SOFTFIRE

Software Defined Networks and Network Function Virtualization Testbed within FIRE+

Grant Agreement Nº 687860

Guidelines, rules and mechanisms governing the usage of the SoftFIRE Testbed for the Third Open Call

Experiment Management

Version:	3.0
Due Date:	July 21st 2017
Delivery Date:	July 21st 2017
Type:	Report ®
Dissemination Level:	PU – Public
Lead partner:	
Authors:	All Partners (See list below)
Internal reviewers:	PMC

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
0.0	Tuesday, 11 April 2017	Roberto Minerva	EIT Digital	Modification to D3.3 Handbook for experimenters about the Programming LifeCycle
1.0	21st April 2017	Roberto Minerva	EIT Digital	Added newer description of testbeds. Added a table of full capabilities
1.1	11 May 2017	Lorenzo Tomasini	TUB	Improvement of Experiment lifecycle documentation
1.2	11 May 2017	Bjoern Riemer	FOKUS	Definition of SDN Resources and improvement of Testbed descriptions
1.3	12 May 2017	Lorenzo Tomasini	TUB	Integration of Partners contributions
1.4	15 May 2017	Lorenzo Tomasini	TUB	Final version for internal review
2.0	15 May 2017	Roberto Minerva	EIT Digital	Final Editing
2.1	21st July 2017	Lorenzo Tomasini	TUB	Update of content
2.2	21st July 2017	Massimiliano Romano	Assembly Data System	Testbed List – SDN resources
3.0	21st July 2017	Roberto Minerva	EIT Digital	Revision and consolidation

Annexes:

Nº	File Name	Title

Contributors:

Contributor	**Partner**
Giuseppe Carella	TUB
Daniele Carmignani	Security Reply
Marius Corici	FOKUS
Peter Feil	Deutsche Telekom AG
Susanne Kuehrer	EIT Digital
Marco Miglione	Ericsson
Roberto Minerva	Telecom Italia
Fabio Paglianti	Security Reply
Björn Riemer	FOKUS
Massimiliano Romano	Assembly Data System
Umberto Stravato	Ericsson
Lorenzo Tomasini	TUB
Serdar Vural	University of Surrey

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

Deliverable Topic Handbook for Third Open Call

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 is based on Deliverable D3.3 (Guidelines, rules and mechanisms governing the usage of the SoftFIRE Testbed) and it has been update to reflect the current status of the SoftFIRE middleware. It presents an updated 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 a general description of the SoftFIRE Federated infrastructure is given. In particular, 2.2 provides a description of the different components testbeds that form the SoftFIRE Federated platform. Section 2.3 provides a view of the new tools that allow the SoftFIRE testbed to be accessible, manageable and integrated with the FIRE infrastructure. Section 3describes 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 security and monitoring of the experiment execution; and the closing and withdraw of the experiment. Section 4 presents some tools and mechanisms that can help the programmers during the experimentation phase and, finally section 5 describes some constraints and limitation in the usage of the SoftFIRE Federated testbed.

This document is an essential source of information for people that wan 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.

Contents

LIST OF FIGURES 8

LIST OF TABLES 9

1 INTRODUCTION 10

2 ARCHITECTURE OF THE FEDERATED TEST BED 12

2.1 THE FEDERATION OF TESTBEDS 12
2.2 THE COMPONENT TESTBEDS 14
2.2.1. ERICSSON LAB 14
2.2.2. TECHNICAL UNIVERSITY BERLIN SUPPORT RESOURCES 17
2.2.3. FRAUNHOFER FOKUS TESTBED 18
2.2.4. UNIVERSITY OF SURREY (UOS) 20
2.2.5. DEUTSCHE TELEKOM (DT) 21
2.2.6. ASSEMBLY DATA SYSTEM (ADS) 22
2.2.7. OTHER TESTBEDS 22
2.3 SOFTFIRE MIDDLEWARE ARCHITECTURE AND SUPPORTING SERVICES 23
2.3.1. NFV FEATURES 25
2.3.2. SDN FEATURES 26
2.3.3. MONITORING FEATURES 27
2.3.4. SECURITY FEATURES 28
2.4 ADDITIONAL RESOURCES AVAILABLE TO THE EXPERIMENTERS 29
2.4.1. VIRTUAL RESOURCES 29
2.4.2. PHYSICAL RESOURCES 32

3 HOW TO USE SOFTFIRE 34

3.1 GETTING ACCESS TO THE PLATFORM 35
3.1.1. GET OPENVPN CERTIFICATE 35
3.1.2. OPENVPN SETUP 35
3.1.3. REGISTER TO EXPERIMENT MANAGER 35
3.2 RUNNING THE EXPERIMENT 36
3.2.1. DESIGN PHASE 36
3.2.2. PROVISION PHASE 41
3.2.3. RUNTIME PHASE 41
3.2.4. CLOSING PHASE 42

4 SOFTFIRE SUPPORT TO EXPERIMENTERS 43

4.1 SUBSCRIPTION 43
4.2 CASE SUBMISSION AND FOLLOWUP 43
4.3 WORKFLOW 44
4.4 SERVICE STANDARDS 45
4.5 SUPPORT CONTACTS 45

5 SOFTFIRE GENERAL USE 46

5.1 CONSTRAINTS ON THE EXPERIMENTERS USE 46
5.2 ADDITIONAL SUPPORT 47

6 REFERENCES 48

7 LIST OF ACRONYMS AND ABBREVIATIONS 49

List of Figures

Figure 1: The SoftFIRE structure for programmers 13
Figure 2: Ericsson Computing View 16
Figure 3: Ericsson OpenStack Components Organization 16
Figure 4: Ericsson Testbed Networking Layout 17
Figure 5: TU-Berlin resources overview 18
Figure 6: FOKUS Testbed overview 19
Figure 7. Experimenter interaction 25
Figure 8: Experiment Lifecycle 34
Figure 9. SEM Experimenter page 36
Figure 10: Redmine home for SoftFIRE 43
Figure 11: Issues tracking 43
Figure 12: Describing a New Issue 44
Figure 13: Process for tracking issues 45

List of Tables

Table 1: Ericsson Hardware Structure of the TestBed 15
Table 2. Fokus OpenStack Hardware 20
Table 3: Network Equipment 21
Table 4: Network Services Provided by the UoS Testbed 30

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 larger ecosystem possible the creation of services as well as the functional extension of the platform itself.

SoftFIRE has three main objectives: supporting interoperability, programming and security of the federated testbed. Security and programmability of the platform are then major goals and they are the focus of the SoftFIRE's Third Open Call. However the experimenters are free to propose and work on any technically relevant topic related to NFV/SDN in the context of 5G evolutions.

The objective of this document is to illustrate the usage of the testbed to third parties and to describe the processes that are in place to monitor and govern the access to resources 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 facilitate 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 by means of a common middleware layer. These functions allow a uniform access and usage of the federated testbed. The project also tried to implement a first set of FIRE requirements to be fulfilled in order to allow a smooth and controlled access to the platform. The tools used in the First Open Call, FITeagle [1] and jFED ([2]), have been substituted with the SoftFIRE Experiment Manager (SEM) middleware in order to grant access to Open Baton [3] and to extend to other functionalities (such as SDN proxying). The SoftFIRE project decided to deprecate the usage of FITeagle and jFED and to move to a new and more industrial architecture based on TOSCA [4]. The work for creating a full architecture is still going on, but some useful functionalities are provided starting from the Second Open Call.

The programmers and experimenters should devote some time to familiarize with the SoftFIRE Software Architecture. 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 be progressively extended (introducing new capabilities) and will 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 be acquainted with 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 sketches also the approach used to support the life cycle of the experimentations on the federated testbed. This is clearly the most useful part of the document for programmers and coders and it is constantly upgraded and improved.

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. Updates to this document can be found in [5].

2 Architecture of the Federated test bed

2.1 The Federation of testbeds

SoftFIRE federates different European testbeds owned by the partners of the project. Currently the federated Testbed are:

- ***RMED Cloud Lab*** from Ericsson, located in Rome;
- **FUSECO Playground** from FOKUS Fraunhofer, located in Berlin;
- **5GIC** from University of Surrey, located in Guildford, Surrey;
- **ADS NFV Lab** from Assembly Data System, located in Rome

New testbeds are now in the integration phase and may soon join the Federation, e.g.,

- Deutsche Telekom;

In the past other Testbed were integrate (and currently not available):

- JoLNet from TIM, spread over several Italian cities;

In order to guarantee a secure communication over the public Internet, secured links (IPsec) are used in order to interconnect the testbeds data and control plane. Secure experimenter access is provided via a central OpenVPN service.

Experimenters can access the available resources through a single access-point, i.e., the *SoftFIRE Experiment Manager*, Figure 1: The SoftFIRE structure for programmers. This tool will be under development during the entire lifecycle of the project in order to progressively manage and orchestrate the allocation of several resources (Virtualization, SDN, 5G Resources, Security, and Monitoring). The Experiment Manager 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. The SoftFIRE Experiment Manager (SEM) will ensure interoperability with other technologies by implementing the standard TOSCA interface.

