Software Defined Networks and Network Function Virtualization Testbed within FIRE+

Grant Agreement Nº 687830

D3.1.
KPIs for evaluating and assessing the features of the Testbed

Version: 3.1
Due Date: April 30th 2016
Delivery Date: June 16th 2017
Type: Report (R)
Dissemination Level: PU – Public
Lead partner: Telecom Italia
Authors: All Partners (See List of Contributors bellow)
Internal reviewers: All Partners - PMC

Disclaimer

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	15/06/2016	Roberto Minerva	Telecom Italia	First Draft completed
2.0	30/06/2016	Roberto Minerva	Telecom Italia	Reviews and comments from partners
2.1	01/07/2016	Gerry Foster	University of Surrey	Completion of target values & reformat of tables.
2.2	08/07/2016	Susanne Kuehrer	EIT Digital	Approval from PMC Final editing and submission
3.0	19/05/2017	Roberto Minerva	EIT Digital	Editing and improvement after Review
3.1	11/06/2017	Serdar Vural	U0S	Review and consistency check
3.1	16/06/2017	Susanne Kuehrer	EIT Digital	Final Checks, PMC approval and sumission

Annexes:

№	File Name	Title

Contributors:

Contributor	Partner
Davide Carmignani	Reply
Antonio Casella	Telecom Italia
Gerry Foster	University of Surrey
Susanne Kuehrer	EIT Digital
Roberto Minerva	Telecom Italia / EIT Digital
Daniel Nehls	TUB
Sergio Nuccio	Telecom Italia
Marco Persichini	Ericsson
Björn Riemer	Fokus
Umberto Stravato	Ericsson
Lorenzo Tomasini	TUB
Mario Ullio	Telecom Italia

Keywords: Key Performance Indicators, NFV, SDN, 5G

Executive Summary:

The Network Function Virtualization (NFV) and Software Defined Networking (SDN) approaches are emerging technologies that will change the way networks and communications services will be implemented and delivered to customers. The Industry is currently focusing on the definition of the functionalities and the mechanisms for implementing various NFV/SDN solutions, especially for virtualization of datacenter and network operator systems.

Best practices are under construction by means of initial deployment and early experimentations; i.e. there is no consolidated corpus of guidelines and rules to be followed. This results in a lack of a consolidated set of Key Performance Indicators (KPIs) that can help in assessing the effectiveness of the technologies, and therefore discriminates between features that specific to different platforms.

The current literature on virtualization KPIs to be used in federated virtualization environments is limited, as experimentation and deployment on large-scale settings, especially from a performance measurement perspective, are not yet under analysis of a wide community (see for example (Nexious, 2016)). Furthermore, current approaches leverage existing definitions of Platform KPIs and they frame them into the usual “fault, configuration, accounting, performance, security” (FCAPS) structure. This approach is however questionable because it essentially focuses on the management of IT platforms and, hence, refers mainly to NFV, while disregarding SDN aspects.

In contrast to these existing approaches, the EU SoftFIRE project has taken a different approach: KPI definitions should comprise of the identification of *Platform* KPIs that refer both to NFV and SDN aspects, as well as those that refer to *Programmability* aspects (which constitute a major discriminant for the adoption of an NFV/SDN solution), and those about *Security* aspects, which are essential for running these solutions. In doing so, SoftFIRE has taken a pragmatic methodology; the project analyzes the requirements and the KPIs of the platform by means of Use Cases. This bottom-up approach has allowed us to collect a large number of use-case driven KPIs. In doing so, one major stage was to disentangle any generic KPIs from a specific Use Case and determine whether such KPIs are of general use or not. This has allowed an initial classification of KPIs in NFV/SDN systems, into the following two KPI classes:

- *Platform-related KPI*s: These refer to the measures, functionalities, or qualities that have a general value, and can be reused independently from a specific application;
- *Application-specific KPI*s: These refer to the measures, features, or qualities that have an essential meaning or importance during the execution of a specific application.

In this document, platform-related KPIs have been compiled, forming the bulk of the KPIs of interest for the project; whereas the application-specific KPIs have been described and collected in the User Requirement document. The defined use cases have provided the opportunity to determine an initial set of KPIs related to programmability and security. This is a sufficient starting point for creating a comprehensive set of measures and indicators so as to evaluate important features of NFV/SDN solutions.

Besides this initial set of KPIs (more than 60), those supported by commonly used tools (OpenStack and Zabbix) can also be considered as valid KPIs that can be used on the SoftFIRE platform. The project’s approach is to first identify the relevant ones, to evaluate them, and then to outline how others could be implemented and progressively used, as the project progresses.

Table of Contents

List of Figures

List of Tables

1 Introduction

A federated testbed comprising of different infrastructures is ideal for shedding a light on several technical issues in regards to the recently popular topics of Network Function Virtualization (NFV) and Software Defined Networking (SDN) technologies from a practical point of view. In a federated environment, interoperability issues are commonly encountered, which must be addressed in order to realise federation of multiple testbeds, and to enable efficient use of the testbed by external users, i.e. experimenters. The efficient use of the platform requires unified interfaces and mechanisms in order to allow programmers and experimenters to smoothly access, use, and program its available functionalities. In addition, sufficient level of security is needed in order to protect experimenters and platform providers from improper usage of the infrastructure.

Between now and reaching the goals of the SoftFIRE project, there are in fact evaluation and assessment stages for NFV/SDN technologies from three perspectives: *interoperability*, *programmability,* and *security*. The project analyzes the requirements and the KPIs of the platform by means of Use Cases. This is an effective but challenging approach, which is not trivial to apply to all the use cases in a similar way, due to potential and inevitable differences in the specific use cases. However, it has provided valuable indications on requirements for any further development. This bottom-up approach has also allowed collection of various Key Performance Indicators (KPIs). In doing so, one major stage is to disentangle the project KPIs from a specific use case and determine whether such KPIs are of general use or not. As a result, the project has produced an general classification of KPIs in NFV/SDN systems:

- *Platform-related KPIs:* These refer to the measures, functionalities, or qualities that have a general value, and can hence be reused independently from a specific application,
- *Application-specific KPIs:* These refer to the measures, features, or qualities that have an essential meaning or importance during the execution of a specific application.

Platform-related KPIs form the bulk of KPIs of interest for the project, and hence are presented this document; the others (application-specific) have been described and presented in the User Requirement document.

