50 10.5.21.51	
CM(D2)	CM PPE	10.5.21.52 10.5.21.53	
Expansion			



NS Slice	vNF	IP Addresses	Description
planning ...			

Experimenter Application Support

A further IP address range is available for experimenters to instantiate their own applications at the PGW, CC or CM level, as 10.5.21.81 to 10.5.21.101 as follows, with approximately 5 addresses per slice.

If the experimenter would like to design applications that benefit from additional Context Aware (CA) information capabilities available in the UoS testbed, then please address the UoS support contacts detailed in this document.

For support on developing CA applications that operate at the edge of the LTE-A RAN (MEC-like) on the UoS testbed please contact support UoS support so that port numbering, IP addressing and configuration of the 5G Context Awareness evolutions can be agreed.

Network Connectivity

A visualization of the connectivity and flows that need to be instantiated to configure an ETE LTE network on the UoS test-bed, is illustrated as follows for the Uplink (UL) flows in Figure 8: UoS Network Connectivity for ETE LTE Service.. A similar set of configuration of flows and connections needs to be setup for the Downlink as well (not illustrated here).

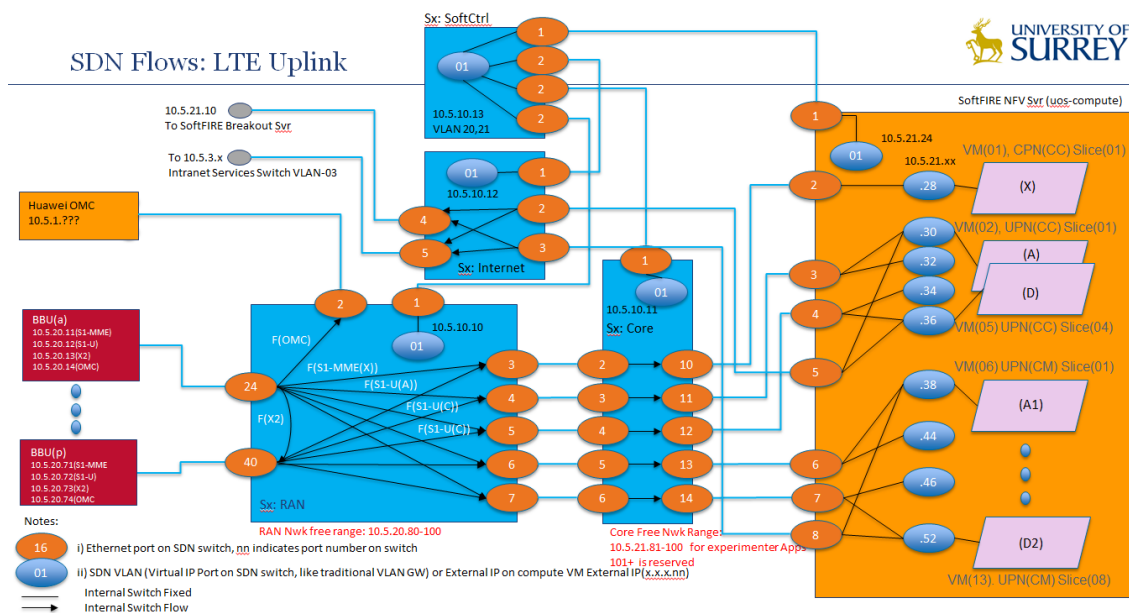


Figure 8: UoS Network Connectivity for ETE LTE Service.

Intranet Design

The UoS testbed provides intranet connectivity services for testing as follows, for web, content and media services. Experimenters are encouraged to discuss with the UoS support team how to make best use of these services, as required.



Svr(Web) 10.5.3.21 (Web)

Svr(Content) 10.5.3.22 (FTP)

Svr(Media/App) 10.5.3.222

Mobile Device Support

The UoS testbed provides some test mobile services and can support remote service activation on mobiles carried by staff at the university by arrangement. It is intended to develop this capability so that it can be exposed via FITeagle for experimenters, however it is always easier to manage mobiles directly when notable amounts of debug are required when developing ETE LTE/5G capabilities involving Mobile application development.

The UoS recommends use of mobiles for live testing that support the following basic features:

Aspect	Specification	Comment
OS	Android Lollipop 5.x.x preferred	Easier to develop research code with more parameters exposed than iOS
LTE Bands	B38, B41	These bands are essential for outdoor and indoor operation at UoS.
Wi-Fi	802.11ac	The UoS has 802.11ac and most previous generations of Wi-Fi support
Category	5,6	A minimum of category 5 is essential to support the Carrier Aggregated LTE-A RAN in order to get the best speeds from the deployed RAN.

UoS Testbed Additional Notes:

- i) The A slice is reserved for UoS experiments.
- ii) Contacts for UoS Testbed are as follows in **Fehler! Verweisquelle konnte nicht gefunden werden.:**

Table 8: UoS Support Contacts

Aspect	Contact
System	g.foster@surrey.ac.uk
IT/Enterprise	chris.clark@surrey.ac.uk
RAN	f.emntezami@surrey.ac.uk

2.2.5. DT

The Testbed of the Telekom Innovation Laboratories is called Multi Radio Access Technology (MultiRAT)-Testbed. It is planned to be a key testbed for innovation and to be an integral part



for selected topics within Deutsche Telekom and its subsidiaries. It is currently declared as an optional testbed part of the overall SoftFIRE project.

The DT node is located at T-Labs in Berlin and offers a network infrastructure basis for several internal and external projects. The current MultiRAT-Testbed allows access via 2G, 4G, Wi-Fi, and WiMAX. However, the main scope lies on network connectivity based on Wi-Fi.

Next to the physical opportunities, the MultiRAT-Testbed is currently starting to be interconnected with an OpenStack Environment (access level: tenant), offering cloud-based computing and storage.

However, the main and current activities of the testbed are to develop and investigate

- NFV/SDN functionality to address 5G Use Cases and scenarios,
- Support of Single and Multi-connection Seamless Mobility, as well as
- Network-Centric Location-based Services.

The MultiRAT-Testbed is an integrated testbed, which connects several access possibilities with a dynamic deployment of Wi-Fi access nodes - offering storage, processing and connectivity on each node.