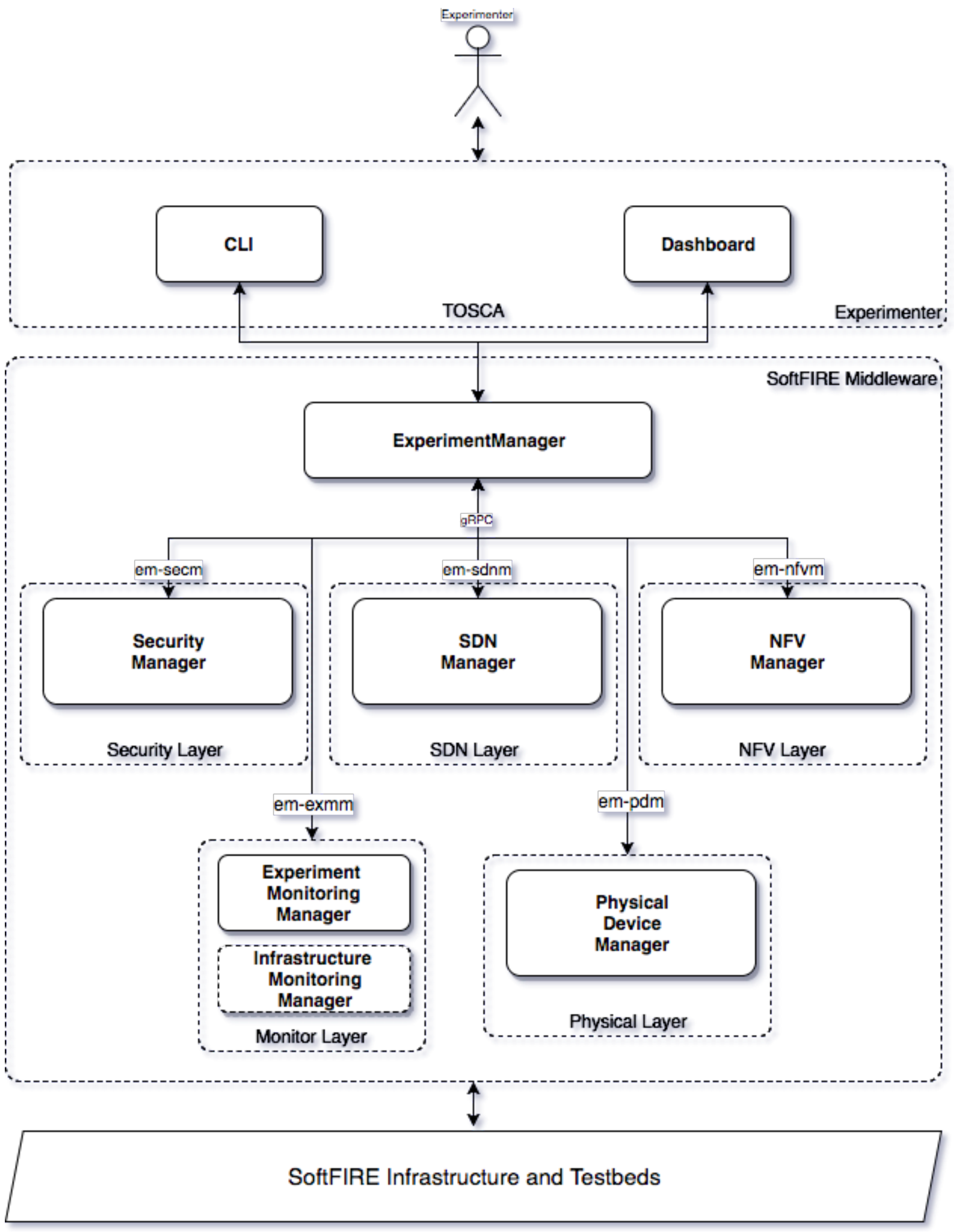


Figure 1: The SoftFIRE structure for programmers

SEM interacts with the *NFV Manager* that manage NFV computing and networking resources of the testbeds, which is an abstraction layer on top of an instance of *Open Baton*. In fact, Open Baton 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 [3]. In addition, it provides auto scaling and fault management based on monitoring information coming from the monitoring system available at the NFVI level.

Moreover, the project is working for integrating the SEM with an *SDN Manager* in charge of allocating SDN related resources to each individual experimenter. Mechanisms for allowing the allocation of SDN resources are currently under implementation and will be available in order to expose resources and APIs of OpenDayLight controller and/or closed source SDN controller.

Each testbed provisions virtual resources by means of an *OpenStack* [6] cloud controller (the adopted release is at least Newton) that controls the physical compute, storage, and networking resources dedicated to the project. This OpenStack controller is connected to the central Open Baton orchestrator, which coordinates the instantiation of virtual machines (VMs) over the testbeds. Exploiting the multiple point of presences supported out of the box by Open Baton, experimenters can choose specific locations meaning of the testbed 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., physical switches, wireless access points or other 5G related resources), SEM will be progressively extended in order to manage those resources, which are then exported towards the experimenters. In case of resources not fully integrated to the SEM mechanisms, experimenters will be able to access them either via direct access by means of SEM or direct APIs.

2.2 The component testbeds

The Federated testbed consists of multiple testbed sites, with the following total amount of virtualisation resources for experimenters. Please note the numbers noted in the table below indicate the total amount available for all experimenter VMs combined (not for each experimenter).

Testbed name	CPU (number)	RAM (in Gbytes)	Disk Storage (in Gbytes)
Fokus osdnc	32	125	12
Fokus dev	56	256	274
Ericsson	80	256	1600
DT	10	20	50
University of Surrey	12	48	240
Assembly Data System	32	96	1800
TOTAL	190	705	2176

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	2
16GB RDIMM, 1866MT/s	8
400GB, SSD SAS Value SLC 6Gbps	2
Intel X520 DP 10Gb DA/SFP+ Server Adapter	2
Intel Ethernet i350 QP 1Gb Network Daughter Card	1

The number of servers reserved for the project are three: one controller and two compute nodes Figure 2: Ericsson Computing View.

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.

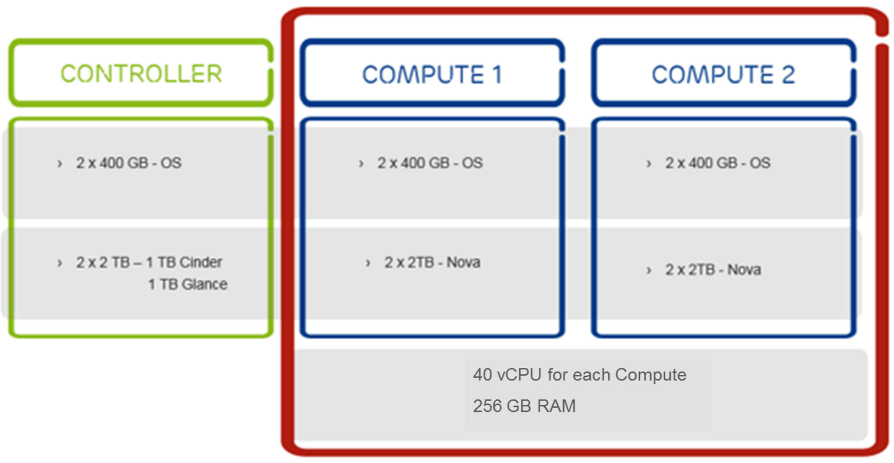


Figure 2: Ericsson Computing View

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

- **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.

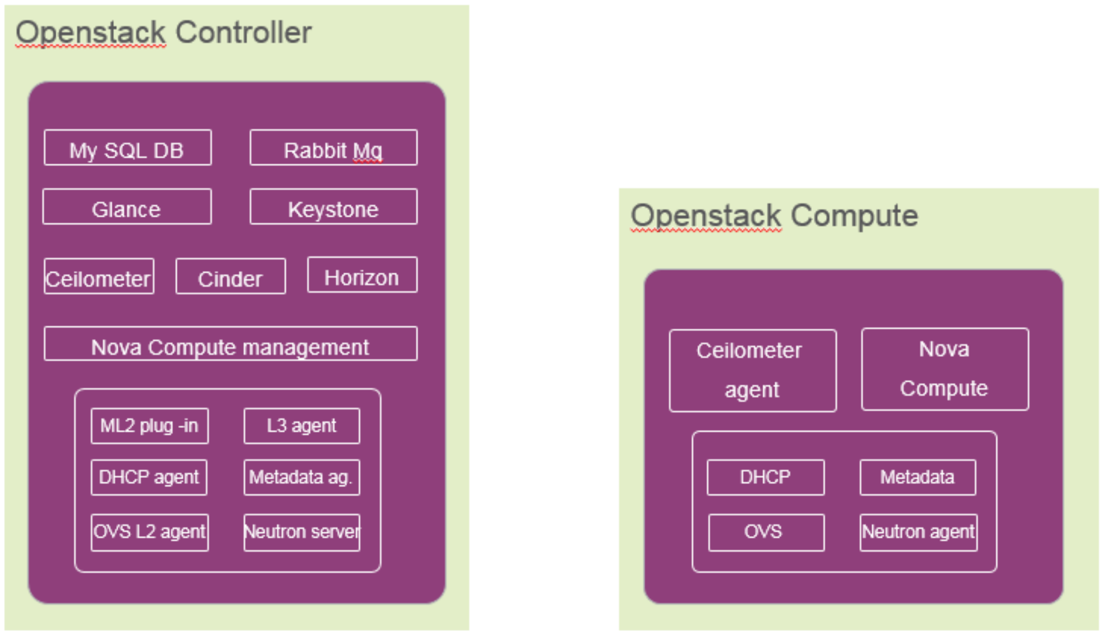


Figure 3: Ericsson OpenStack Components Organization

The following variants have been put in place:

- Instead of My SQL DB has been used PostGres;

- L3 agent is not installed cause of the presence of the external SDN controller;
- On the Compute Node, no DHCP and metadata are activated and OVS is loaded via the networking-odl pseudo agent

To extend OpenStack clouds capabilities with SDN functionality, OpenDaylight SDN controller (ODL Boron SR-2) under OpenStack Neutron has been integrated into the Neutron component of Openstack via the the OVSDB 2.7.0 south-bound plug-in.

OpenDaylight also provides the user with accessible API to control and refine the configuration of the attached OVS switches offering in this way basic SDN capabilities.

Network Design

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

- IDRAC network: is the console network;
- MGMT: Administration for operation and maintenance
- NB_API: Openstack Nortbound API;
- SB_API: Openstack southbound interface API (Openstack internal services)
- Tunnel network: compute interconnection for VMs traffic;
- Floating network: external network (flat) providing floating IPs to the VMs

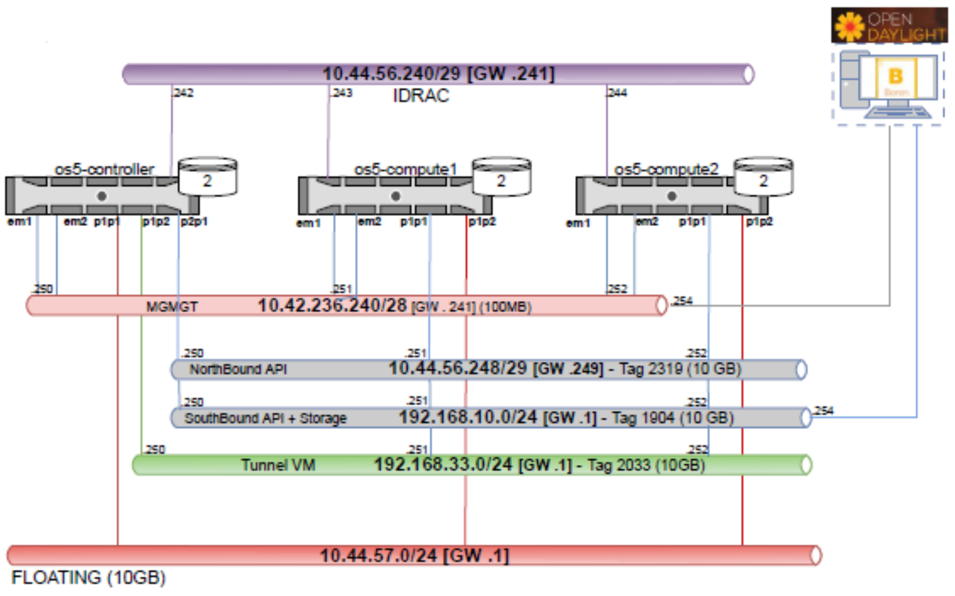


Figure 4: Ericsson Testbed Networking Layout

2.2.2. Technical University Berlin Support Resources

The TU-Berlin node is the center of the SoftFIRE's VPN network. It is realized as a pure virtual network that is feed as VLAN into the two Virtualization environments at TUB (see Figure 5: TU-Berlin resources overview). The VPN encryption is handled by a virtual instance of an OpenBSD based Firewall running on a VMware based virtualization cluster.