The wide adoption of software technologies in the network and the move from a functional system architecture to a distributed software architecture requires an assessment and possibly some changes to the way in which network systems are measured, monitored, and managed. This has been evident as a result of the bottom up analysis performed by the project. KPIs have been considered on the basis of their possible effectiveness in measuring and monitoring some platform capabilities, rather than their alignment with existing categories. While for in case of Interoperability KPIs, some of the measures can be considered as a revised FCAPS framework; in case of programmability KPIs, new software approaches and in general an agile development do pose some challenges. It is necessary to note that it is not an objective of this document to propose a new management framework and new categories, but it is evident from the initial analysis conducted in the use case definition that a software-based architecture may change the classical FCAPS based approach. For instance, NFV orchestration is a powerful means that has built in Configuration, Fault, Accounting and Performance features. Its management functions can be considered as real-time or at least a very dynamic way of allocating and configuring resources with strong intertwining with Fault and Recovery issues (e.g., starting a new instance of a faulty virtual machine).

Security in an NFV/SDN environment requires its own new framework. This is evident from the objectives of the SDN/NFV itself: programming the network is strongly coupled with security. It is not a wise business approach to allow access to APIs and interfaces that have not been designed and thought within a security framework. Unfortunately, the current definitions do not emphasize this important relationship and often security consideration are disregarded

NFV systems tend to manage storage, computing, and the internal communication capabilities (i.e. networking features that are needed in order to allow communication between components of the platform and its applications. The SoftFIRE infrastructure considers not only NFV, but also the availability of external resources, which makes the system a blend of NFV and SDN, and inevitably add complexities. These resources may be SDN nodes with their standard interfaces, and their system characteristics to be monitored. These extended functionalities and resources introduce a need for different quantities to be monitored (KPIs) and also functions to be controlled.

Figure 1 depicts a high-level architecture of the SoftFIRE infrastructure.

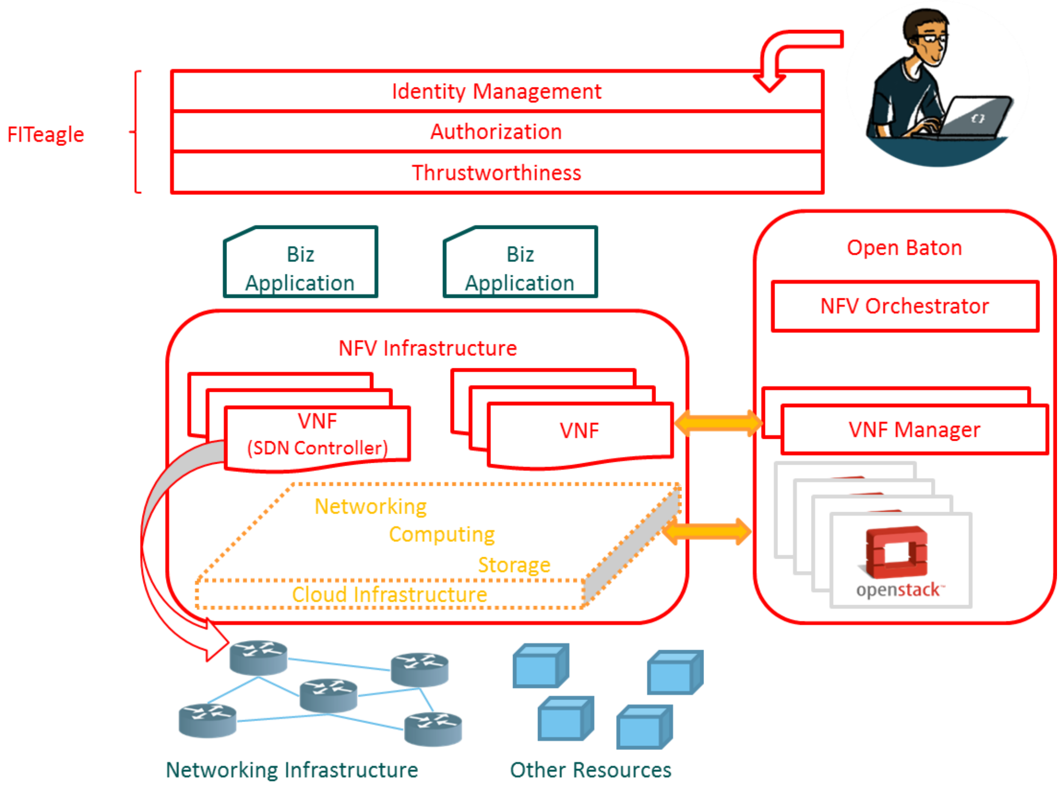

Figure 1: High level SoftFIRE architecture

A number of functional blocks need to be monitored, which can be listed as follows:

- An access module that deals with identification and authorization of users and experimenters, and grants them access credentials to use the platform. Therefore, its functionalities are very bound to security,
- An orchestration module dealing with the governance of resource allocation,
- An NFV infrastructure supporting the execution of the virtual machines and some existing services,
- A set of applications running on top of the infrastructure

- A set of resources that can be used by applications in order to execute special tasks or to control a network (i.e., SDN switches).

Since the SoftFIRE infrastructure is a combination of several testbeds. As such, it has integrated the different functionalities offered by its component testbeds into a single infrastructure. This has inevitably caused a number of interoperability issues, which have been considered. In fact, the realm of Interoperability comprises of different kinds of KPIs: (i) those that fully apply to the computing infrastructure (with particular reference to the NFV system), (ii) those that apply to the interaction between virtual machines and resources, and (iii) those that refer to the internal working of business applications. Figure 2 depicts these three groups of possible KPIs groups:

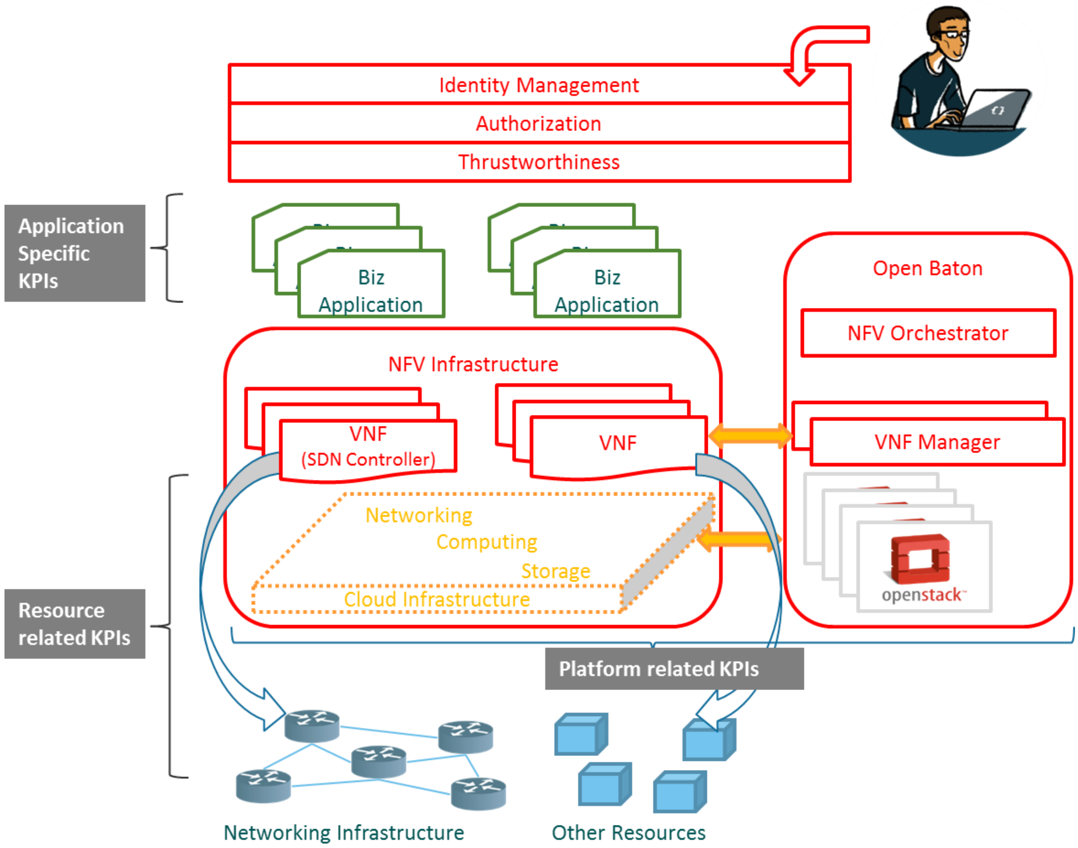

Figure 2: Different KPIs types in SoftFIRE

Further classification of the interoperability KPIs are presented in Section 3.

While NFV has received some attention with respect to KPIs and monitoring [(Nexious, 2016) and (ETSI, 2014) the integration of NFV and SDN has not yet been fully studied. In this regard, the approach of the SoftFIRE project is to focus on platform and resource related KPIs, while applications specific KPIs are outside the scope.

With respect to programmability, KPI definitions are at an initial stage, mainly focusing on achieving initial measures of the easiness of accessing and programming the virtualization platforms. Other measurements related to the quality and completeness of the functions offered by the platform are still to be defined.

The experimentation phases to be carried out by the project will help to determine new parameters and features to monitor, as well as measuring the strengths and weaknesses of different platforms. Such evaluation of the performance of existing virtualization platforms will provide strong guidelines for later attempts on deploying integrated NFV/SDN platforms that are to be extensively programmed. Feedback from experimenters during the different waves of experimenter use of the SoftFIRE platform will also help define new KPIs related with monitoring various previously not listed features and properties.

Security is no doubt an essential feature that can determine the success of the deployment of NFV/SDN systems, since such systems are not mature enough with sufficiently integrated security measures in place. As a first step, the project has considered an initial set of KPIs that measure the possibility of damage from approved users of the platform, e.g. access to the virtual machine instances of created by other users. In this case, the subsequent waves of experimentations will extend the number of security KPIs. However, it is important to stress here that security KPIs will play a relevant role in the consolidation of this technology and that only a platform that can truly offer a high level of security and monitoring will receive attention from businesses and users of virtualization platforms.

2 Existing Measures in the TestBeds

The individual component testbeds in SoftFIRE do not employ massive monitoring capabilities. Furthermore, some of the testbeds do not collect any KPI measurements or the like; these testbeds are operated as a working tool to provide for Proof of Concept (PoC) demonstrations or in-house research and experimentation. In order to be able to monitor the necessary KPIs for SoftFIRE, the monitoring capabilities provided by OpenStack or tools like Zabbix are used.

2.1 The Zabbix monitoring tool

Zabbix can provide more than 100 different *metrics* (Zabbix refers to KPIs as *metrics*), part of which are monitored by Zabbix by default. It is possible to define and create new customized metrics at *agent* level in Zabbix, which will then need to be also configured at the Zabbix server side, as Zabbix agents cannot push new metric definitions to the server. All collected metric values, can be monitored via the Zabbix portal.

The Zabbix monitoring solution provides its own template that includes many metrics that have default configurations. Users can reconfigure these metrics as needed. This can be manually done either through Zabbix API or more easily through the Zabbix GUI. Furthermore, many attributes of a metric can be reconfigured, e.g. update interval (how often the metric is measured), period of time to keep a metric value in history, etc.

Users can also create their own templates, including the required metrics with relevant specific configurations. These templates can be saved and used at any time by any monitored machine. By default, a number of items are taken into consideration, which are related to the computing infrastructure and to the configuration of virtual machines.

2.2 The KPIs monitored via Zabbix

The operating system that is typically used to host the virtual infrastructure manager OpenStack is Linux. The most typical KPIs monitored on Linux server machines via Zabbix can be listed as follows:

- *Average CPU usage* – the average single CPU usage (time average of single CPU core) and the average aggregate CPU usage (time average of the aggregate over all CPU cores),
- *Maximum CPU usage* – the maximum CPU usage observed; can be monitored for single and aggregate cases,
- *Average drive space* – the amount of storage used by the system and the running applications,
- *Maximum drive space* – the maximum amount of disk space allocated (although not a KPI as such, this indicates how much is space is available at a time, and enables to monitor any changes that might have diverse results)
- *Minimum free memory* – the minimum available system memory,
- *Maximum number of virtual machine* – it indicates the limits in terms of possible instances of VMs on the platform (although not a KPI as such, this indicates how many VMs are allowed at a time, and enables to monitor any changes that might have diverse results)

- *Average number of virtual machines* –the average number of running VMs on a specific testbed, which indicates a count-based indication of the system's virtualization resources.

More fine-grained Zabbix metrics can be listed as follows, some of which can be used as performance KPIs:

CPU

- CPU idle time
- CPU interrupt time
- CPU iowait time (IO wait)
- CPU nice time
- CPU softirq time (Soft IRQ)
- CPU steal time
- CPU system time
- CPU user time

Filesystem

Zabbix will automatically discover all the file systems mounted on servers (either virtual or physical). It is at the discretion of the infrastructure owner to decide which ones are relevant to be monitored in their virtualization environment.