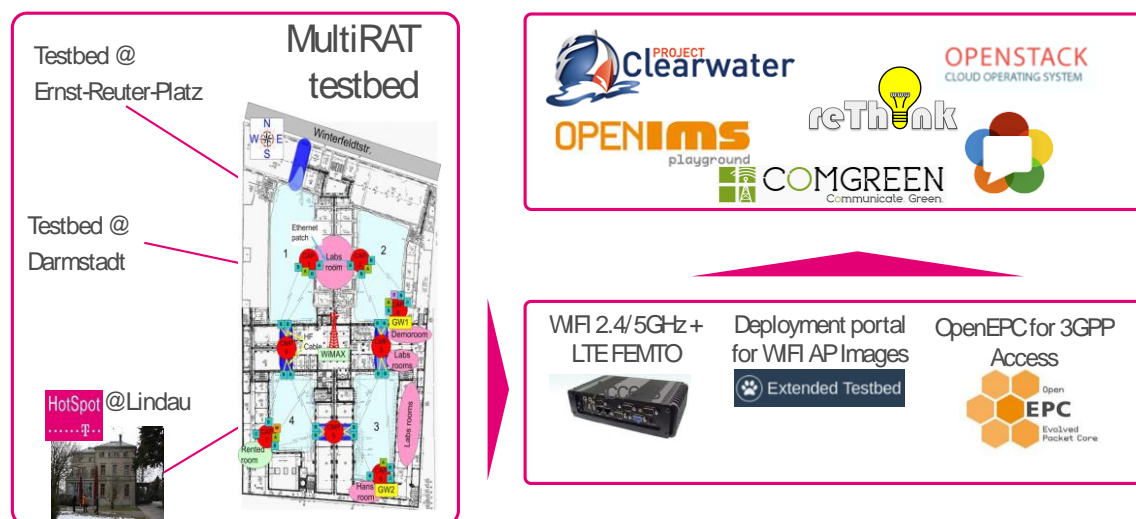


Figure 9: MultiRAT Testbed – Logical overview

Figure 9: MultiRAT Testbed – Logical overview shows the logical view of the DT testbed infrastructure. The building is equipped with around 40 Wi-Fi access nodes as well as one 3GPP LTE Femto cell – managed by the OpenEPC. For the dynamic deployment of Wi-Fi access point specific software, a Deployment Portal offers configurations with different levels of granularity (Project, Access Node, and Runtime based) to run newly developed access node software on those access nodes. These components offer connectivity to different services on top of the Testbed, such as several IMS instances, Web-based communication platforms (WebRTC and reThink), and to the newly connected OpenStack environment.

The access nodes have the following configuration:

- x86 64-bit architecture
- At least 2 physical Wi-Fi Interface Cards available (a/b/g/n/ac)



- Bluetooth
- Computing Power + Storage for Cloud-Edge Services (no hardware virtualization available!)
- External Remote Power Controller to enforce energy efficient configurations
- Around 45 Access Nodes at Location Winterfeldtstraße (1/3 is connected via fibre backhauling)

The access to the testbed is possible via visiting the location to integrate and run the developed software; however, monitoring and steering the developed applications might be possible via a remote VPN connectivity.

2.2.6. Other Testbeds

The SoftFIRE project is also involved in international activities. There are joint activities with other possible partner to create a wider federation of Testbeds. A first step towards it is to include in the federated testbed the EIT Digital Silicon Valley testbed. It could be the entry point for further collaborations with organizations in the USA, one example could be ON.Lab (ON.Lab).

2.3 The individual offerings of Testbed (services and virtual network functions)

Currently the project and its partners are experimenting some functionalities and related virtual machines. If they will achieve an acceptable level of stability and robustness, the federated testbed will be complemented with a set of well-formed and ready to use Network Functions. The programmers will be capable of using them in order to create services and applications taking advantage of more programmable building blocks.

The topics addressed are existing network architectures (like IMS and its evolution) and a special attention is given to initial building blocks for 5G. This will allow the programmers to exploit these functionalities and start design applications for the 5G environments.

In order to support the programmers, in case of a release of Network Functions, this document will be extended and will present a description of the basic functionalities, the APIs and a guide to use them.



3 The FIRE approach to accessing and using the Testbeds

3.1 Slice Based Federation Architecture

The Slice-based Federation Architecture (SFA) allows user authentication, and resource discovery, reservation and release. In the context of the European Union's Future Internet Research Initiative (FIRE), SFA has been adopted in Fed4Fire (FED4Fire) and other projects dealing with federated testbeds and infrastructures.

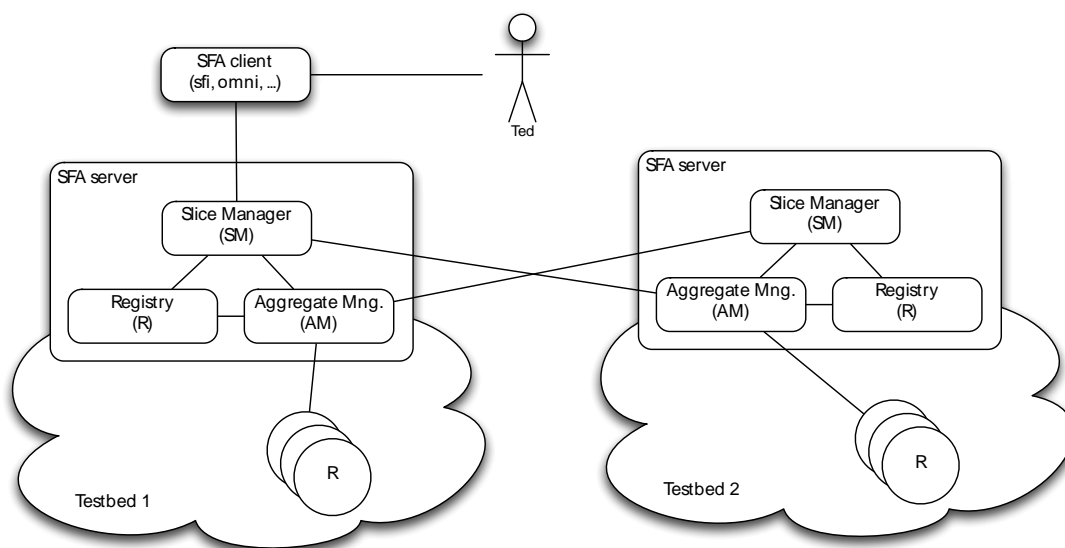


Figure 10: SFA Architecture

Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden.Figure 10: SFA Architecture **Fehler! Verweisquelle konnte nicht gefunden werden.**shows the normally implemented architecture for SFA. As pictured, the SFA server e.g. aggregates information about testbed's resources and offers these to experimenters.

SFA uses Resource Specification (RSpec), based on a predefined XML schema, to describe resources. The XML schema is designed to be extensible, in order to support resources of various kinds.

Besides describing resources provided by the testbed, these RSpecs are also used to request specific resources by the experimenter. Thereby an RSpec may be fully bound, requesting a specific, named resource; or unbound, where for example the Aggregate Manager (AM) chooses a fitting resource on behalf of the user

3.2 FITeagle

FITeagle is a semantic- and microservices-based resource management toolkit for federated infrastructures, such as testbeds. On the north bound FITeagle provides a set of well-defined interfaces to cover the whole experimentation lifecycle. This includes native Representational State Transfer (REST) based APIs as well as FRCP and OML



Resources, on the other hand, are interfaced by the southbound interfaces. A resource in this context can be a physical or virtual resource.

An adapter is responsible for describing, provisioning, controlling and monitoring a single or multiple resources and their instances by publishing, receiving and subscribing to semantically annotated information.

As one of the main aspects of FITeagle is its resource centeredness, the implementation of different protocols (delivery mechanisms) is achieved transparently to the underlying resources. The core functionalities, such as a resource repository, reservation management, orchestration, elasticity or authorization decisions are located in the westbound area. Requirements for these modules are derived by identifying common functionalities needed in the northbound interfaces. Finally, the integration of existing services is located at the eastbound area. This includes services such as an OML Measurement Stream protocol service enabling resource and experiment monitoring.