TUB also utilizes a private OpenStack cluster that is used to host generic Support Services like the development versions of the Orchestration tools used by SoftFIRE. This environment also hosts the Experiment manager plugin component that is used to store VM images because the bandwidth provided at TUB is much faster than on the other testbeds. The OpenStack cluster is not provided to the SoftFIRE experimenters to execute experiments.

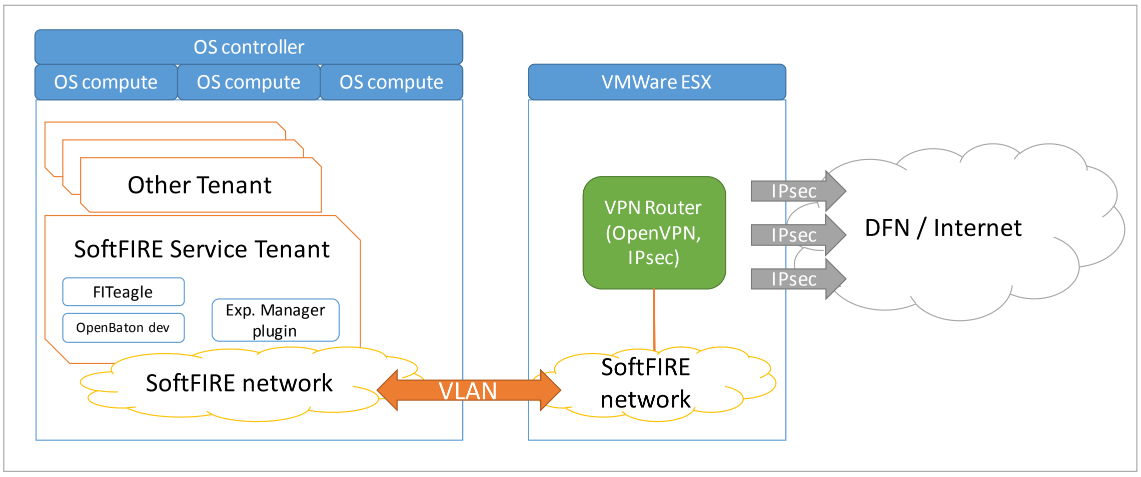


Figure 5: TU-Berlin resources overview

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 University. 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. The VPN Server also provides 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. The certificates are automatically generated by the SoftFIRE portal. This VPN server provides network access to registered experimenters so they can directly interact with their own NVFs.

2.2.3. Fraunhofer FOKUS Testbed

The SoftFIRE Testbed node located at Fraunhofer FOKUS in Berlin is realized as two separated OpenStack instances: One pure OpenStack installation and the other part as an integrated installation with OpenSDNcore to provide SDN features. Figure 6: FOKUS Testbed overview. The connectivity to the distributed parts of the SoftFIRE Testbeds is realized by an IPsec secured VPN link to the TUB.

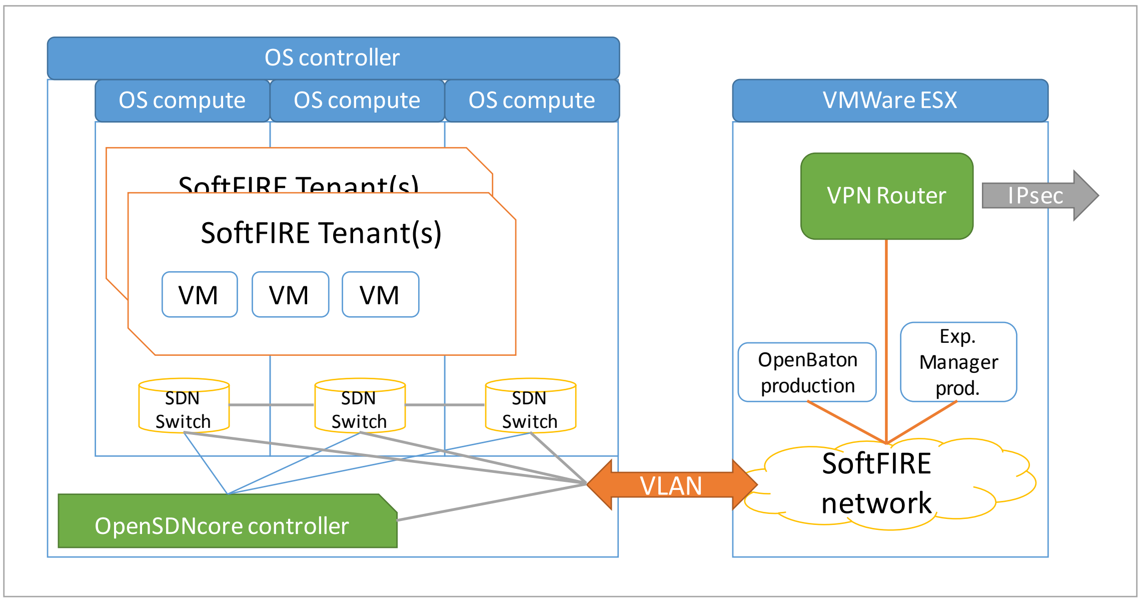


Figure 6: FOKUS Testbed overview

OpenStack Cluster

The compute resources are provided by two OpenStack Newton cluster installations that are dedicated to the SoftFIRE Project. A dedicated deployment was chosen to minimize the interference from other active projects on to the Experiments. The setup is based on one combined controller, compute and networking host and one additional compute node for the "pure" OpenStack installation. For the SDN featured version of OpenStack the Controller also houses the OpenSDNcore controller that acts as a master controller for each of the OpenSDNcore vSwitches that are deployed in each compute host. The used servers are manufactured by Dell and are in the Blade form factor. Table 2. FOKUS OpenStack Hardware lists the details of the used server hardware. Storage Capacity is provided by a central Storage Array Network (SAN) manufactured by NetApp accessed via GBit 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 in realized as an external provider network that is connected via the OpenSDNcore vSwitches (or the Networking host of the OpenStack) and dedicated to the SoftFIRE project.

Table 2. FOKUS OpenStack Hardware

	Type	RAM	CPU	Storage
Controller & Compute 1	Dell PowerEdge M620	128 GB	2x Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz (32vcores)	360 GB SSD 1 TB HDD NAS: 500 GB
Compute 2	Dell PowerEdge M620	128 GB	2x Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz (24vcores)	150 GB SAS NAS: 500 GB
Controller & Compute & OpenSDNCore	Dell PowerEdge M620	128 GB	2x Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz (32vcores)	150 GB SAS NAS: 500 GB

Orchestration and Management Services

The resources of the Federated SoftFIRE testbed are managed via SoftFIRE Experiment Manager (SEM) and Open Baton 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 on dedicated servers. For security reasons, these services are installed on a dedicated hardware that cannot be accidentally modified by the SoftFIRE components. The VPN router that manages the Interconnection to the VPN hub at TUB is realized as a virtual instance in a different virtualization environment. This second virtualization environment is realized by a VMware ESX cluster. A Zabbix [7] proxy service is provided on the same instance to support the collection of KPI values.

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 testbed 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 end-to-end (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
- Proof-of-Concept (PoC) whilst connected to experimenters' equipment or remote site
- Validation and Certification on Customer specific solution

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 (Newton version) system access to infrastructure that can be used to instantiate core slices for experimentation with the local RAN components.
2. Access to the 5GIC in-building LTE-A RAN
3. Access to the 5GIC in-building Wi-Fi system for multi-access 5G use cases

In order to deliver the infrastructure as a service (IaaS) capabilities for creating and managing as a data-centre, the following network hardware is provided for experimenter use, as shown in Table 3.

Table 3: Network Equipment

Description (single node configuration)	Qty
Dell PowerEdge R920 (deployed as SoftFIRE OpenStack Controller and Compute server) **Total 12 CPUs available for experimenter VMs**	1
Indoor, Wi-Fi Access Points **Wi-Fi 802.11ac Access Points from Aruba**	6
Indoor, LTE-A, FDD, Femto-cells, Band 20	1

Resources available per Experimenter at UoS Testbed

Description (single node configuration)	Qty
CPUs	4
Disk storage	<=80 GB
RAM	16 GB

Please note that in OpenStack environment, this can support 2 "m1.medium" flavours (2 CPUs, 40 GB disk space, and 8 GB RAM).

2.2.5. Deutsche Telekom (DT)

The SoftFIRE project has also worked towards the extension of the Federated testbed. Starting from the Second Open Call some infrastructure resources from a testbed of DT have been integrated and made available for the experiments.

The testbed contribution of DT is part of a larger OpenStack cluster located in a commercial data center in Berlin, which is used for a number of innovation projects. Currently the OpenStack release "Mitaka" is installed but the migration to the "Newton" release will be available before the start of the Third Open Call. SoftFIRE will have access to a portion of the overall resources as described in the table at the beginning of chapter 2.2. Depending on the demand from the experimenters there is some room to increase the virtual resources during the Open Call if needed.

2.2.6. Assembly Data System (ADS)

ADS SoftFIRE testbed is located in Rome. The aim of this environment is to set up a Lab to develop NFV skills inside ADS Software Factory and provide a platform for customer demo and PoC. The Lab will be also an optimal working environment for the development and test of own-brand VNF and benchmarking cloud virtual network performance.