- Free disk space on a specific file system
- Free inodes on a specific file system (inode: a filesystem object, such as a file or directory)
- Total disk space on a specific file system
- Used disk space on a specific file system

ICMP

- ICMP loss
- ICMP ping
- ICMP response time

Memory

- Available memory
- Free swap space
- Total memory
- Total swap space

Network Interface

Zabbix automatically discovers all network interfaces (either virtual or physical). It is at the discretion of the infrastructure owner to decide which ones are relevant in their virtualization environments.

- Incoming network traffic on network interfaces
- Outgoing network traffic on network interfaces

Operating System

- Host boot time

- Host local time
- Host name
- Maximum number of open files
- Maximum number of running processes
- System information
- System uptime

Performance

- Context switches per second
- Interrupts per second
- Processor load (1 min average per core)
- Processor load (5 min average per core)
- Processor load (15 min average per core)

Processes

- Number of processes
- Number of running processes

Security

- Checksum of the Linux file /etc/passwrd
- Number of logged in users

Zabbix Agent

- Agent ping
- Host name of zabbix_agentd running (zabbix_agentd : daemon process of Zabbix)
- Version of zabbix_agentd running

3 Selection and Definition of Initial KPIs

As previously mentioned, the SoftFIRE project has collected a number of KPIs specific to its federated testbed framework[1]. From a platform deployment perspective, the following KPI groups are monitored:

- Infrastructure (including resources with special focus on SDN related ones) (Table 1),
- Platform services (Table 2)
- Self-Organized Networking (SON) features (Table 3). Note that this is essential for next generation mobile networking environments, i.e. 5G.

The other two KPI groups are as follows:

- Programmability (Table 4)
- Security (Table 5).

Before presenting the specific KPIs, the consideration and decision approaches in KPI definition and selection are first described next. Hence, the initial KPIs, especially regarding the Programmability KPIs, are defined according to the following initial consideration points.

3.1 Phases of virtual application development

The definition and selection of KPIs considers the following three major development phases of a virtual software application:

- *Design phase:* In this phase, those tools and documentation that make the design phase easier should be used.
- *Development phase:* In this phase, the tools and documentation as well as the actual services, APIs, and libraries offered should be carefully selected. During this phase a set of KPIs related to development efficiency and ease should be considered and defined.
- *Execution phase:* In this phase, the number of services, functions, and APIs offered by the platform should be measured in terms of their usage.

The KPIs that refer to, measure, and evaluate the development process of applications that will be deployed on a platform are out of scope in SoftFIRE (and similar platforms). In fact, different development models can be applied (essentially Agile ones with plenty of supporting methodologies and tools (Fuggetta)), but little attention has been so far devoted to programmability of platforms. SoftFIRE focuses on deployment and execution of virtual services on its virtualised federated testbed environment.

[1] In addition to these KPIs, the Use Case exercise carried out by the project also helped to find some application specific KPIs. These KPIs are not discussed here because they are not of general use for the platform. They may eventually be presented if there is an interest from experimenters or other parties. These KPIs measure the specific behavior of services and applications that run on top of platform.

3.2 Software Metrics and Programmability

The programmability and measurement metrics of a virtualization platform has been left essentially to to the provider of the platforms; such metrics are typically not exposed or generally disclosed. In addition, these metrics are sometimes used in the context of service engineering development models. Therefore, in order to lay the foundation for measuring the "programmability of a platform", an initial definition is needed.

Generally speaking, a software metric is a function measuring the degree to which a software system or process possesses some property. In this sense, project SoftFIRE defines platform programmability according to the following features:

a. The additional software development needed to integrate the application/service on the platform (in other terms the additional development needed to adapt the envisaged properties to run on the chosen platform),
b. The number of services and interfaces (APIs) and libraries that a programmer can easily access and that are directly offered by the platform,
c. The set of tools and mechanisms that the platform offers, which are useful for software development and management,
d. The set of tools that allow rapid and efficient testing and deployment of applications and services on the platform.

With this empirical definition of "programmability" in mind, it is intuitive that the feature a., i.e. additional software development and platform adaptation, should be minimized. While Feature b. above, i.e. offered services, should be maximized (the more services and interfaces, the the higher the programmability). Feature c., i.e. supporting services, should be somehow "optimized" in the sense that the platform should provide the "right" tools and mechanisms without overwhelming or superimposing certain tools and methodologies to programmers. Feature d., i.e. the debugging capability, is intertwined with the chosen development environment and as such it should be supported by the specific tools chosen by developers. The platform should provide easy mechanisms to integrate and support these tools.

These contexts (and features) help identify the metrics and KPIs. However, as mentioned before, the software development methodologies and related KPIs used to develop the application are to be kept outside of the platform ones and are at the care of the development teams that will use the SoftFIRE Platform.

3.3 Deployment vs Development: Efficiency in virtual software deployment in SoftFIRE

As a rule of thumb, the effort to deploy an application software on the SoftFIRE platform should be a fraction of the total development effort of the experimenters. In order to be effective, the platform should ensure that the development time is only a fraction of the total time of the platform adaptation:

total time for application development >> actual time of its porting on SoftFIRE.

This property should also hold for the estimation of the needed time during the design phase and any subsequent performance checks. To have a quantitative measure of the effort made in developing and then deploying an application, we can define *Magnitude of Relative Error (MRE)*, formally represented as follows:

$$MRE = |\text{actual effort} - \text{estimated effort}| / \text{actual effort}.$$

Besides the total development time, it is also desired that any amount of error in estimated time of development/deployment relative to the actual amount of time spent towards such effort must also be minimised. In other words, it is the desire of SoftFIRE to have much less deviation from its estimated software deployment time on the platform, as compared to typical relative errors made in the actual software development time estimation. With the definition of MRE as above, we then should have the following:

$$MRE_{(app)} >> MRE_{(dep)},$$

where $MRE_{(app)}$ is the error in application development time, and $MRE_{(dep)}$ is the error in deployment time. During the experimentations and the usage of the platform these values should be collected and compared; this requires help from experimenters during their software development and deployment stages.

4 The List of identified KPIs

The KPIs represented in the following tables are described in terms of the following attributes:

- KPI Name,
- a short description,
- measure units,
- KPI type,
- applicability to LTE (existing 4G mobile networks),
- Success measure or range,
- scope in the lifetime cycle of experimentation,
- objects to be measured.

The Infrastructure KPIs are represented in Table 1.