In order to allow external experimenters to utilize the 5G capabilities of the SoftFIRE testbeds, an adapter to the testbeds management system FITeagle was written. This adapter translates between the SFA interface of FITeagle and the HTTP interface of the Open Baton.

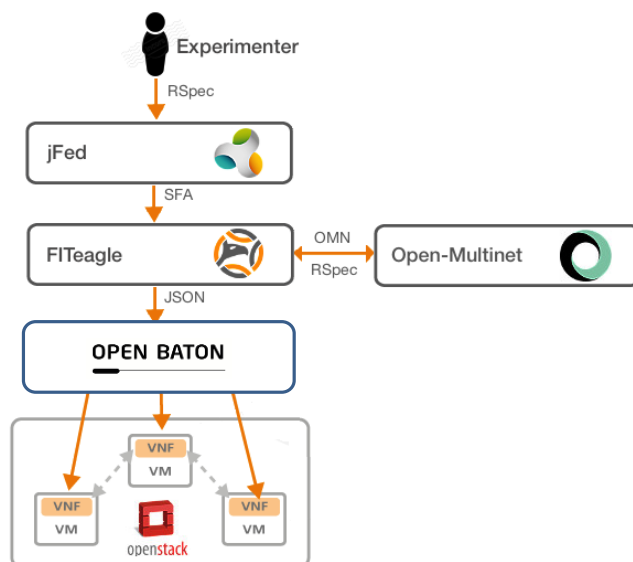


Figure 11: 5G Orchestration Architecture

Figure 11: 5G Orchestration Architecture shows that the user interacts with an SFA tool, in this case the jFed (jFED) probe GUI, by inputting a GENI RSpec, which is sent to the SFA API of a running instance of FITeagle. FITeagle in turn translates the RSpec into OMN and then communicates with Open Baton via its JSON API. Open Baton then provisions instances of the requested VNFs within a virtual machine in the respective OpenStack data center.

3.3 Open Baton

As shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**, FITeagle platform interfaces with the Open Baton (OpenBaton) platform. Open Baton is the first open-source ETSI NFV standard-compliant platform for the virtualization of network functions. It addresses both network-operators as well as cloud computing service providers and is suitable for the



virtualization of 5G and critical networks, as well as M2M and multimedia platforms. Open Baton enables virtual network services deployments on top of cloud-infrastructure and thereby builds a bridge between cloud computing service providers that have to understand network functions and network function providers, which require the appropriate infrastructure support for the virtualization. Open Baton is enabling in particular dynamic deployments of core network environments services.

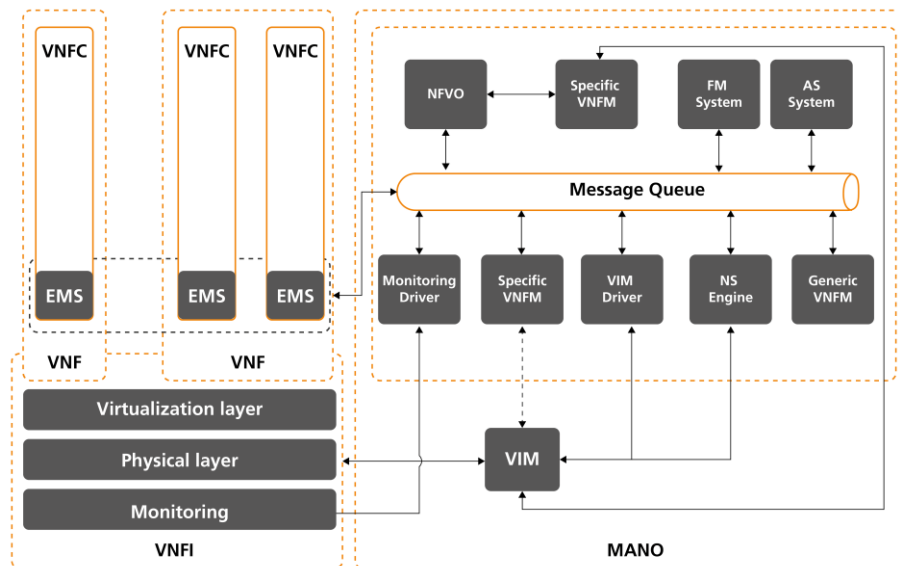


Figure 12: Open Baton platform

Open Baton is based on the ETSI specification NFV Mano v1.1.1 specification(ETSI, 2014), which was published at the end of 2014. The platform can be easily installed on existing cloud-infrastructures like OpenStack and consists of (see Figure 12: Open Baton platform):

- NFV Orchestrator (NFVO) that dynamically orchestrates carrier-grade network functions and services as well as infrastructure resources
- A generic Virtual Network Function Manager (VNFM) that dynamically manages virtual network the functions
- A set of libraries (SDK) for the creation of customized VNFMs and for interfacing with the northbound ReST APIs
- A user-friendly dashboard through which the platform can be administrated.
- An external module dedicated to execute Fault Management specific actions on deployed Network Services
- An external module dedicated to perform Auto-Scaling on deployed Virtual Network Functions
- An extendible set of plugins that enable the communication with the Virtualized Infrastructure Manager and with the monitoring system

3.4 jFed Experimenter Client



As SFA is standardized, experimenters can use any compliant client of their choice to interact with the testbed. The most convenient client as tested with the FITeagle implementation of the SFA interface is developed by iMinds and is called jFed Experimenter Client.

With jFed experimenters can allocate and provision resources of various kinds. The usage is described in detail in Section 4.

3.5 OMF 6 – Federated Resource Control Protocol

OMF (OMF) is a control, measurement and management framework for experimental platforms originally developed by Rutgers University and NICTA.

OMF provides a set of tools to describe and instrument an experiment, execute it and collect its results.

To do this, the user describes his experiment in a high-level domain-specific language (OEDL), and passes it on to OMF. The framework will in turn deploy and configure the experiment on the testbed(s) according to the user's description. Then it will initiate and control the execution of this experiment. Finally, during the experiment execution, the framework will measure and collect data according to the user's description. These measurements are sent to a repository available to the user and can also be used to dynamically steer the experiment control.