ADS, has based its NFV infrastructure on Red Hat Openstack 10 (Newton) released on December 2016. It's the first Long Term Release that will have a support life cycle up to five years. This release is suggested for real production infrastructures for Telco Operators, who need stable and supported environments. Currently the ADS NFV platform consists of 5 nodes with the following roles:

Qty	Role	Hardware Description
1	Director	DELL Power Edge - 2x Quad Xeon with 16 GB Ram
1	Controller	DELL Power Edge - 2x Quad Xeon with 16 GB Ram
2	Compute	DELL Power Edge - 2x Quad Xeon with 48 GB Ram (Tot 96 GB)
1	Storage (CEPH)	DELL Power Edge - 2x Quad Xeon with 16 GB Ram
1	Monitoring and Management Tools	SUN Server X4-2 - 2x Quad Xeon with 32 GB Ram

The Private Cloud will be interconnected within the SoftFIRE project in order to make experiments on a geographically distributed NFV platform.

The storage backed of the infrastructure is based on Ceph Software Defined Storage, which provides a High Availability and High Scalability Object and Block Storage on general-purpose hardware.

The integration with OpenDayLight will enable more advanced SDN functionalities than those provided by default OpenStack Neutron module.

Furthermore, compute nodes leverage on Real Time KVM Hypervisor to virtualize VNF requiring Real Time functionalities.

2.2.7. Other Testbeds

For interested companies, there is the possibility to integrate other testbeds in order extend the functionalities and capabilities offered by the Federation. In case there is this need, we recommend getting in touch with the SoftFIRE contacts.

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 was the integration in the federated testbed of the EIT Digital Silicon Valley testbed. It could be the entry point for further collaborations with organizations in the USA.

2.3 SoftFIRE Middleware Architecture and supporting services

The SoftFIRE middleware architecture has gone through a deep restructuring and consolidation in order to provide better services to experimenters.

With reference to Figure 1, several "managers" have been designed, implemented and provided to experimenters in order to support the experiment lifecycle.

The identified managers are *NFV Manager*, for virtualized resources, *SDN Manager* for the allocation of the SoftFIRE provided SDN functionalities, a *Security Manager* that will support the allocation of security functions made available by SoftFIRE and usable by the experimenters, the *Monitoring Manager* for some related monitoring functionalities. The NFV Manager and the SDN Manager are under development and will be provided in time for the second wave. The other managers will be progressively introduced in order to be used in the following wave of experimentations or Hackathons/Challenge.

It is important to clarify that access to the individual OpenStack instances will be granted only via the SoftFIRE Experiment Manager (i.e., the SoftFIRE Middleware), and in most of the cases OpenStack APIs won't be available to be directly consumed by experimenters.

Nevertheless, the testbed will be aligned with industry-oriented standardisation efforts: TOSCA is exposed to experimenters for deploying and provisioning resources on the federated infrastructure. This means that SoftFIRE generic resources are described as TOSCA nodes and the SoftFIRE Experimenter Manager will support the "translation" and allocation of the requested resource onto the SoftFIRE available resources. Experimenters could already start to get familiar with the TOSCA APIs and functionalities offered by the Open Baton framework [3]. Although the final ones, which will be exposed by the SoftFIRE middleware, may be slightly different, the SoftFIRE consortium will try to maintain compatibility with the functionalities provided by the Open Baton APIs. SoftFIRE will make use of the Open Baton 4th release which has been launched by the end of April 2017. The Open Baton installation in SoftFIRE provides the following components: NFVO, Generic VNFM, Auto-scaling Engine, Zabbix Plugin, and OpenStack driver. Experimenters could also consider extending the set of components provided by the Open Baton framework. For this they could make use of the Open Baton SDK and build either a new VIM driver, Monitoring driver, VNF Manager, or external component (using the event mechanism, please refer to the documentation). In this case, the experimenter should host the additional developed components on their own premises and interconnect them to the Open Baton framework via the RabbitMQ message bus.

Monitoring functionalities will be implemented by the Zabbix monitoring system, and will be provided as a service to experimenters.

Additional Security features will be added in order to further enrich the potential number of use cases to be developed on the federated testbeds. Complex open source security-oriented VNFs will be made available to the experimenters, offering also Network Intrusion Detection System with Deep Packet Inspection capabilities (Suricata NIDS). Virtual firewalling capabilities will be also enhanced giving the possibility to manage and recreate real case scenarios.

Figure 7 shows the generic process by which the programmers can use the federated testbed. The picture indicates how the federated testbed will grant access and will provide allocation and usage of resources. A registration phase is needed and when an experiment has been authorized and will be done automatically, the Experimenter will get a certificate to be used for the experiment team in order to access to the SoftFIRE resources (the SoftFIRE web portal will be used in order to distribute the certificates. When the access to the infrastructure is granted, the experimenters will be able to request resources. If they are available they will be allocated and an acknowledgment will be returned to the users) and instantiated (essentially a virtual instance of them will be created and exclusively instantiated for the experiment) and hence usable for the specific goals of the experiment. The SoftFIRE Experiment Manager segments the experiments in such a way to avoid interference between different usages of the platform (e.g., other experiments).

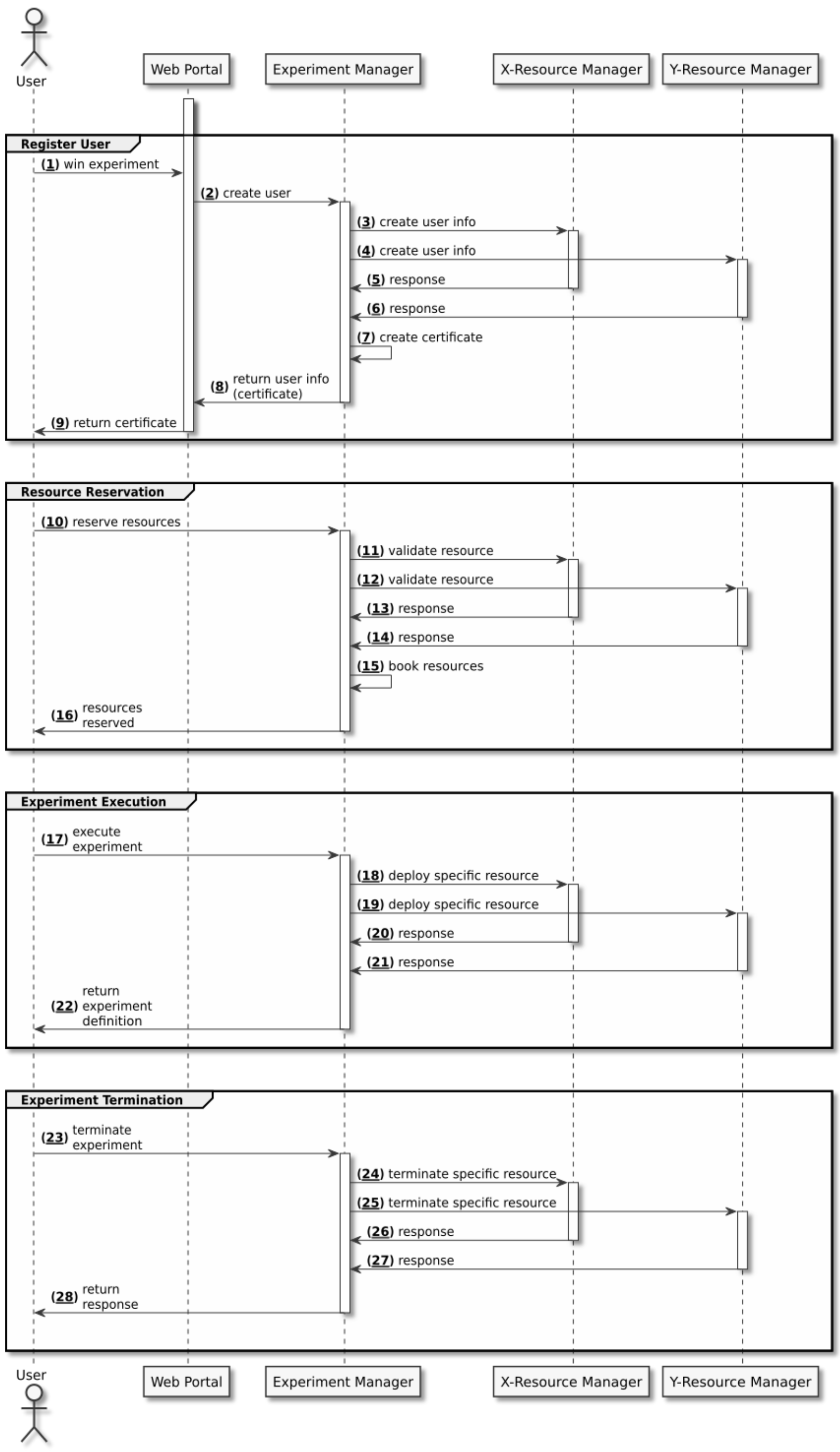


Figure 7. Experimenter interaction

2.3.1. NFV Features

The experimenter will be dealing with NFV Resources in every experiment since NFV is the core technology exploited in the SoftFIRE Middleware. The Experimenter can reserve and deploy

Network Services on demand or can also define his own Network Service to be deployed in one or more testbed of the SoftFIRE infrastructure.

The preconfigured Network Services available are defined in the Section 2.4.1 where all the Virtual Resources are described. Please note that due to copyright constraints some Network Services are constrained to be deployed in a specific testbed and the experimenter will not be able to access the deployed VMs.

In particular, the SoftFIRE middleware NFV features are provided by the ETSI NFV MANO compliant orchestrator Open Baton [3]. For that reason, in order to define a different Network Service, you may find of help the Open Baton documentation and tutorial on CSAR definition [9], however more details will be provided in the section 3.2.1.

2.3.2. SDN Features

From the second wave of experiments, the SoftFIRE middleware provides SDN programmability features to the experimenters. This enables them to create innovative experiments and use cases that make use of advanced networking features that are provided by the SDN middleware components in the SoftFIRE platform. Currently the project is working towards the possibility to introduce Service Function Chaining. In case these functions will be made available, the project will inform and advertise it through www.softfire.eu portal.

In addition to the changes on the SoftFIRE Middleware, the SoftFIRE consortium is providing SDN technologies either as virtualised or physical entities available in some of the testbeds provided. Currently, some of the individual testbeds already integrated OpenDaylight controller (ODL Boron SR-2) under OpenStack Neutron to manage the network flows on the compute nodes via the OVSDB 2.7.0 south-bound plug-in. Network programmability would be realised via a set of OpenDaylight RESTCONF APIs exposed by the ODL controller.