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
1	Tsetup(vm,intra)	Time to negotiate, allocate and instantiate a virtual machine on same test-bed.	ms	Infrastructure	<1 minute	No worse than benchmark on federated TB	Run-Time	VM	Intra
2	Tsend(vm,intra)	Time to send from VM instance (a) to receive at VM (b), where the VM(a) and VM(b) are on the same test-bed.	ms	Infrastructure	BM(01) see below	<(*1) +10%	Run-Time	VM	Intra
3	Tsetup(sdn,intra)	Time to negotiate, allocate and instantiate a single SDN Switch within the same test-bed, from Orchestration level	ms	Infrastructure	N/A	<1 minute	Run-Time	SDN-Sx	Intra
4	Tsetup(flow,intra)	Mean time to negotiate, allocate and instantiate each flow within the same test-bed	ms	Infrastructure	N/A	<1s	Run-Time	SDN-Flow	Intra
5	Tsetup(sdn-cn, intra)	Time to negotiate, allocate and instantiate all SDN switches required for a minimal base Core Network slice for scope CN01 (see below) within the same testbed.	ms	Infrastructure	N/A	<10 minutes	Run-Time	SDN(01),CN(01)	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
6	Tsetup(sdn-nwk, intra)	Time to negotiate, allocate and instantiate all SDN switches required for a minimal base Core Network slice and connect to at least one Access Technology for scope CN01 (see below) within the same testbed.	ms	Infrastructure	N/A	<25 minutes	Run-Time	SDN(01), Nwk(01)	Intra
7	Tsetup(nfv-cn, intra)	Time to negotiate, allocate and instantiate an NFV, CN on the same testbed. See Staging(01)	ms	Infrastructure	4 hours+	<10 Minutes	Run-Time	NFV, CN(01)	Intra
8	Tsetup(nfv-nwk, intra)	Time to negotiate, allocate and instantiate an NFV, Nwk on the same testbed. See Staging(01)	ms	Infrastructure	4 hours+	<25 Minutes	Run-Time	NFV, Nwk(01)	Intra
9	Tsetup(vm,inter)	Time to negotiate, allocate and instantiate a virtual machine from Test-bed(A) on Test-bed(B).	ms	Infrastructure	<1 minute	No worse than benchmark on federated TB	Run-Time	VM	Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
10	Tsend(vm,inter)	Time to send from VM instance (a) to receive at VM (b), where the VM(a) and VM(b) are on different test-beds.	ms	Infrastructure	BM(01) see below	<(*1) +15%	Run-Time	VM	Inter
11	Tsetup(sdn,inter)	Time to negotiate, allocate and instantiate a single SDN Switch from Test-bed(A) on Test-bed(B).	ms	Infrastructure	N/A	<1.5 minute	Run-Time	SDN-Sx	Inter
12	Tsetup(flow,inter)	Mean time to negotiate, allocate and instantiate a flow from Test-bed(A) on Test-bed(B).	ms	Infrastructure	N/A	<1.5s	Run-Time	SDN-Flow	Inter
13	Tsetup(sdn-cn, inter)	Time to negotiate, allocate and instantiate all SDN switches required for a minimal base Core Network slice for scope CN01 (see below), from Test-bed(A) on Test-bed(B).	ms	Infrastructure	N/A	<15 minutes	Run-Time	SDN(01),CN(01)	Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
14	Tsetup(sdn-nwk, inter)	Time to negotiate, allocate and instantiate all SDN switches required for a minimal base Core Network slice and connect to at least one Access Technology for scope CN01 (see below) from Test-bed(A) on Test-bed(B).	ms	Infrastructure	N/A	<35 minutes	Run-Time	SDN(01), Nwk(01)	Inter
15	Tsetup(nfv-cn, inter)	Time to negotiate, allocate and instantiate an NFV, CN from Test-bed(A) on Test-bed(B). See Staging(01)	ms	Infrastructure	4 hours+	<15 Minutes	Run-Time	NFV, CN(01),	Inter
16	Tsetup(nfv-nwk, inter)	Time to negotiate, allocate and instantiate an NFV, Nwk from Test-bed(A) on Test-bed(B). See Staging(01)	ms	Infrastructure	4 hours+	<35 Minutes	Run-Time	NFV, Nwk(01),	Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
17	Tsetup(nfv-nwk, split-nwk)	Time to negotiate, allocate and instantiate an NFV, CN from Test-bed(A) where the CN part is on Test-bed(B) and the Access Network is on Testbed(A), See Staging(01)	ms	Infrastructure	4 hours+	<20 Minutes	Run-Time	NFV, Nwk(02),	Inter
18	Tsetup(nfv-nwk, split-cn)	Time to negotiate, allocate and instantiate an NFV, CN from Test-bed(A) where part of the CN is on Testbed(A) and another part is on Test-bed(B). The Access Network is on Testbed(A) See Staging(01)	ms	Infrastructure	4 hours+	<20 Minutes	Run-Time	NFV, Nwk(03),	Inter
19	Tsetup(nfv-nwk, split-cn&nwk)	Time to negotiate, allocate and instantiate an NFV, CN from Test-bed(A) where part of the CN is on Testbed(A) and another part is on Test-bed(B). The Access Network is on Testbed(B) See Staging(01)	ms	Infrastructure	4 hours+	<40 Minutes	Run-Time	NFV, Nwk(04),	Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
20	TreconfigN(nfv-cn, intra)	Reconfiguring time following a resource crash of 1 to N x CN NFV resource controlled from the same test-bed Where: - N is the total number of Nodes in the Slice	ms	Infrastructure	4 hours+	15 Minutes	Run-Time	NFV, Nwk(01),	Intra
21	TreconfigN(nfv-cn, inter)	Reconfiguring time following a resource crash of 1 to N x CN NFV resource in Testbed(B) which is controlled from controlled from Test-bed(A)	ms	Infrastructure	4 hours+	<20 Minutes	Run-Time	NFV, Nwk(01),	Inter
22	TSetLocalShadow (ResourceId, TestbedId, replicationValue, policyId)	Time to define and set up Resource Shadowing in the same testbed, i.e. to setup different VM to run together in shadow mode according to a specific policy To measure the possibility and the level to offer further resilience in the system	ms	Infrastructure	N/A	< 1 sec (no more delay during session than for setup)	Run-Time	vNF, SDN, Whole TB	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
23	TSetDistrShadow (resourceId, TestbedIdList, replicationValueList , PolicyId)	Time to define and Setup Shadowing in different testbeds according to a specific policy To measure the possibility and the level to offer further resilience in the system	ms	Infrastructure	N/A	< 1 sec (no more delay during session than for setup)	Run-Time	vNF, SDN, Whole TB	Inter
23	Num(vm/test-bed, intra)	Max number of VM resources able to be allocated per test-bed	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	VM	Intra
24	Num(sdn-sx/test-bed,intra)	Max number of SDN Sx resources able to be allocated per test-bed	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	SDN-Sx,Nwk(01)	Intra
25	Num(sdn-flows/sx,intra)	Max number of SDN flow resources able to be allocated per SDN Sx.	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	SDN-Flow	Intra
26	Num(nfv-mme/test-bed,intra)	Max number of NFV(MME) resources able to be allocated per test-bed.	#	Infrastructure	1-10/ PLMN	Dependent upon component testbed	Run-Time	NFV(MME)	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
27	Num(nfv-hss/test-bed,intra)	Max number of NFV(HSS) resources able to be allocated per test-bed.	#	Infrastructure	1-5/ PLMN	Dependent upon component testbed	Run-Time	NFV(HSS)	Intra
28	Num(nfv-sgw/test-bed,intra)	Max number of NFV(SGW) resources able to be allocated per test-bed.	#	Infrastructure	1-10/ PLMN	Dependent upon component testbed	Run-Time	NFVSGW)	Intra
29	Num(nfv-pgw/test-bed,intra)	Max number of NFV(PGW) resources able to be allocated per test-bed.	#	Infrastructure	1-5/ PLMN	Dependent upon component testbed	Run-Time	NFV(PGW)	Intra
30	Num(nfv-CC/test-bed,intra)	Max number of NFV(5G-CC) resources able to be allocated per test-bed.	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	NFV(5G-CC)	Intra
31	Num(nfv-CM/test-bed,intra)	Max number of NFV(5G-CM) resources able to be allocated per test-bed.	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	NFV(5G-CM)	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
32	Num(nfv-PPE/test-bed,intra)	Max number of NFV(5G-PPE) resources able to be allocated per test-bed.	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	NFV(5G-PPE)	Intra
33	Num(Cells/test-bed,intra)	Max number of LTE-A Cell resources able to be allocated per test-bed.	#	Infrastructure	50-200,000/ PLMN	Dependent upon component testbed	Run-Time	LTE-A_Cell	Intra
34	Num(AP/test-bed,intra)	Max number of Wi-Fi resources able to be allocated per test-bed.	#	Infrastructure	50-200,000/ PLMN	Dependent upon component testbed	Run-Time	Wi-Fi_AP	Intra
35	Num(UE/test-bed,intra)	Max number of UE resources able to be provisioned from SoftFIRE per test-bed.	#	Infrastructure	0.5-200M/ PLMN	Dependent upon component testbed	Run-Time	UE	Intra
36	NumAlloc(vm/test-bed,inter)	Max number of VM resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	VM	Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
37	NumAlloc(sdn-sx/test-bed,inter)	Max number of SDN Sx resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	SDN Sx,Nwk(01)	Inter
38	NumAlloc(sdn-flows/sx,inter)	Max number of SDN flow resources allocated per SDN Sx from Testbed(A) on Testbed(B).	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	SDN-Flows	Inter
39	NumAlloc(nfv-mme/test-bed,inter)	Max number of NFV(MME) resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	1-10/ PLMN	Dependent upon component testbed	Run-Time	NFV(MME)	Inter
40	NumAlloc(nfv-hss/test-bed,inter)	Max number of NFV(HSS) resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	1-5/ PLMN	Dependent upon component testbed	Run-Time	NFV(HSS)	Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
41	NumAlloc(nfv-sgw/test-bed,inter)	Max number of NFV(SGW) resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	1-10/ PLMN	Dependent upon component testbed	Run-Time	NFV(SGW)	Inter
42	NumAlloc(nfv-pgw/test-bed,inter)	Max number of NFV(PGW) resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	1-5/ PLMN	Dependent upon component testbed	Run-Time	NFV(PGW)	Inter
43	NumAlloc(nfv-CC/test-bed,inter)	Max number of NFV(5G-CC) resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	NFV(5G-CC)	Inter
44	NumAlloc(nfv-CM/test-bed,inter)	Max number of NFV(5G-CM) resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	NFV(5G-CM)	Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
45	NumAlloc(nfv-PPE/test-bed,inter)	Max number of NFV(5G-PPE) resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	N/A	Dependent upon component testbed	Run-Time	NFV(5G-PPE)	Inter
46	NumAlloc(Cells/test-bed,inter)	Max number of LTE-A Cell resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	50-200,000	Dependent upon component testbed	Run-Time	LTE-A-Cell	Inter
47	NumAlloc(AP/test-bed,inter)	Max number of Wi-Fi resources allocated per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	50-200,000	Dependent upon component testbed	Run-Time	Wi-Fi_AP	Inter
48	NumAlloc(UE/test-bed,inter)	Max number of UE resources able to be provisioned from SoftFIRE per test-bed from Testbed(A) on Testbed(B).	#	Infrastructure	0.5-200M/PLMN	Dependent upon component testbed	Run-Time	UE	Inter