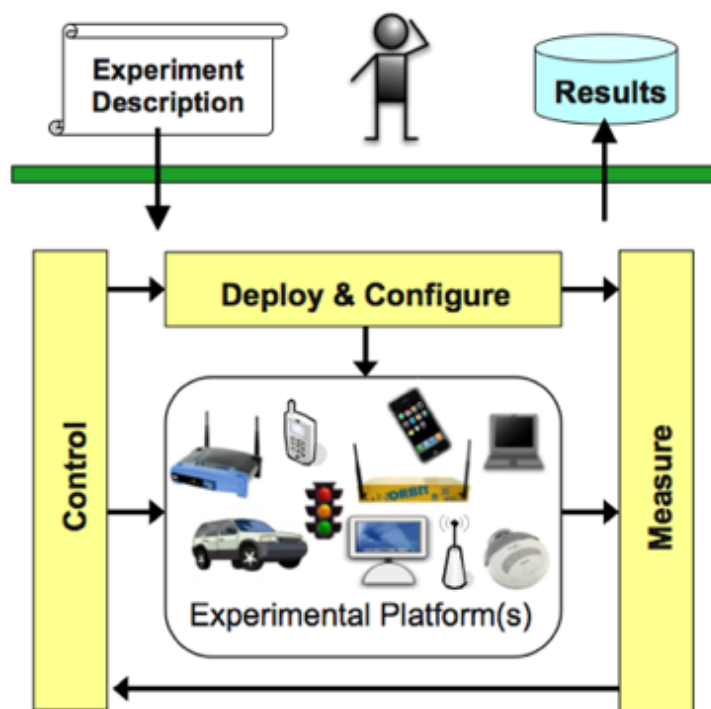


Figure 13: OMF

OMF 6 implements the Federated Resource Control Protocol (FRCP), a publish-subscribe (PubSub) based message protocol on top of either XMPP or AMQP (Figure 13: OMF).

For each resource to be controlled a proxy client resource controller (RC) registers at the PubSub-Server. When the experiments start, the experimenter sends his requests to this server, which forwards to the RC (Figure 14: FRCP Sequence Diagram).

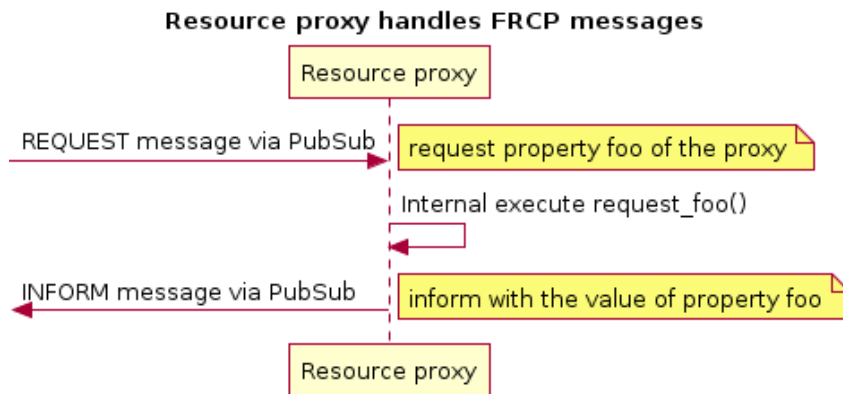


Figure 14: FRCP Sequence Diagram

3.6 OML

OML is an instrumentation tool that allows application writers to define customizable measurement points (MP) inside new or pre-existing applications. Experimenters running the applications can then direct the measurement streams (MS) from these MPs to remote collection points, for storage in measurement databases.

(<http://omf.mytestbed.net/projects/oml/wiki>)

OML can be used to collect data from any source, such as statistics about network traffic flows, CPU and memory usage.

It is a generic framework that can be adapted to many different uses and consists of two main components:

- *OML client library*: the OML client library provides a C API for applications to collect measurements that they produce. The library includes a dynamically configurable filtering mechanism that can perform some processing on each measurement stream before it is forwarded to the *OML Server*.
- *OML Server*: the OML server component is responsible for collecting and storing measurements inside a database.

A couple of OML-instrumented applications that perform measurements and filter and collect them using OML, including an OML-capable version of Iperf can be found at (http://omf.mytestbed.net/projects/omlapp/wiki/OML-instrumented_applications).

4 The experimenter's Life Cycle

4.1 Experiment Lifecycle

The life cycle of an experiment is depicted in Figure 15: Experiment Lifecycle. The actual conduction usually consists of the following steps the experimenter takes:

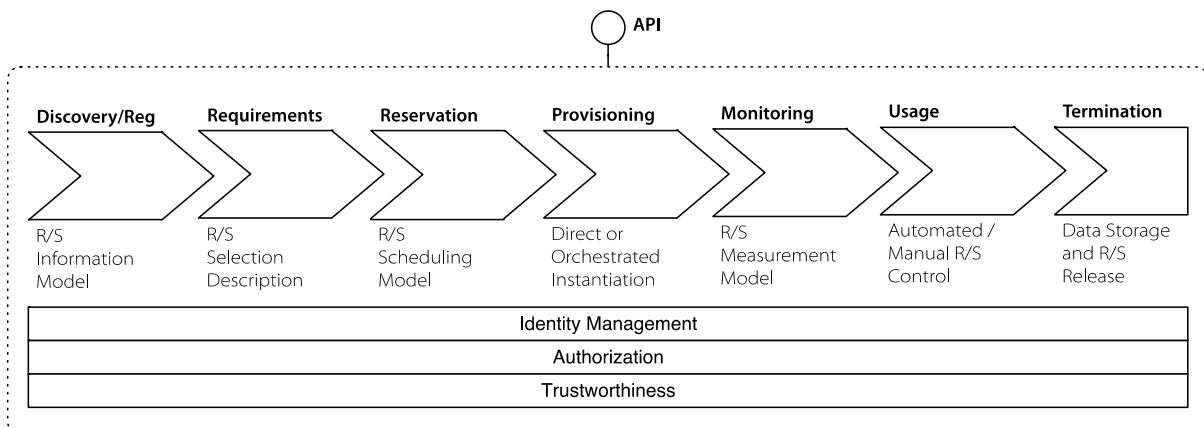
- Discovery:



- List the resources that are available on the testbed/federation.
- Requirements:
 - Select and formally specify the resources the experimenter aims to use for performing the experiment.
- Reservation:
 - Reserve a timeslot to access the resources; in the case of LTE and other radio resources an exclusive access is of importance to avoid interferences.
- Provision:
 - In this step the resources are provisioned for the experimenter. E.g. this can be the inclusion of the experimenters ssh key to provide such access or start-up of core network services with configuration parameters specified by the experimenter.
- Monitoring:
 - The setup of appropriate monitoring tools, so that during the experiments' execution the resources are being monitored for subsequent evaluation.
- Usage:
 - The actual experiment is run in this phase
- Termination:
 - After the experiment is finalized, the resources have to be released and restored.

All of these experiment life cycle steps are support by mechanisms for identity management, authorization and authentication.

The SoftFIRE approach allows in addition to these steps the upload of custom Network function images into the Testbeds. The upload is done before the discovery step by using the SoftFIRE Software portal. The NFV image is then automatically distributed into all connected Testbeds where they can be discovered via the SFA API.

**Figure 15: Experiment Lifecycle**

The use cases described in following paragraphs all follow this experiment lifecycle

4.2 Design and Programming

The first entry point for experimenters to access the resources offered by SoftFIRE is the SoftFIRE portal. This portal aggregates information and services useful and needed by the experimenters.

A set of tutorials provides knowledge needed on how to book and use resources.

As written above, the methods provided by the testbed to allocate and provision services are aggregated within the SFA interface. SFA uses XML RSpecs to describe either requests, manifests or advertisement of resources. The portal offers example RSpecs to provision e.g. an Open5GCore instance.

If the user wants to add own implementations of Virtual Network Functions the portal offers suitable interfaces and tutorials on how to create and build the needed service descriptors and packages.