Fraunhofer FOKUS offers access recent developed integration of OpenSDNcore SDN platform into the OpenStack neutron networking. The OpenSDNcore Controller in collaboration with the OpenSDNcore vSwitch integrated into the compute servers provide full control of the network used by all NFVs deployed on this particular OpenStack. The experimenters can make use of the JSON-RPC based API of the Controller to implement advanced SDN features like Service Function Chaining (SFC), Traffic redirection and load balancing.

However, the access to the individual controller API will be subject to mechanisms to lower the risk of interfering among virtual object user data configuration. Therefore, access to the SDN technologies may be granted in such a way that each experiment could work independently without interferences from other running experiments.

Due to the fact that each Testbed uses different controllers and vSwitch implementations the SoftFIRE platform can’t offer SDN WAN features during the third open call phase.

Please note that these specifications may change according to the progress and the enrichment of the Testbeds.

2.3.2.1. Fraunhofer FOKUS OpenSDNCore

The OpenSDNCore developed by Fraunhofer FOKUS provides advanced SDN features that are tightly integrated into the OpenStack platform. This platform can be used to develop new kind of SDN applications, or utilize the current features to implement custom use cases based on Flow rules. The Platform provides the following features:

- Dedicated flow tables for each Experiment

- Access is limited to this flow tables and to rules that directly match on ports and mac-addresses used by the Experiment
- The Open Stack integration uses predefined set of flow tables to realize basic features like, ARP, Floating-IPs, Internet-Access, Tenant network separation.
- Traffic will traverse tables 0 for classification and 6 for firewalling and is then forwarded to application specific tables before it leaves the switch via an output port.
- Tenant specific flow-tables are inserted just after the firewall before application specific tables

The SDN controller exposes a JSON-RPC API that is used to control the inner workings of the Controller and all connected vSwitches. The extensive documentation of this API can be found on the SoftFIRE SDN documentation website. [1]

2.3.2.2. OpenDayLight NetVirt

OpenDaylight OPENFLOW PLUGIN RESTCONF APIs are exposed by the Boron SR-2 ODL controller for Network programmability. The user can take advantage of Network Virtualization by using OpenDaylight SDN functionalities whilst utilizing OpenvSwitch. The stack includes a Neutron Northbound, a Neutron Virtualization Layer and an OVSDB southbound plugin and an OpenFlow southbound plugin.

To allow experimenters to operate their own network programmability schemas preventive measures have been adopted such as limiting the access to experiment dedicated flow tables along with involving OVS port and MAC address used by the experiment.

2.3.3. Monitoring Features

Experimenter Monitoring functionality is made available as a service to experimenters who needs to get monitoring data about their own VM.

The monitoring solution offered by SoftFIRE is based on the open source monitoring software Zabbix. The monitoring service comprises two major software components: Zabbix server and Zabbix agent.

The Monitoring Resource (Zabbix Server) request is performed via Tosca API exposed by the Experiment Manager and must be defined in resource request definition Template along with information on which testbed to deploy.

The server is deployed in a separate and dedicated resource into the indicated testbed.
At the Request Resource completion, the experimenter is informed about the Zabbix Server floating IP along with user credential preconfigured to access.

The Zabbix agent will be installed in all the VMs the experimenter describes into the NFV node_types description.

Many metrics are being monitored in VM level and are provided as standard metrics and can be activated or deactivated on request. Some of these, but not limited to, per VM are: total memory, used memory, free memory, total swap memory, used swap memory, free swap

[1] http://docs.softfire.eu/opensdncore/

memory, total storage, used storage, free storage, CPU load, CPU utilization (%), CPU count, network metrics (e.g. incoming and outgoing traffic in interfaces, OS-related metrics, processes-related metrics (e.g. number of running processes), services-related metrics (e.g. FTP-/Email-/SSH-server is running), etc.

SoftFIRE does not provide application monitoring metrics addressing information about the state of the running application, its performance and other application specific information. These metrics are not provided by default by Zabbix, the experiments need to explicitly configure them at the agent and the server levels.

2.3.4. Security Features

The Security Manager inside the SoftFIRE Middleware makes available to the Experimenter a series of security related functionalities that she might decide to include and use within her activities on the SoftFIRE platform.

Here is the list of the available features.

1. The Experimenter can deploy a Security Resource
2. The Experimenter can statically configure her Security Resource by means of its descriptor
    a. The Experimenter can enable logs collection from her Resource
    b. The Experimenter can statically configure some rules on her Resource
3. The Experimenter can dynamically configure her Resource once it has been deployed
4. The Experimenter can see her Resources logs in a web dashboard
5. The Experimenter can perform searches among her Resources logs in a web dashboard
6. The Experimenter can see statistics related to her Resources logs in a web dashboard

2.3.4.1. Security Resources

A Security Resource is a commonly used security agent that the Experimenter can include in her experiment. She can access and configure it through a static initial configuration, included in the TOSCA description of the Resource, or, once deployed, through a REST interface that exposes its main services.

The Experimenter can also ask the Security Resource to send its log messages to a remote log collector, which makes them available in a simple web page reserved to her. The Experimenter could easily access it through its web browser and check the behaviour of her all security agents, and to see some statistics.

The Experimenter can get the Security Resource in two different formats:

- As an agent directly installed in the VM that she wants to monitor. The system will provide her a script that the Experimenter has just to run inside the VM. It will be already configured as required in the TOSCA description of the resource. The output of the script will provide to the Experimenter information on how to access the deployed resource (URLs, etc.)
- As a standalone VM the Security Resource will be deployed directly by the Security Manager in the testbed chosen by the Experimenter. The Security Manager will take care of the initial configuration of the resource.

The Experimenter has to set up on her own the redirection of the network traffic that she wants to control through the Security Resource VM (by means of tunnelling or SDN capabilities).

To date the only Security Resource available on the SoftFIRE environment is the firewall.

2.3.4.2. Firewall

A firewall is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined rules.

The available firewall resource is built upon Ubuntu UFW (Uncomplicated FireWall), to which a control system, based on a ReST interface, has been added.

The firewall agent is available for Ubuntu OS only.

The rules that can be defined on this type of firewall are stateless (they do not maintain information about the context). It works as a packet filter, which looks at network addresses, ports and protocols.

Services specifically available for the firewall Resource are:

1. The Experimenter can statically define a list of allowed IP addresses (or CIDR masks)
2. The Experimenter can statically define a list of denied IP addresses (or CIDR masks)
3. The Experimenter can statically define the default behaviour of the firewall
4. The Experimenter can get the status of the firewall
5. The Experimenter can get the rules installed on the firewall
6. The Experimenter can dynamically add a rule to the firewall
7. The Experimenter can dynamically update a rule on the firewall
8. The Experimenter can dynamically remove a rule from the firewall

2.4 Additional resources available to the experimenters

Currently the project and its partners are developing additional functionalities and related virtual machines. Provided that an acceptable level of stability and robustness is reached, the federated testbed will be complemented with such sets of well formed and ready to use network functions. The programmers will then be able to use them and 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.

2.4.1. Virtual Resources

2.4.1.1. Widely deployable

- OpenIMSCore (Fokus):
  The Open IMS Core is an Open Source implementation of IMS Call Session Control Functions (CSCFs) and a lightweight Home Subscriber Server (HSS), which together

form the core elements of all IMS/NGN architectures as specified today within 3GPP, 3GPP2, ETSI TISPAN and the Packet Cable initiative. The four components are all based upon Open Source software (e.g. the SIP Express Router (SER) or MySQL). For more details please visit the OpenIMS website[2]

2.4.1.2. FOKUS

- Open5GCore (Fokus):
  Open5GCore is a prototype implementation of the pre-standard 5G network. The software is available from November 2014 and its main features are described on www.open5gcore.net. Open5GCore represents the continuation of the OpenEPC project towards R&D testbed deployments. It has been used over the years in multiple projects as a reference vEPC implementation. The following limitations apply when this VNF is used by Experimenters:
  - This VNF can only be used inside the Fraunhofer FOKUS testbed due to licensing constrains.
  - It will provide a complete virtualized 5G-core network with virtualized UE components and benchmarking features.
  - Interconnection of physical RAN equipment is also supported; however, the experimenter then needs to be present into the lab of Fraunhofer FOKUS in Berlin

2.4.1.3. University of Surrey

The UoS testbed provides LTE-A core network slices with added 5G features. UoS runs VMs for a User Plane (UP) slice where a dedicated UP node (UPN) runs for an experimenter, as well as a single VM where a Control Plane (CP) node (CPN) runs. Table 4 briefly describes the virtual core network components.

Table 4: Network Services Provided by the UoS Testbed

Network Service	Max # of instances	VNFs	Description
CPN	1	HSS, MME, integrated (SGWc, PGWc)	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. 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. The PLMN-id used is 235 91 which is a Vodafone UK test id with label “5GSF”
UPN(CC)	3	CC,	This Network Service is instantiated per experimenter for LTE User Plane service and

[2] http://www.openimscore.org/

		Integrated (SGWu, PGWu)	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). This slice provides best performance for Internet access. Each authorized experimenter is provided with a single instance of this VM on the UoS testbed.
UPN(CM)	3	CM, integrated (SGWu, PGWu) = a.k.a. Packet Processing Entity (PPE)	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). This slice provides best performance for Intranet access via one of the Cluster Members. CM has the interface to an experimenter-provided server application (that would run on an experimenter VM) Each authorized experimenter is provided with a single instance of this VM on the UoS testbed.

For the Third Open Call, these services will be offered by UoS:

- A CUPS evolved EPC as UPN and CPN operating with an already deployed CPN. A single UPN will be instantiated for each approved experimenter on the UoS testbed, whereas the CPN is shared among experimenters.
- The Experiments must be related with and designed to work with LTE EPC. The Experimenter VNFD must be a virtualised server application to be deployed on a VM on UoS OpenStack. This VNF is to interact with EPC on the standard SGi interface (to LTE PGW). The Experimenter must develop an Android app that would interact with this server application. UoS will provide an Android library as well as the API to pass and receive messages to/from the server application that the Experimenters are to develop.
- The Experimenters are to submit their app as an APK to UoS at the following contact: Dr. Serdar Vural (s.vural@surrey.ac.uk).
- Additional functionality for reservation of mobile phones remotely is available via the Physical Resource Manager of the SoftFIRE Experiment Manager. This also provides the experimenter with some remote control capabilities of the mobile phone.
- The UoS RAN or core network infrastructure cannot be modified to fit specific purposes of an experiment. Exceptions can be made for exceptionally interesting experiments that add additional capability to the infrastructure.
- The CPN VM (shared by all experimenters) and the UPN VMs (CC and CM) assigned to an experimenter are assigned specific IP addresses by UoS. The experimenter VMs, if they need mobile data connectivity, can talk to the mobile core on these IP addresses.