Table 1: Infrastructure KPIs

The reference notes for Table 1 are as follows:

Reference	Description
BM(01)	NAS EPS(Bearer Setup) EPC message from MME to SGW on VM-Ware VMs
CN(01)	An NFV slice comprising at least one of each of the EPC Core Network elements including: HSS, MME, SGW, PGW or combined portions thereof.
Nwk(01)	An NFV slice comprising one CN(01) and connectivity to at least one Access Technology.
Nwk(02)	An NFV slice comprising one CN(01) on Test-bed(A) and the Access Network on Test-bed(B).
Nwk(03)	An NFV slice comprising one part of CN(01) is on Test-bed(A) and another is on Test-bed(B) the Access Network is on Test-bed(A) and connectivity to at least one Access Technology is provided on Test-bed(A).
Nwk(03)	An NFV slice comprising one part of CN(01) is on Test-bed(A) and another is on Test-bed(B) the Access Network is on Test-bed(A) and connectivity to at least one Access Technology is provided on Test-bed(B).
SDN(01)	SDN comprising 3 or more switches with at least 8 ports per switch
Staging(01) (*1)	Assumes that required VMs and SDN resources have already been instantiated and are available (as other KPIs capture these parts of performance separately) Within the time of one MTU size packet between the instances at the layer 2 level line speed between them.

The platform services KPIs are represented in Table 2.