Finally, experimenters can download X509 certificates in order to authenticate against the testbed.

Preparation phase

- In any case, upload the image to the **SoftFIRE software portal**
- Upload package to FIT/OB through **SoftFIRE software portal**

As described in the D2.1, the SoftFIRE Software Portal (SSP) is in charge of dealing with the images and the VNF packages.

The Experimenter will interface with the SSP in order to upload the already prepared images and VNF packages. Hence in a second moment, these VNFs will be available to be deployed into the federated testbed. An image has to contain the VNF required software, even precompiled, in order to allow the VNF to be instantiated in a short timeframe and without any dependencies besides the other deployed VNFs.

The name or id of the uploaded image has to be then specified into the VNF package. The SSP is in charge of deploying the package directly to the NFVO. The correct uploaded images have



to be specified thus the VNF that will be available afterwards will point to that image containing the correct software.

4.3 Access to the Federated test bed

To access the federated testbed users have to use a suitable SFA client, e.g. jFed. The following sections show how jFed is used.

After receiving a certificate via the SoftFIRE portal, the experimenters can start their experiment.



Figure 16: jFed Login Screen

Figure 16: jFed Login Screen shows the login screen of the jFed Experimenter GUI.

4.4 Allocation of Resources

After the experimenter has logged in to his jFed client a new experiment can be created by clicking the "New"-button (Figure 17: jFed - New Experiment).

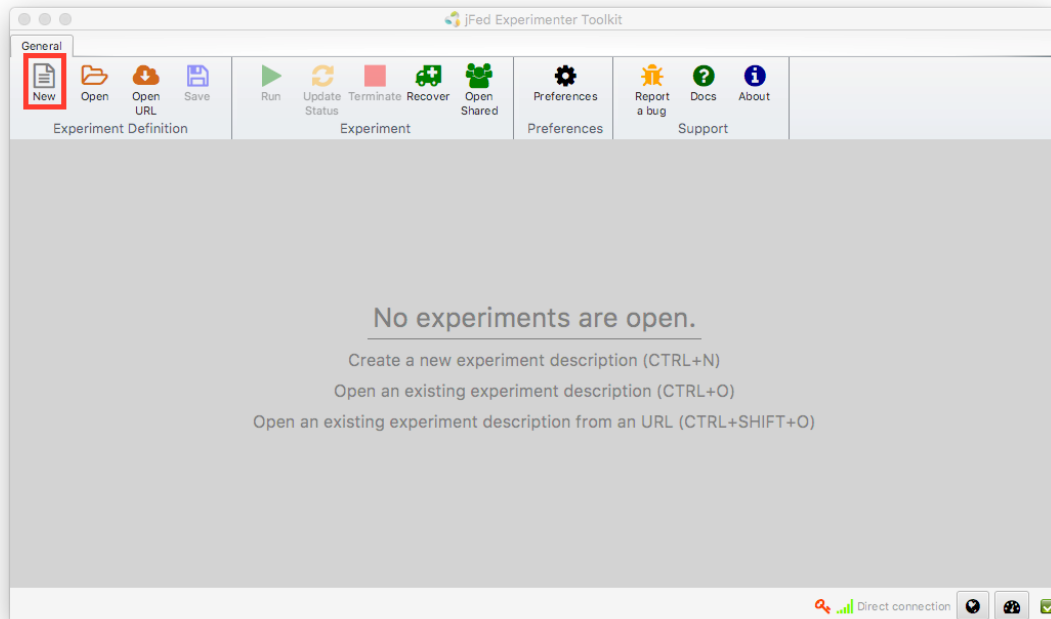


Figure 17: jFed - New Experiment

Afterwards the user can start to select resources from the federated testbed. The drag-and-drop functionality of jFed is not yet fully supported and since the VNF's might need additional configuration parameters, thereby utilizing the extensibility of RSpecs.

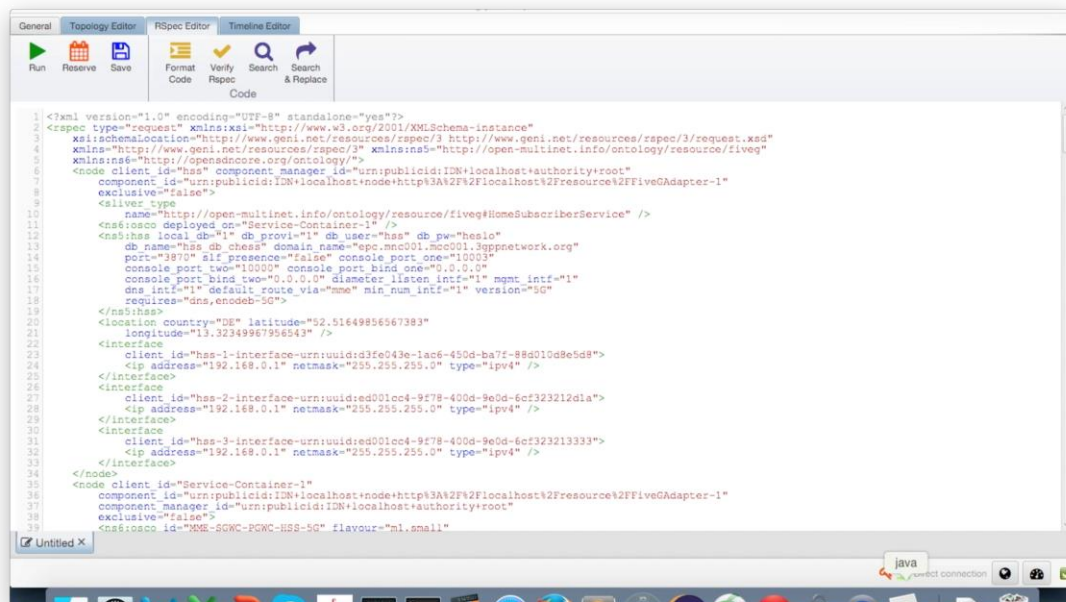


Figure 18: 5G RSpec

Figure 18: 5G RSpec shows an excerpt of an RSpec used to provision Open5GCore. A complete example will be provided at the SoftFIRE portal.



When clicking the “Run”-button in the top left corner, the topology is provisioned. This means, a request is sent to FITeagle, where it is handled and forwarded to appropriate interfaces of Open Baton.

Open Baton then takes care of VNF orchestration.

If the startup and configuration of services was successful is indicated by green colouring of the nodes in the “Topology” tab (Figure 19: jFed 5G Topology).