- Experimenter Android apps must be compatible with stock (Google) Android >=7.1.1. UoS can provide one mobile for each experimenter, or multiple mobiles depending on the number of simultaneous experimenters using the UoS equipment.

Mobile Device Requirements

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

Aspect	Specification	Comment
OS	Stock Android >=7.1.1 required!	Easier to develop research code with more parameters exposed than iOS
LTE Bands	B20	These bands operate on Femto cell RAN equipment connected to SoftFIRE's softcore.
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.

2.4.2. Physical Resources

2.4.2.1. FOKUS

The Fraunhofer FOKUS testbed provides access to a physical LTE eNodeB femtocell and PC or Android based UE that can be used by experimenters if they visit the Testbed located in Berlin. Experimenters can also bring their own LTE equipment to be used with the experiment. However in case of own eNodeB devices please base in mind that FOKUS Testbed has a very restrictive license in regard to frequency band and transmission power.

Device	Vendor	LTE Bands	Qty	Notes
Phone	Google Nexus 5	*	1	Android 6
Phone	Samsung S4	*	2	Android 6
PC-Dongle	Huawei E3372	13, 7, *	2	
LTE eNodeB	Ip-access	13, 7	1	LTE Release 10
WiFi AP	Various	802.11ac		On Request

2.4.2.2. University of Surrey

University of Surrey provides an indoor LTE femtocell and 6 WiFi access points over two floors of the 5GIC building. The following table summarises the available physical resources. Experimenters should note that 5GIC has license for LTE FDD Band 20.

Device	Vendor	LTE Bands	Quantity	Notes
Phone	Google Pixel	20	3	Android 7.1
LTE FDD Femtocell	ip.access	20	1	
WiFi AP	Aruba	802.11ac	6	Separate WiFi SSID for SoftFIRE experimenters

To use the physical resources of the UoS testbed, the experimenters are to provide an Android app. UoS staff will place the three mobile phones in a fixed location – the use of the mobile phones are to be shared by multiple experimenters. However, depending on the number of experimenters requiring mobile phones, there can be different scenarios regarding the use of the mobile phones: one phone per experimenter, or shared mobiles among experimenters.

The UoS testbed provides a set of 5G capabilities; mainly control and user plane separation (CUPS), as well as an improved user plane slice. The app to be designed by an experimenter needs to include a library APK provided by UoS, and two special simple identifiers to send/receive application messages to/from the network.

5GIC provides a separate virtual user plane (UP) for each experimenter on its OpenStack environment, with a maximum of 3 UP slices, i.e. 3 experimenters at a time when all such experimenters require the use of a virtual mobile core network.

The experiment scenario that uses the 5GIC testbed (radio access network, the virtual core VM, and the UP slice) is a client-server application, in which a user app running on Android mobile phone(s) communicates with a server application running on a virtual machine (VM). Since this is a well-tested scenario, UoS requires experimenters to follow this scenario, should they wish to use the physical resources in UoS.

3 How to use SoftFIRE

The SoftFIRE Middleware provides to the experimenter different APIs. The usage of these APIs is described in the following sections. This part of the document will guide the experimenter through the process of designing, provisioning and terminating the experiment. The experiment overview steps are described in Figure 8 and it defines the steps of the experiment lifecycle and the usual execution order.

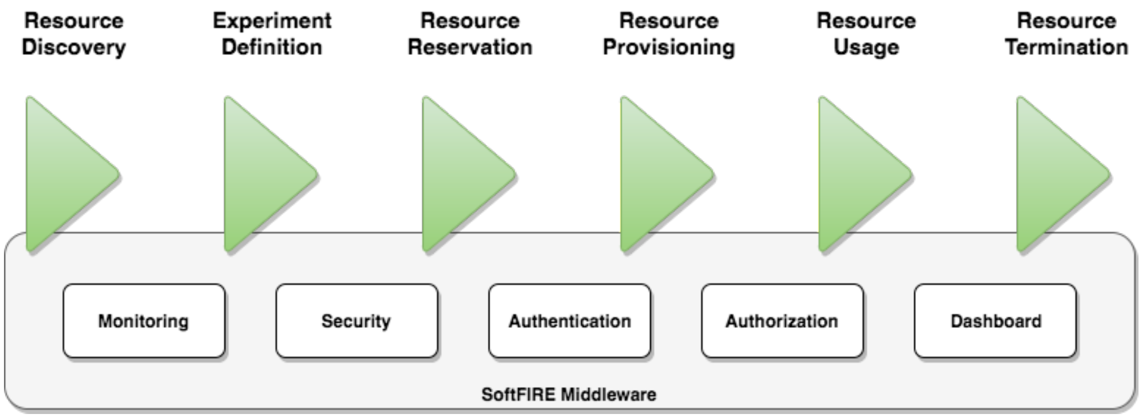


Figure 8: Experiment Lifecycle

However, at the time of writing this tutorial, new development has been done on the SoftFIRE middleware. For that reason, we suggest to follow the more up to date SoftFIRE Experimenter guidelines and usage manual, always available here at our documentation website [5].

The life cycle of an experiment is depicted in Figure 8:

- **Design phase**
    - List resources: listing all the available resources.
    - Define the experiment: based on the previous step, decide what to reserve and define the experiment in order to fulfil your requirement.
- **Provision phase**
    - Resource reservation and provisioning: first 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). Then deploy the experiment defined in the previous step.
- **Runtime phase**
    - Resource monitoring: once deployed, if the monitoring of the resources were selected in the previous steps, it will be possible to monitor the deployed resources using a monitoring server dedicated to the experimenter
    - Resource control at runtime: It is also possible to access remotely some of the virtual resources when deployed correctly and additionally some external services, such as auto scaling will be available.
- **Closing phase**
    - Resource termination: when the experiment is finished, either because the time has expired or because the experimenter has finished, the resources will be released.

3.1 Getting Access to the platform

3.1.1. Get OpenVPN certificate

OpenVPN is used to provide access to the SoftFIRE network based on certificates. Each accepted experiment will receive a certificate that can be used to access the OpenVPN service for the duration of the open call period. The certificate will be generated by the SoftFIRE Portal, please follow the following steps to retrieve your personal certificate.

1. Register to the SoftFIRE portal
2. Get accepted by the consortium
3. Download the OpenVPN configuration file from the SoftFIRE portal

If the automatic generation of OpenVPN configuration file fails or if your client software needs a different format of configuration file, additional documentation is available on the SoftFIRE Software documentation website[3].

3.1.2. OpenVPN setup

1. Install an OpenVPN client matching your operation system
    - Linux: openvpn packages available from the distributions repository
    - Mac OS: various versions available, we have tested “tunnelblick”
    - Windows: OpenVPN-GUI was tested by us but the usage of windows is not recommended
2. Import the downloaded configuration file into your chosen OpenVPN client application
3. Connect to the VPN server using the client application

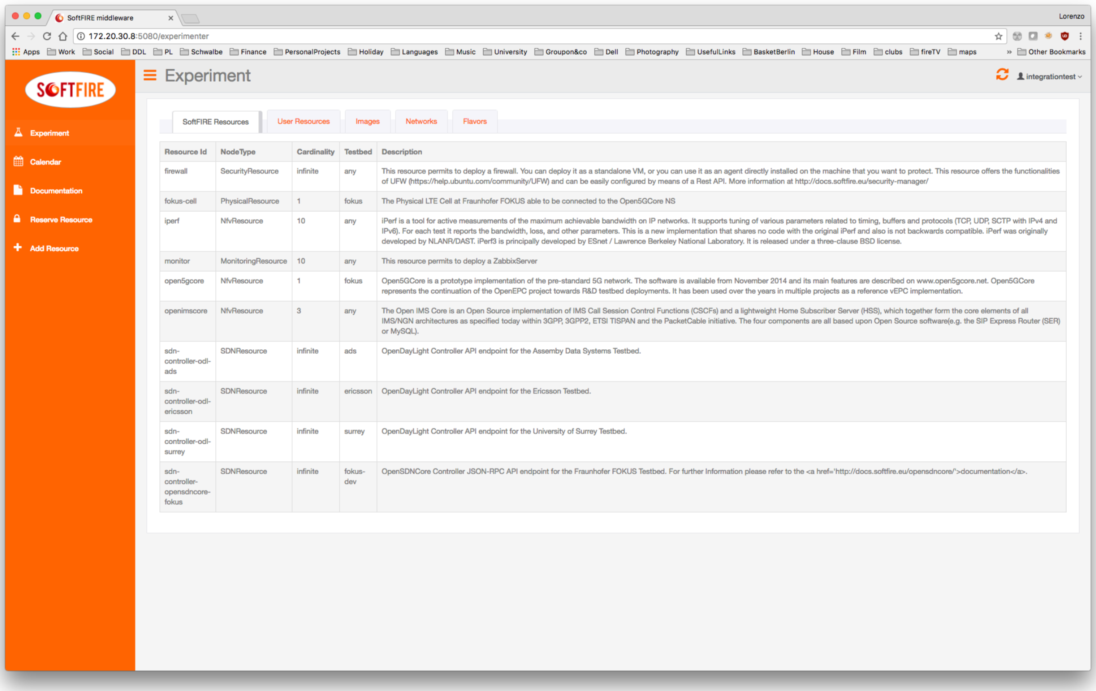
3.1.3. Register to Experiment Manager

The registration of an Experimenter to the Experiment Manager will be done automatically so there is not any relevant information regarding this matter that concerns the Experimenter. In next releases of this document we will provide details for platform extension development purposes. However, always be up to date by using this documentation available online [5].

[3] http://docs.softfire.eu/openvpnconfig/

3.2 Running the experiment

Figure 7 defines the interactions between the Experimenter and the SEM. These steps are described in the following subsections. The reference page is the Experimenter Page shown in Figure 9.