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
62	Throughput(X)	Testbed traffic data volume throughput at node, or NFV "X" in unit time.	Gbytes/s	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service @ named Node or @ named interface	Intra
63	RbMax(X)	Maximum testbed bit rate at node, or NFV "X" in unit time.	Mbit/s	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service @ named Node or @ named interface	Intra
64	RbMean(X)	Mean testbed bit rate at node, or NFV "X" in unit time.	Mbit/s	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service @ named Node or @ named interface	Intra
65	Latency(A-B)	Mean IP packet delay for standard MTU length between reference points X and Y	ms	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service: - between named nodes	Intra
66	Ttrans(X)	Time for a defined transaction "X" to complete successfully Eg a 2kbyte web page or 15 second	ms	Service	2kbyte page < 1s Google welcome page request from mobile	Comparable or better to LTE benchmark	Run-Time	Service - Transaction between 2 entities defined as X.	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
		UHD Video file with defined format.			downloaded to mobile				

Table 2: Platform Services KPIs

In Table 3, KPIs related to Self-Organized Network (SON) parameters are represented. They represent further measures to be integrated in the platform.

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
49	NLB(UL)	Uplink Load balancing efficiency(Neta) improvement factor. As ratio of data volume Gbytes per network across all PGW with and without load balancing activated, as percentage	%	SON	N/A	Target 25% measurable improvement	Run-Time	PLMN	Intra
50	NLB(DL)	Downlink Load balancing efficiency (Neta)improvement factor As ratio of data volume Gbytes per network across all PGW with and without load balancing activated, as percentage	%	SON	N/A	Target 25% measurable improvement	Run-Time	PLMN	Intra

Table 3: Self-Organized Network KPIs

The Programmability KPIs (represented in Table 4) try to capture the easiness of the implementation of applications and the related process. They are initial KPIs to be improved, extended and consolidated during the experimentation waves.

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
51	Tadd(nfv-new)	Typical configuration and process time required to define a new resource in the system - Assuming new software NE instance is available as static compiled code version with config script. - All test-beds	mins	Programmability	N/A	**	Setup	NFV(New)	Intra
52	Rating(api)	Availability and effectiveness of management APIs of the federated testbed Assessment by mean score per API of interface ease of use via ranking questionnaire from experimenters	%	Programmability	N/A	**	Setup	NFV(New)	Intra
53	Rating(api)	Effectiveness and simplicity of access to platform for programming purposes	%	Programmability	N/A	**	Setup	NFV(New)	Intra
54	Rating(api)	Programmer evaluation of federated functionalities and their accessibility	%	Programmability	N/A	**	Setup	SoftFIRE	Inter
55	TDescribeLocalShadow (resourceId, TestbedId, UserId, policyId)	Typical configuration and process time required to define a new shadow mechanism within a single	mins	Programmability	N/A	**	Setup	NFV(New)	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
		testbed for a specific experimenter (User)							
56	TDescribeDistrShadow (resourceId, TestbedIdList, UserId, policyId)	Typical configuration and process time required to define a new shadow mechanism within different nodes of the testbed for a specific experimenter (User)	mins	Programmability	N/A	**	Setup	NFV(New)	Intra
57	MRE(plat)	Magnitude of Relative Error in software evaluation effort on the platform	person month	Programmability	N/A	value < of 10% of MRU(app)	Design and Development	SoftFIRE	Intra
58	Rating(documentation)	Programmer evaluation of received documentation	rate from 0 to 10	Programmability	N/A	> 6	Design and Development	SoftFIRE	Intra
59	TLearningCycle(Programmer)	Average Time for the learning curve of the platform of an expert programmer	days	Programmability	N/A	value < 90	Development and execution	SoftFIRE	Intra
60	Num(DevTools)	Number of Developping tools integrated in the platform	# of unit	Programmability	N/A	>2	Design and Development	SoftFIRE	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
61	Rating(DevTools)	Programmer evaluation of the available development tools	rating from 0 to 10	Programmability	N/A	>6	Design and Development	SoftFIRE	Intra
62	Num(Serivces)	Number of services offered to the programmers	# of units	Programmability	N/A	>3	Design and development	SoftFIRE	Intra
63	Rating(reuse of services)	Programmer evaluation about the reusability and importance of offered services	rating from 0 to 10	Programmability	N/A	>6	Design and Development	SoftFIRE	Intra
64	Num(APIs)	Number of APIs offered by the platform	# of units	Programmability	N/A	> 4	Design and Development	SoftFIRE	Intra
65	Rating(APIs)	Programmer evaluation of number and quality of offered APIs	rating from 0 to 10	Programmability	N/A	>7	Design and Development	SoftFIRE	Intra
66	NumCall(SystemAPIs)	Number of calls to SoftFIRE basic APIs	# of units / day	Programmability	N/A		Execution	SoftFIRE	Intra
67	NumFaultyCall(SystemAPIs)	Number of Call to system APIs that return and error	# of units / day	Programmability	N/A		Execution	SoftFIRE	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
68	AverageRate(FaultyAPIsCall)	this is = NumFaultyCall(System/APIs) * 100/NumCall(SystemAPIs)	percentage	Programmability	N/A	value < 1%	Execution	SoftFIRE	Intra
69	NumCall(PlatService)	Number of Call to Platform Services	# of units / day	Programmability	N/A		Execution	SoftFIRE	Intra
70	NumFaultyCall(PlatService)	Number of Faulty Calls to Platform Services	# of units / day	Programmability	N/A		Execution	SoftFIRE	Intra
71	AverageRate(FaultyPlatService)	This is NumFaultyCall(PlatService) * 100/ NumCall(PlatService)	percentage	Programmability	N/A	value <3%	Execution	SoftFIRE	Intra
72	NumCall(Service)	Number of Calls to a specific Service running on the platform or component	# of units / day	Programmability	N/A		Execution	SoftFIRE	Intra
73	NumFaultyCall(Service)	Number of Faulty Calls to a specific Service running on the platform or component	# of units / day	Programmability	N/A		Execution	SoftFIRE	Intra
74	AverageRate(FaultyPlatService)	This is NumFaultyCall(Service) * 100/ NumCall(Service)	percentage	Programmability	N/A	value < 5%	Execution	SoftFIRE	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
75	Tbug(Solved)	Time elapsed between bug report from the programmer to the solution of the problem	hours	Programmability	N/A	value < 72	Execution	SoftFIRE	Intra
76	NumComponents(InitialDepl)	Initial Number of components deployed for starting the offering of the application with a global faulty rate < 5%	# of units	Programmability	N/A		Execution	SoftFIRE	Intra
77	NumComponents(CurrentDepl)	Current Number of components deployed for starting the offering of the application with a global faulty rate < 5%	# of units	Programmability	N/A		Execution	SoftFIRE	Intra
78	ScalabilityFactor(CurrentDepl)	This is = NumComponents(CurrentDepl) / NumComponents(InitialDepl)	fact of scale	Programmability		value > 3	Execution	SoftFIRE	Intra

Table 4: Programmability KPIs

The reference notes for Table 4 are as follows:

Reference	Description
**	To be set per experiment.