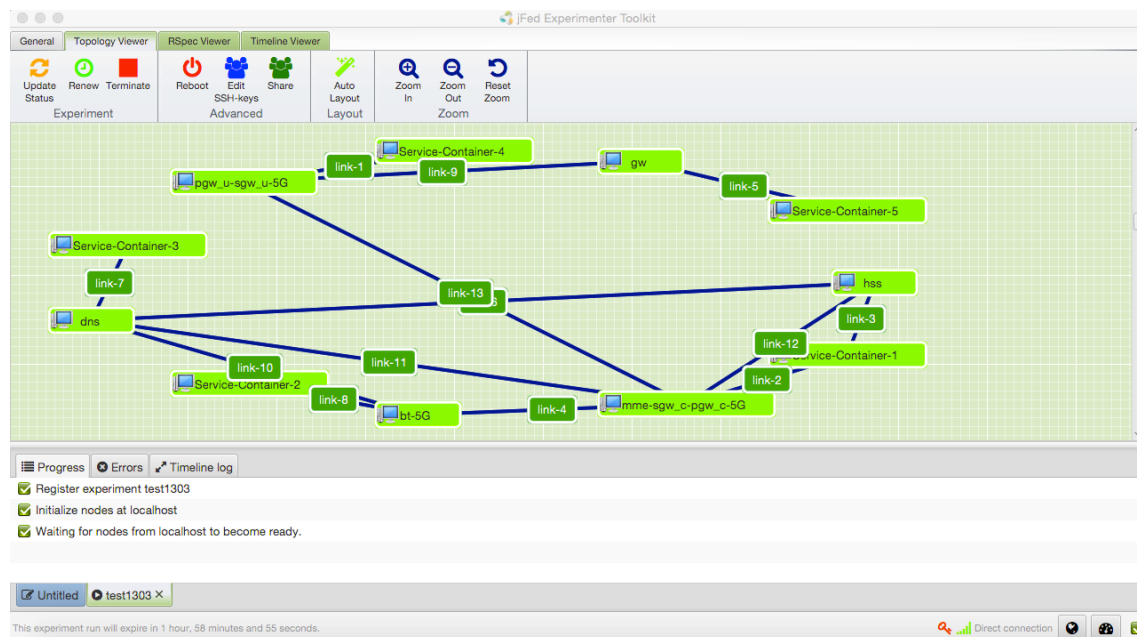


Figure 19: jFed 5G Topology

Provisioning of resources (deployment phase)

- From jFed user can build his own NSD with the VNFD uploaded through the **SoftFIRE software portal**
- Run NSD → means also create NSR
- Check the logs of deployment

4.5 Running an experiment

When the allocated resources have been successfully provisioned, the manifest RSpec returned by FITeagle contain the names of the created nodes.

With these names the experimenter can build the OEDL script, which executes the experiment.



```
defProperty("a_resource", "replace-with-your-resource-name", "ID of a
resource")

defApplication('ping') do |app|
  app.description = 'Simple Definition for the ping-oml2 application'
  app.binary_path = '/usr/bin/ping-oml2'
  app.defProperty('target', 'Address to ping', '', {:type => :string})
  app.defProperty('count', 'Number of times to ping', '-c', {:type =>
:integer})
end

defGroup('My_Pinger', property.a_resource) do |g|
  g.addApplication("ping") do |app|
    app.setProperty('target', 'mytestbed.net')
    app.setProperty('count', 3)
  end
end

onEvent(:ALL_UP_AND_INSTALLED) do |event|
  allGroups.startApplications
  after 5.seconds do
    Experiment.leave_memberships
    Experiment.done
  end
end
```

?

Listing 1: OEDL Example

Listing 1 contains an example OEDL script, which executes a “ping” command.



The experimenter feeds this script to the experimenter client.

4.6 Troubleshooting during the experiment

Once the deployment of the requested resources is completed, it is possible to check the log of the instantiation, configuration and start scripts of each Virtual Network Function. The logs will be shown through the jFed tool. The experimenter can in this way check if any error occurs during the instantiation of his experiment.

Anyway the overview of the deployment logs can be not enough to actually find the issue that may happen. Moreover, the logs refer only to the instantiation process and not to the run time execution of the VNF.

For that reason, it will be given the possibility to the experimenter to access his own VMs through SSH. The access will be controlled by the usage of certificates and is only possible within the SoftFIRE VPN. This is a necessary feature to be given to the experimenter not only to execute troubleshooting of the running experiments but also in order to achieve a full control of the experiments.

Access to the private network of the federated testbed is possible via an OpenVPN server that is provided by TUB. To access this OpenVPN the experimenter has to use the same certificate that is used to authenticate via the jFed SFA client. By using the VPN the experimenter gains direct access to his virtual network functions.

In case of any issues related to the SoftFIRE infrastructure, please see Section 5 and Section 6.

4.7 Security and Monitoring Tips

4.7.1. Monitoring

The SoftFIRE project has identified a number of KPIs that are platform related and cover several features of it. They are intended for use of the Platform Provider. The SoftFIRE project deems that monitoring policies of specific applications has to be considered an activity internal to the applications itself and it will be up to the programmers to choose tools and mechanisms to implement the proper monitoring policies.

However, SoftFIRE platform KPIs will be implemented in a progressive manner so that the measurements and evaluation of the platform can benefit from them. In order to collect the measures, some tools are used like Zabbix and functionalities offered by basic platform components like OpenStack. The Application programmer familiar with these tools can use them in order to monitor his experiment. During the project lifetime, KPIs and measures will be made available and the programmers could take advantage of these measures.

For further details, the experimenter can refer directly to the official Zabbix documentation at <http://www.zabbix.com/documentation.php>.

4.7.2. Security

The current state of the infrastructure does not provide a security monitoring system. It will be added in a future version. Further information on the Security Policy Management and the future design of a monitoring system can be found Del. 2.2 Security Policy Management, which is available for authorised users.



This handbook will be extended with a description of the different features offered to the experimenter, how to enable and configure the optional ones and how to interact with them.

In particular we are planning to develop:

- An intrusion monitoring system for the layered infrastructure (this service cannot be disabled by the experimenter).
- A service-specific security mechanism.

This feature will provide to the experimenter the ability to include in the network services additional components related to the security monitoring.

Here we will describe how the experimenter can enable and configure security add-ons.

- A ticketing mechanism for the management of security issues.

As soon as these valuable functionalities and mechanisms will be available, this document will explain how the experimenter can be aware of a security issue and how he can deal with its management by means of these new functions.

4.8 Closing the experiment and feedbacks

After the user has commenced the experiment and has gathered all logs and results, termination of the experiment, which means release of resources, is the final step.

Via jFed this is done by clicking the “Terminate”-button (Figure 20: jFed Terminate Topology).