Experiment

Resource Id	NodeType	Cardinality	Testbed	Description
firewall	SecurityResource	infinite	any	This resource permits to deploy a firewall. You can deploy it as a standalone VM, or you can use it as an agent directly installed on the machine that you want to protect. This resource offers the functionalities of UFW (https://help.ubuntu.com/community/UFW) and can be easily configured by means of a Rest API. More information at http://docs.softfire.eu/security-manager/
fokus-cell	PhysicalResource	1	fokus	The Physical LTE Cell at Fraunhofer FOKUS able to be connected to the Open5GCore NS
iperf	NfvResource	10	any	iPerf is a tool for active measurements of the maximum achievable bandwidth on IP networks. It supports tuning of various parameters related to timing, buffers and protocols (TCP, UDP, SCTP with IPv4 and IPv6). For each test it reports the bandwidth, loss, and other parameters. This is a new implementation that shares no code with the original iPerf and also is not backwards compatible. iPerf was originally developed by NLANR/DAST. iPerf3 is principally developed by ESnet / Lawrence Berkeley National Laboratory. It is released under a three-clause BSD license.
monitor	MonitoringResource	10	any	This resource permits to deploy a ZabbixServer
open5gcore	NfvResource	1	fokus	Open5GCore is a prototype implementation of the pre-standard 5G network. The software is available from November 2014 and its main features are described on www.open5gcore.net. Open5GCore represents the continuation of the OpenEPC project towards R&D testbed deployments. It has been used over the years in multiple projects as a reference vEPC implementation.
openimscore	NfvResource	3	any	The Open IMS Core is an Open Source implementation of IMS Call Session Control Functions (CSCFs) and a lightweight Home Subscriber Server (HSS), which together form the core elements of all IMS/NGN architectures as specified today within 3GPP, 3GPP2, ETSI TISPAN and the PacketCable initiative. The four components are all based upon Open Source software(e.g. the SIP Express Router (SER) or MySQL).
sdn-controller-odl-ads	SDNResource	infinite	ads	OpenDayLight Controller API endpoint for the Assemby Data Systems Testbed.
sdn-controller-odl-ericsson	SDNResource	infinite	ericsson	OpenDayLight Controller API endpoint for the Ericsson Testbed.
sdn-controller-odl-surrey	SDNResource	infinite	surrey	OpenDayLight Controller API endpoint for the University of Surrey Testbed.
sdn-controller-opensdncore-fokus	SDNResource	infinite	fokus-dev	OpenSDNCore Controller JSON-RPC API endpoint for the Fraunhofer FOKUS Testbed. For further Information please refer to the <a href='http://docs.softfire.eu/opensdncore/'>documentation</a>.

Figure 9. SEM Experimenter page

In the following sections, we will refer to the testbeds. Each testbed as a string id that will be used in the TOSCA properties of some nodes.

- **ericsson:** *RMED Cloud Lab* from Ericsson, located in Rome
- **fokus:** FUSECO Playground from FOKUS Fraunhofer, located in Berlin
- **surrey:** *5GIC* from University of Surrey, located in Guildford, Surrey
- **ads:** *Assembly Data System Testbed,* located in Rome
- **dt:** *Deutsche Telekom Testbed*, located in Berlin

3.2.1. Design phase

During this phase, the Experimenter is required to define his experiment. This requires that the Experimenter is aware of the available resources.

3.2.1.1. Resource discovery

First step is to list which resources are available to be requested. In Figure 9 the available resources are listed in the left of the orange and grey table. This list has three important columns:

- **Name:** This is an id of the resource and must be used later when defining the experiment

- **Description:** This is a detailed description of the resource. Please read it carefully before deploying the resource, since it may not fit your requirements
- **Node Type:** this filed is the TOSCA node type associated to that resource.

3.2.1.2. Experiment definition

Once you are aware of the available resources, you can start defining the experiment. The experiment is defined using Topology and Orchestration Specification for Cloud Applications (TOSCA) [4]. In particular, the Experimenter has to create a CSAR [10] zip file containing all the necessary files and definitions for letting the Experiment Manager instantiate everything. The structure of the CSAR is defined as following:

```
├── Definitions
│   └── experiment.yaml
├── Files
│   └── nsd.csar
└── TOSCA-Metadata
    ├── Metadata.yaml
    └── TOSCA.meta
```

There are three two mandatory folders plus one optional. *Definitions* and *TOSCA-Metadata* are mandatory folder containing the metadata files and the experiment definition, as described in the following sections. The *Files* folder contains some additional files needed in case some resources specify extra requirements.

3.2.1.2.1. TOSCA-Metadata

The TOSCA-Metadata folder contains the TOSCA.meta file and the Metadata.yaml file. The TOSCA.meta file must contain the reference to the template in this case *Entry-Definitions: Definitions/experiment.yaml*.

For example:

```
TOSCA-Meta-File-Version: 1.0
CSAR-Version: 1.1
Created-By: MyCompany
Entry-Definitions: Definitions/experiment.yaml
```

The Metadata.yaml contains experiment Meta information:

- the name
- the start date
- the end date

As follows:

```
name: Experiment Name
start-data: "9/5/17"
end-data: "10/5/17"
```

3.2.1.2.2. Definitions

The *experiment.yaml* must follow a specific structure. An example is show in the following lines:

```
tosca_definitions_version: tosca_simple_yaml_1_0
description: Template for SoftFIRE yaml resource request definition
imports:
  - softfire_node_types: http://docs.softfire.eu/etc/softfire_node_typ
es.yaml
topology_template:
  node_templates:
    zabbix_server:
      type: MonitoringNode
      properties:
        resource_id: monitoring
        testbed: ericsson
    sdn_ericsson:
      type: SDNResource
      properties:
        resource_id: sdn_ericsson
        testbed: ericsson
    iperf:
      type: NFVResource
      properties:
        resource_id: iperf
        nsd_name: The Iperf NSD
        testbeds: { ALL: ericsson }
```

As shown in the example, the SoftFIRE experiment yaml file must contain the TOSCA definition version as *tosca_definitions_version: tosca_simple_yaml_1_0*. This line is followed by a description of the experiment.

The imports section must be specified as in the example because the EM will only accept specific node types defined in [11]

Each node type specifies a *resource_id* that must be chosen from the available resources. The node name is arbitrary. Each node type can have some additional properties and they can be different from each other's. Check the node type specification [11] to understand all the node types.

3.2.1.2.2.1 NFV Resource

The NfvResource node type is defined as follows:

```
 NfvResource:
    derived_from: eu.softfire.BaseResource
    description:  "Defines  a  NFV  resource  request  in  the  SoftFIRE
Middleware"
    properties:
      testbeds:
        type: map
        entry_schema:
          description: "mapping between vnf types and testbed. Or
                       'ANY':<testbed_name> for all in one"
          type: string
      nsd_name:
        type: string
        description: "Name of the NS to deploy"
      file_name:
        type: string
        description: "relative path to the Open Baton CSAR. It is always starting with Files/..."
        required: false
      ssh_pub_key:
        type: string
        required: false
        description: "the public ssh key that will be injected in the VM in order to give access to the experimenter"
```

This node type has different properties:

- **resource_id:** The resource id that can be found from the list resource table.
- **testbeds:** a map where you can define the testbed where each VNF will be deployed. It is defined as **vnf type** and **testbed name**
- **nsd_name:** the name of the NS
- **file_name:** in case the preconfigured NS are not sufficient for your experiment you can upload your own NS in CSAR format and place it in the *Files* folder. This field contains the name of the file

3.2.1.2.2.2 SDNResource

The SDNResource type is defined as follows:

Copy definition here

```
  SDNResource:
    derived_from: eu.softfire.BaseResource
    description: Defines a SDN resource request in the SoftFIRE Middleware
```

The SDNResource type does not provide any additional properties besides the one inherit from the BaseResource type:

- **resource_id:** The resource id that can be found from the list resource table. Depending on this id the testbed that is used to provide the SDN resource implicit is chosen.

3.2.1.2.2.3 MonitoringResource

The MonitoringResource node type is defined as follows:

```
MonitoringResource:
    derived_from: eu.softfire.BaseResource
    description: Defines the Zabbix monitoring resource requested
    properties:
        testbed:
            type: string
            required: false
            description: Location where to deploy the monitoring server
```

This node type has the following properties:

- **resource_id:** The resource id that can be found from the list resource table
- **testbed:** the testbed where you want the Zabbix server to be deployed.

3.2.1.2.2.4 SecurityResource

The SecurityResource node type is defined as follows:

```
SecurityResource:
    derived_from: eu.softfire.BaseResource
    description: Defines a Security agent to be deployed. More details
on *docu_URL*
    properties:
        resource_id:
            type: string
            required: true
            default: firewall
        testbed:
            type: string
            required: false
        want_agent:
            type: boolean
            required: true
            default: true
        logging:
            type: boolean
            required: true
            default: true
        allowed_ips:
            type: list
            entry_schema:
                type: string
            required: false
        denied_ips:
            type: list
            entry_schema:
              type: string
            required: false
        default_rule:
            type: string
            required: true
```

This node type has different properties:

- **resource_id**: Defines the type of the Security Resource. To date only *firewall* is accepted
- **testbed**: Defines where to deploy the Security Resource selected. It is ignored if want_agent is True
- **want_agent**: Defines if the Experimenter wants the security resource to be an agent directly installed on the VM that she wants to monitor

- **logging**: Defines if the Experimenter wants the security resource to send its log messages to a collector and she wants to see them on a dashboard
- **allowed_ips**: List of IPs (or CIDR masks) allowed by the firewall. [allow from *IP*]
- **denied_ips**: List of IPs (or CIDR masks) denied by the firewall [deny from *IP*]
- **default_rule**: Default rule applied by the firewall (allow/deny)

3.2.1.2.3. Files

This folder contains an inner CSAR of a NS. This is only used in case the NFV Resource you want to deploy is not one of the preconfigured one. In this case, the how to build this CSAR is explained in [9]. And the NfvResource *file_name* field must point to this file.

Moreover, there is some additional information that can be configured and that are specific of the SoftFIRE platform. The ***vimInstanceName*** field can contain zero or more of the following options:

- vim-instance-fokus, in order to deploy in Fokus testbed
- vim-instance-er, in order to deploy in Ericsson testbed
- vim-instance-uos, in order to deploy in Surrey testbed
- vim-instance-ads, in order to deploy in ADS testbed
- vim-instance-dt, in order to deploy in Deutsche Telekom testbed

Of course, these testbeds are limited by the availability. The rules of choice will follow the rules defined in the Open Baton documentation [3].

There are no specific constraints for what concerns the networks.

3.2.2. Provision phase

3.2.2.1. Resource reservation and provisioning

Once the experiment CSAR is uploaded, the resource reservation is done. Under the orange and grey table, there will be another table showing the uploaded experiment with a button to deploy it. By clicking it, the experiment previously defined will be deployed and shortly you will be able to access it. The status will then become **ACTIVE**.

3.2.3. Runtime phase

3.2.3.1. Resource monitoring

In case you have asked for a monitoring resource, you will receive an Ip of a Zabbix server where all the VMs are registered. In this Zabbix server you will have full admin rights and you will be able to monitor your resources.