Finally the initial set of KPIs for security evaluation are represented in Table 5.

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
57	SecurityRanking	Number of top 0 to 100 noted security vulnerabilities per Test-bed and Ranking of: Impact as:High, Medium or Low High = persistent loss of service grade for test-bed (3) Medium = reduced service (2) Low = occasional reduction in service (1) Ease of Attack: H, M L High = easy, within <=10 mins (3) Low = takes 1-2 days to setup (1) = issue sum(Impact + Ease)/6	%	Security	N/A	95	Run-Time	Testbed	Intra
58	SecurityLoss(vnf's)	Ability to damage other Users instances As Number of critical security loss vulnerabilities identified that can damage a VNF so at to render as less than 50% efficient. = sum(vulnerability x #instances allocated in testbed)/total VNF in Test-bed	%	Security	N/A	98	Run-Time	Testbed	Intra
59	SecurityPrivacy(Volume)	Ability to access user data. As sum of # bytes of user data accessible via all threats x number of users in test-bed.	Gbytes	Security	N/A	98	Run-Time	Testbed	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
60	SecurityPrivacy(Volume)	Ability to access user data. As sum of # bytes of user data accessible via all threats x number of users in test-bed versus total Gbytes of user data on the test-bed	%	Security	N/A	98	Run-Time	Testbed	Intra
61	SecurityApp(X)	Ability to interfere with user application “X” to degree where application is unusable or at least 50% slower than normal. Expressed as units of #Users affected	#Users	Security	N/A	98	Run-Time	Testbed	Intra
62	Throughput(X)	Testbed traffic data volume throughput at node, or NFV “X” in unit time.	Gbytes/s	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service @ named Node or @ named interface	Intra
63	RbMax(X)	Maximum testbed bit rate at node, or NFV “X” in unit time.	Mbit/s	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service @ named Node or @ named interface	Intra
64	RbMean(X)	Mean testbed bit rate at node, or NFV “X” in unit time.	Mbit/s	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service @ named Node or @ named interface	Intra

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
65	Latency(A-B)	Mean IP packet delay for standard MTU length between reference points X and Y	ms	Service	N/A	Better than H/W node implementation for same processing capacity.	Run-Time	Service: - between named nodes	Intra
66	Ttrans(X)	Time for a defined transaction "X" to complete successfully Eg a 2kbyte web page or 15 second UHD Video file with defined format.	ms	Service	2kbyte page < 1s Google welcome page request from mobile downloaded to mobile	Comparable or better to LTE benchmark	Run-Time	Service - Transaction between 2 entities defined as X.	Intra
67	FIT_NID rate	Number of intrusions from the network detected by the system, divided by the total number of attack instances against FITeagle server	%	Security	N/A	>99,8	Run-Time		Inter
68	OB_NID rate	Number of intrusions from the network detected by the system, divided by the total number of attack instances against OpenBaton server	%	Security	N/A	>99,8	Run-Time		Inter
69	OB_acc_ID_rate	Number of intrusions detected by the system, divided by the total number of illicit OpenBaton accesses	%	Security	N/A	>99,8	Run-Time		Inter

KPI #	KPI Name	KPI Description	KPI (Units)	KPI Type	Benchmark LTE (Rel-13)	Success Measure (Target)	Time Scope	Measurement Object(s)	Intra/Inter Test-Bed (Scope)
70	OS_acc_ID:rate	Number of intrusions detected by the system, divided by the total number of illicit OpenStack accesses	%	Security	N/A	>99,8	Run-Time		Inter

Table 5: Initial set of Security KPIs

5 How to measure the KPIs: Measurement processes during the Experimentation phases

The project will use built-in functionalities of OpenStack plus the support of a tool like Zabbix for collecting the information for deriving the KPIs. How to use Zabbix has been described in the SoftFIRE Handbook [(SoftFIRE, 2016)

In this initial phase of the project, the first wave of experimentation will keep the measures, and initial KPIs will be considered. The following basic measurements will be taken:

- Number of users accessing the system,
    - For each user, information on Number of instantiated VMs, average CPU and storage allocated,
- Number of hardware fault events,
- Total CPU allocated,
- Total storage allocated.

In the second phase, the number of the KPIs will be increased and the KPI set will be tuned up according to the needs of the users and their experiments. Security KPIs will be also integrated into the systems by means of specific solutions developed by the project.

By the end of the experimentation a consistent number of KPIs will be available and considered. From a user perspective, the Zabbix monitoring system is made available for use for business applications. This will allow the user to monitor and collect the measurements needed for their services within the SoftFIRE infrastructure.

6 Conclusions

Key Performance Indicators for SDN and NFV and in perspective of 5G software architecture require a considerable amount of definition work and consolidation. The meaningful measures and the way to achieve them should be progressively implemented and tried, through experiments to be carried out on a test platform such as SoftFIRE.

The novelty of the approach and the impacts that these measures can have on Operators' process are still to be determined, yet they seem to be well-aligned with the changes required for introducing DevOps approaches.

SoftFIRE's approach in this area will be very pragmatic: KPIs will be implemented and measured according to the identified needs of the platform as it is used by the experimenters. The selected KPIs will also have a perspective value for moving towards 5G experimentation that may follow the already planned ones.

The feedback of the experimenters in terms of evaluations of the infrastructure, its programmability and security will be the leading input in order to determine a minimal set of KPIs that an NFV/SDN platform should provide in order to be functional and fully operational. This feedback and the related results will be used as an important input for SoftFIRE's contribution to technical standardization activities in this field as planned in Task 4.3.

7 Bibliography

ETSI. (2014). *Network Functions Virtualisation (NFV); NFV Performance & Portability Best Practises.* Sophia Antipolis: ETSI.

Nexious. (2016). *Research Paper: Organizational Collaboration and New KPIs for NFV Success - A Roadmap for an NFV Implementation.* Nexius.

SoftFIRE. (2016). *Handbook: Guidelines and Rules for on-demand access to the SoftFIRE Testbed.* SoftFIRE.

8 List of Acronyms and Abbreviations

Acronym	Meaning
5G	Fifth Mobile Generation
API	Application Programming Interface
CPU	Central Processing Unit
FCAPS	Fault, configuration, accounting, performance, security
KPI	Key Performance Indicator
NFV	Network Function Virtualization
SDN	Software Defined Network
VM	Virtual Machine