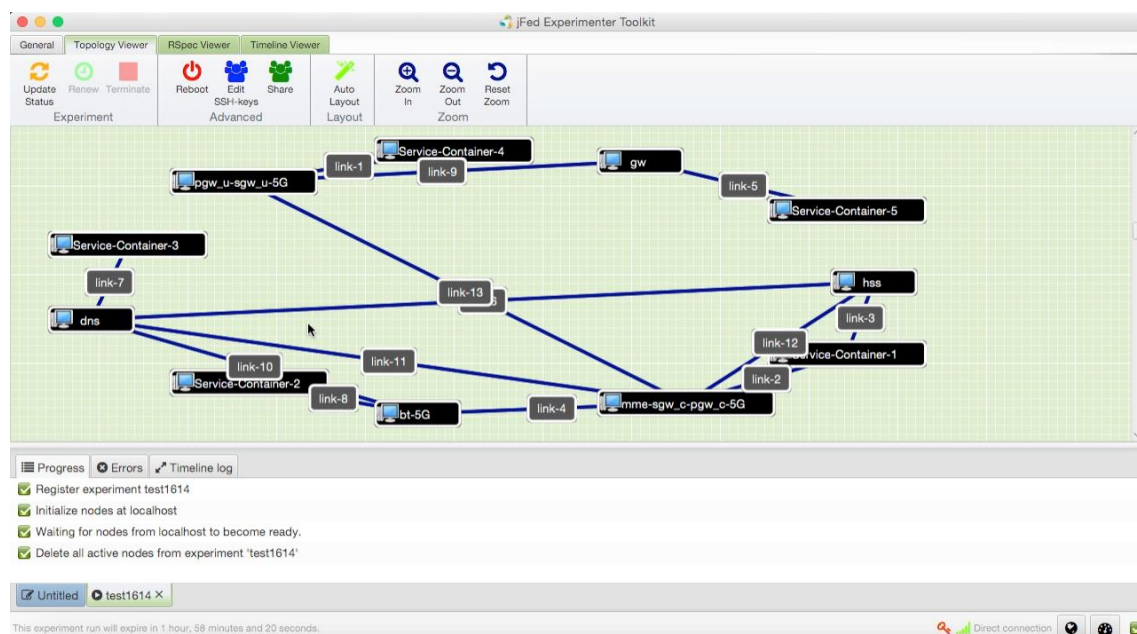


Figure 20: jFed Terminate Topology



5 SoftFIRE support to experimenters

The experimenter can require support related to the usage of the federated testbeds by means of a First Level Support systems (e.g. Trouble Ticket System), implemented with [REDMINE](https://redmine.softfire.eu/) Tool accessible at the following address <https://redmine.softfire.eu/>

Before opening a new case it is recommended to verify in REDMINE if similar cases have been recently issued (<https://redmine.softfire.eu/projects/softfire/issues>).

5.1 Subscription

The selected experimenter is configured in Redmine tool portal with proper credentials in order to run case submissions. The Redmine portal will provide experimenter with credential via mail notification at the start of the open call. The subscription will last till 15 days after the closure of open call.

5.2 Case submission and followup

Once entered in Redmine (Figure 21: Redmine home for SoftFIRE) with the assigned Login/Password jump to the SoftFIRE Project:



Figure 21: Redmine home for SoftFIRE

Here you can check the current issues or open a new one (Figure 22: Issues tracking) by means of the following Tabs:

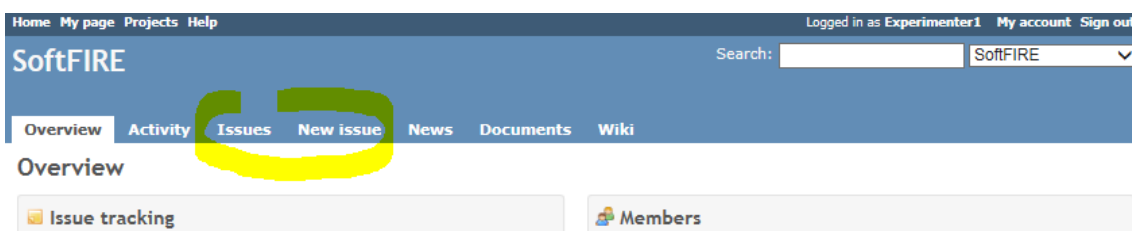



Figure 22: Issues tracking



To open a new case, from “New Issue” tab (Figure 23: Describing a New Issue) you access to the page in the picture below where you have to:

1. Select the most suitable Tracker typology
2. Assign a short and significant slogan in the Subject field
3. Write an accurate description
4. Set a priority based on its Urgency (Normal as default), that will be then revised by the receiver
5. If any file can better support the case, attach it in the Files field, clicking on “Browse...”.

If you want to add a snapshot in the text description you have to:

- a. attach the related .JPG file by clicking on “Browse...”
- b. click on the “Image” icon  and “!...!” will be added in the description
- c. Copy the “File Name.JPG” of the picture between “!...!”

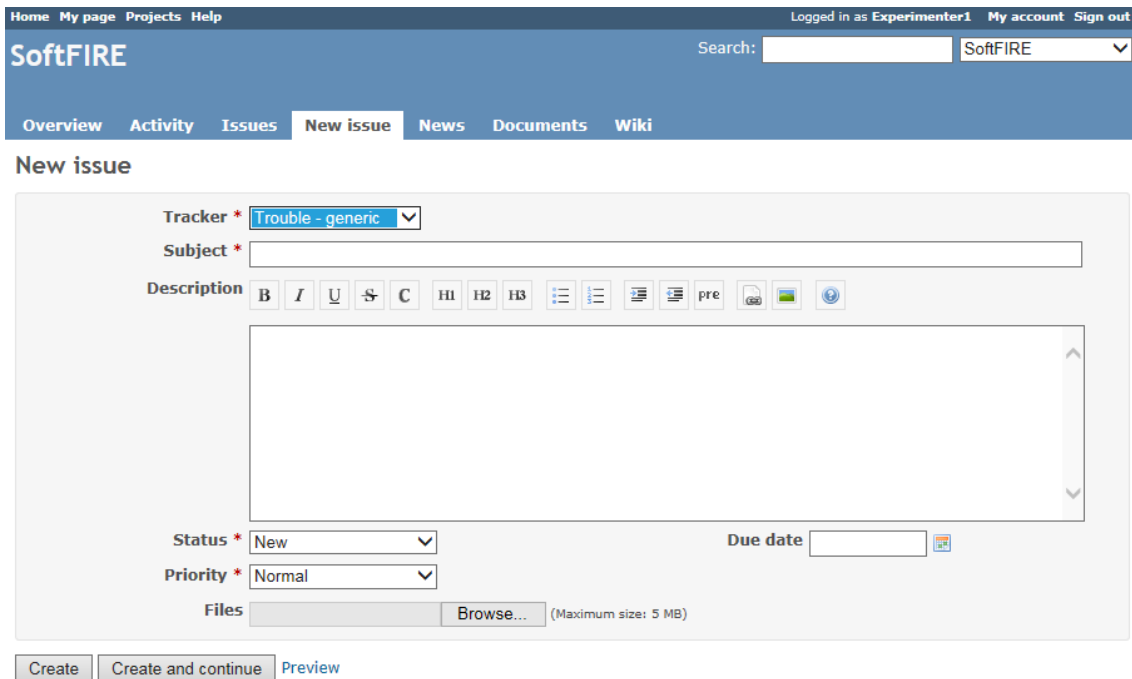


Figure 23: Describing a New Issue

Check if everything is correctly filled by clicking on “Preview” and finalize it by clicking on “Create”.

If you need to update your own issue, go under Issue Tab click on the Item and then on  **Edit**

5.3 Workflow

The just opened Cases will be immediately notified to a reference person for each Infrastructure: they will analyze it and, based on its description, will assign it to the most suitable responsible for follow up (stepping the item in “In Progress” status, asking for Feedback if something unclear or moving to resolved if a solution has been identified meanwhile).