3.2.3.2. Resource control at runtime

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. 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 4 and Section 5.

3.2.4. Closing phase

3.2.4.1. Resource termination

For releasing the resources, you have to put the experiment id in the release resource input field on the right side of the Experimenter page, and click on the button. If the end date is reached before you release the resources, the resources will be automatically released.

The collection of monitoring information is a responsibility of the Experimenter.

4 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 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).

4.1 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.

4.2 Case submission and followup

Once entered in Redmine (Figure 10: Redmine home for SoftFIRE) with the assigned Login/Password jump to the SoftFIRE Project:

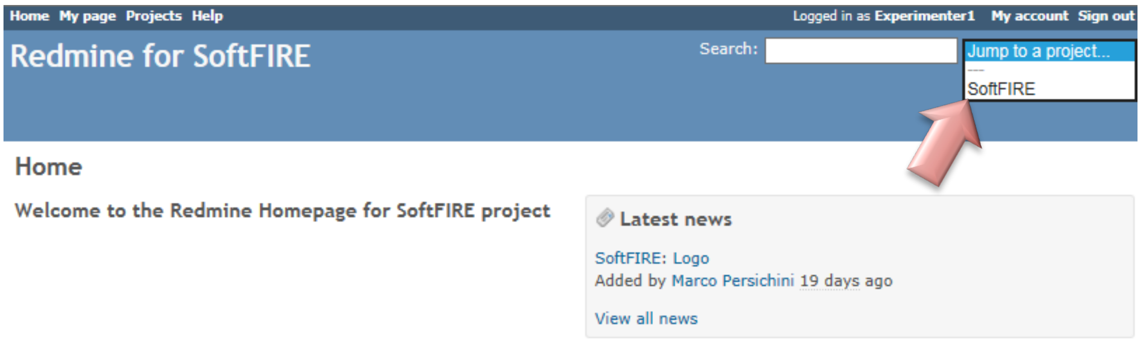

Home My page Projects Help
Logged in as Experimenter1 My account Sign out
Redmine for SoftFIRE
Search:
Jump to a project...
SoftFIRE
Home
Welcome to the Redmine Homepage for SoftFIRE project
Latest news
SoftFIRE: Logo
Added by Marco Persichini 19 days ago
View all news

Figure 10: Redmine home for SoftFIRE

Here you can check the current issues or open a new one (Figure 11: Issues tracking) by means of the following Tabs:

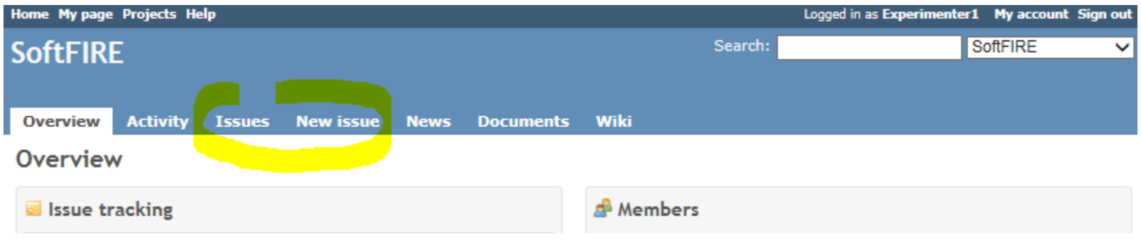


Figure 11: Issues tracking

To open a new case, from "New Issue" tab (Figure 12: Describing a New Issue) you access to the page in the picture below where you have to:

1. Select among two options: Bug or Feature Support
2. Assign a short and significant slogan in the Subject field

3. Write an accurate description, indicating:
    a. STR: Steps to Reproduce
    b. BD: Bug Description
    c. ER: Expected Result
4. Choose the right Category, considering where you need assistance
5. Set a priority based on its Urgency (Normal as default), that will be then revised by the receiver
6. 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…"click on the "Image" Icon and "!...!" will be added in the description
    b. Copy the "File Name.JPG" of the picture between "!...!"

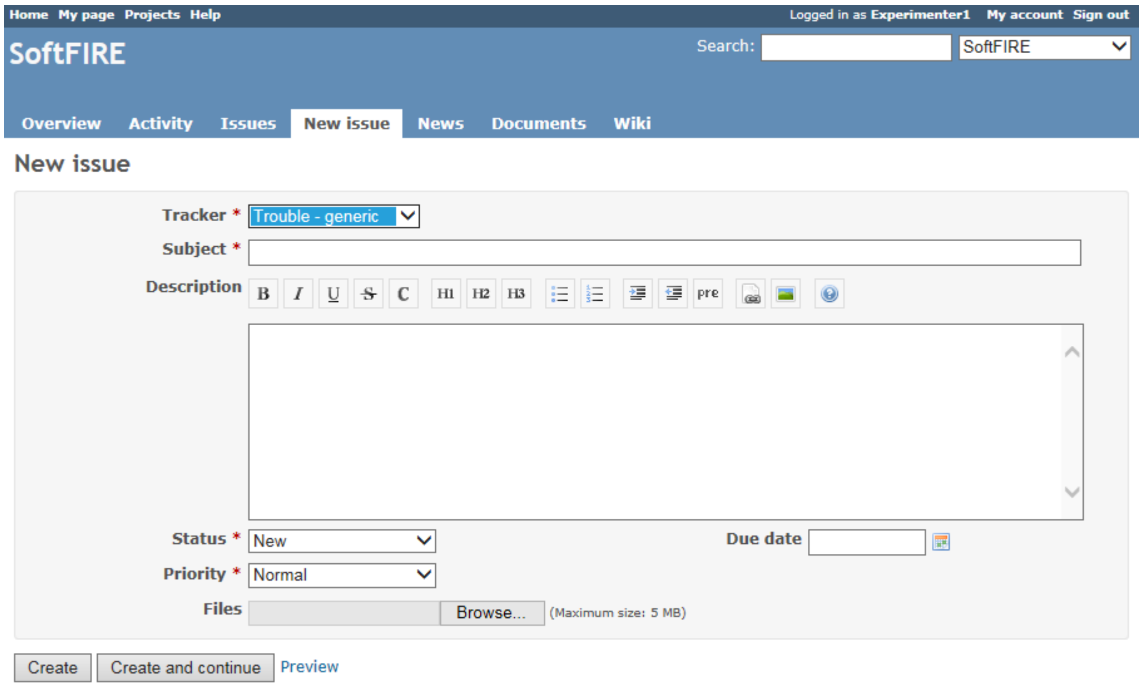


Figure 12: Describing a New Issue

Check if everything is filled in correctly 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

4.3 Workflow

The just opened Cases will be immediately notified to a reference person for each Infrastructure: they will analyse 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 13: Process for tracking issues (*only the Infrastructure Owners can directly Reject or Close the issue*):

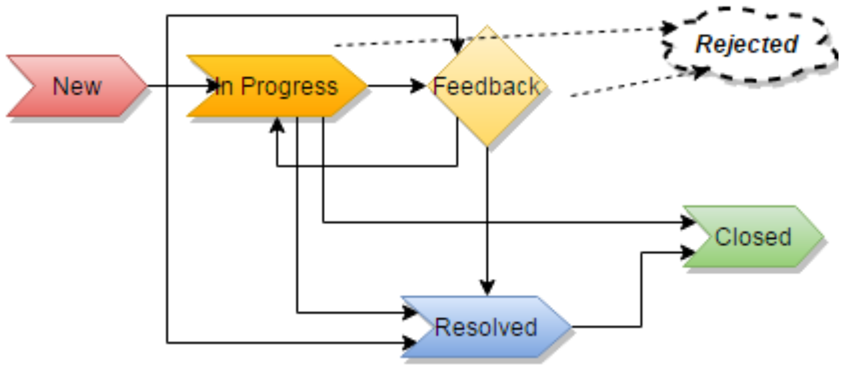


Figure 13: Process for tracking issues

4.4 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/)

4.5 Support Contacts

Partner	Contact
Surrey System	s.vural@surrey.ac.uk
Surrey IT/Enterprise	chris.clark@surrey.ac.uk
Surrey RAN	f.entezami@surrey.ac.uk
TUB	lorenzo.tomasini@tu-berlin.de
FOKUS	bjoern.riemer@fokus.fraunhofer.de
Ericsson	umberto.stravato@ericcson.com
Reply	d.carmignani@reply.it
DT	peter.feil@telekom.de
EIT	roberto.minerva@eitdigital.eu
Assembly Data System	massimiliano.romano@assembly.it

5 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.

5.1 Constraints on the Experimenters Use

The SoftFIRE infrastructure is composed by loosely integrated platform under different administrative domains. In addition, the different platforms 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 result 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 any problem.

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

5.2 Additional support

For more personalized services experimenters are referred to the [5]as well as to the contact persons on the SoftFIRE website.

Updated 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.

6 References

[1] FITEagle, “A Semantic Resource Management Framework,” [Online].

[2] A. j.-b. f. f. t. f. jFED, “jFed,” [Online].

[3] Open Baton, “An open source Network Function Virtualisation Orchestrator (NFVO),” [Online].

[4] OASIS, “TOSCA Simple Profile in YAML Version 1.0,” OASIS, 2016.

[5] SoftFIRE, “SoftFIRE Experimenter Usage Manual Documentation,” [Online]. Available: http://docs.softfire.eu/.

[6] OpenStack, “Open source software for creating private and public clouds,” [Online].

[7] Zabbix, “The Ultimate Enterprise-class Monitoring Platform,” [Online].

[8] ON.Lab, “Bringing openness and innovation to the Internet and Cloud,” [Online].

[9] Open Baton, “TOSCA CSAR on-boarding,” [Online]. Available: https://openbaton.github.io/documentation/tosca-CSAR-onboarding/.

[10] OASIS, “TOSCA Cloud Serivice Archive,” [Online]. Available: http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/csd03/TOSCA-Simple-Profile-YAML-v1.0-csd03.html#_Toc419746172.

[11] SoftFIRE, “Node Type definition,” [Online]. Available: http://docs.softfire.eu/etc/softfire_node_types.yaml.

[12] OpendayLight, “Open Source SDN Platform,” [Online].

[13] ONOS, “Open Network Operating System,” [Online].

[14] FIRE, “ Future Internet Research and Experimentation,” [Online].

[15] FED4Fire, “Federation for Future Internet Research and Experimentation,” [Online].

[16] OMF, “Open Management Framework,” [Online].

[17] ETSI, “Network Functions Virtualisation (NFV);,” ETSI, 2014.

7 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