The Case will follow the workflow described in Figure 24: Process for tracking issues (*only the Infrastructure Owners can directly Reject or Close the issue*):

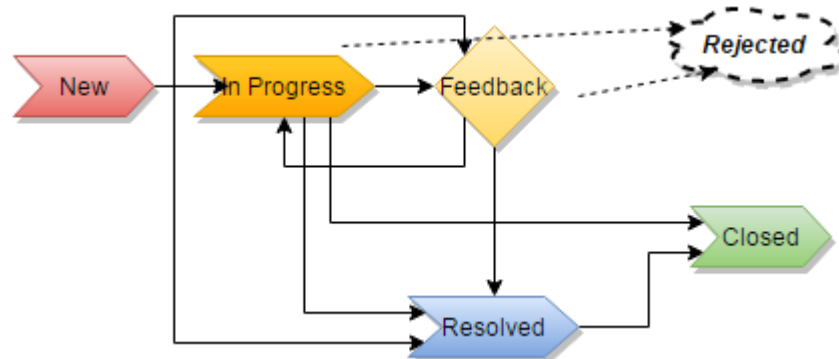


Figure 24: Process for tracking issues

5.4 Service Standards

The Platform will be operational during these time frames:

Working hours: 10:00 a.m. to 5:00 p.m. CEST

Monday to Friday GMT

Outside this timeframe: issues/requests will be emailed and managed at best effort

Please check your local time correspondence (<http://www.worldtimebuddy.com/>)



6 SoftFIRE General Use

SoftFIRE can be used by experimenters that have applied to Open Calls as well as from organizations or entities that want to experiment on the platform outside of the process of the Open Calls. For the use of the testbed there are general rules that are described in the following sections and generally apply.

6.1 Constraints on the Experimenters Use

The SoftFIRE infrastructure is composed by loosely integrated platform under different administrative domains. In addition the different platform are ran for experimental purposes and they are not yet considered a mass production tool. This means that bugs and issues in the platform behaviour can occur and will occur. Actually the scope of the experiments is also to support the tune up and the assessment of the platform as a whole.

The Platform is still under development and it has a basic set of functionalities that have to be tuned up and it is missing a number of features that will be progressively added in the future. SoftFIRE is by no means to be considered a product and so the usual support for software development cannot be granted.

Even if SoftFIRE aims at programmers, not all the features to allow for a fast programming approach are provided. This is due to differences in the component testbeds and to security controls imposed by different administrative domains. The programming phases could results cumbersome and not particularly attractive, however they may improve along the lifetime of the project.

Service level agreements (SLA) do not apply during the experimentation phases. Because this is a period to test and explore SoftFIRE, the experimenters should not run production applications on the infrastructure Platform during the trial.

The SoftFIRE project reserves the right to discontinue at any time the service if the use is not consistent with the purpose of SDN/NFV and/or violate any aspects of infrastructure security or shall conflicts any ongoing experimentation.

During the running period of the experiments, the SoftFIRE project will put in place a team that will support the experiments in their work on the platform. As said this is not offered as a professional service and its working will be on the base of best effort. The entire infrastructure should be considered more a sort of α -test platform. Possible downs could occur without notice or due to overload caused by parallel experiments.

SoftFIRE will offer expertise available by email (with possible follow-up by phone) and two hours per day during the experimentation phase in order to collect issues and provide responses. We'll try to provide most of the answers by 24 hours (typically next morning or afternoon). Some issues could be not solvable due to the short time of the experiment period or due to the need to intervene on the platform. The supporting time will work with experimenters for circumvent the problems.

The project will also issue limits and constraints on the allocation of available resources. This is due to the need to support and allow parallel experimentations. This limitation depends on the total capabilities of the federated platform as well as the number of experimenters and their requests in terms of resources. Typical limitation could be related to the max number of VMs to be instanced, the number of physical resources usable or the max memory usable per



experimenters. Other limitations could apply, or be notified during the course of the experimentations.

Paid support: If you have extensive support needs during the experimentation, you may purchase paid support for the SoftFIRE Platform. The paid support package offers increasing levels of hands-on support and response time.

A few aids for the experimenters could be added in due time in the SoftFIRE portal at <http://www.softfire.eu>

- An “on line” tutorial on how to access and use the platform will be prepared before the experimentation phase
- A presentation with a use case
- Other educational material

6.2 Additional support

For more personalized services experimenters are referred to the [Handbook: Guidelines and Rules for ondemand access to the SoftFIRE Testbed](#) as well as to the contact persons on the SoftFIRE website.

Update references will be posted in the Contact section of the SoftFIRE web site <https://www.softfire.eu/contact-support/>.

The feedback from experiments will be of great use in order to assess the maturity of the solutions and their applicability and potential to support business related implementations, so the project invites all the experimenters and users in being very proactive in providing this type of information.



7 References

- ETSI. (2014). *Network Functions Virtualisation (NFV)*. ETSI.
- FED4Fire. (n.d.). *Federation for Future Internet Research and Experimentation*. Retrieved from <http://www.fed4fire.eu/>.
- FIRE. (n.d.). *Future Internet Research and Experimentation*. Retrieved from <https://www.ict-fire.eu/>.
- FITEagle. (n.d.). *A Semantic Resource Management Framework*. Retrieved from <http://fiteagle.github.io/>.
- jFED, A. j.-b. (n.d.). *jFed*. Retrieved from <http://jfed.iminds.be/>.
- OMF. (n.d.). *Open Management Framework*. Retrieved from http://omf.mytestbed.net/projects/omf/wiki/OMF_Main_Page.
- ON.Lab. (n.d.). *Bringing openness and innovation to the Internet and Cloud*. Retrieved from <http://onlab.us/>.
- ONOS. (n.d.). *Open Network Operating System*. Retrieved from <http://onosproject.org/>.
- OpenBaton. (n.d.). *An open source Network Function Virtualisation Orchestrator (NFVO)*. Retrieved from <http://openbaton.github.io/>.
- Opendaylight. (n.d.). *Open Source SDN Platform*. Retrieved from <https://www.opendaylight.org/>.
- OpenStack. (n.d.). *Open source software for creating private and public clouds*. Retrieved from <https://www.openstack.org/>.
- Zabbix. (n.d.). *The Ultimate Enterprise-class Monitoring Platform*. Retrieved from <http://www.zabbix.com/>.



8 List of Acronyms and Abbreviations

Acronym	Meaning
CA	LTE-A Carrier Aggregation
CC	UoS 5G Cluster Member
CM	UoS 5G Cluster Controller
EMM	EPS Mobility Management
EPC	Evolved Packet Core (LTE-A)
GUI	Graphical User Interface
HSS	Home Subscriber Server
IaaS	Infrastructure as a Service
LTE-A	Advanced Long Term Evolution
MME	Mobility Management Entity
NF	Network Function
NS	Network Service
PDN	Packet Data Network
PGW	PDN Gateway
PoC	Point of Contact
PPE	UoE CUPS evolved combined Packet Processing Entity including SGWu and PGWu NF entities
RAN	Radio Access Network
SFA	Slice-based Federation Architecture
SGW	Serving Gateway
UoS	University of Surrey
vNF	Virtual Network Function