D6.1: Report on design of federated cloud environment

Deliverable

Document ID	NECOS-D6.1
Status	Final
Version	1.0
Editors(s)	Billy Pinheiro (UFPA), Antônio Abelém (UFPA)
Due	30/10/2018
Submitted	29/10/2018

Abstract

This deliverable presents the evaluation environments and benchmark implementations developed to validate the NECOS platform through its key characteristics, including the set of addressed requirements (defined in prior deliverables) and the key characteristics validated in each implementation, the intended objectives, the software and hardware required for experimentation, a description of the used infrastructure, and the KPIs used in the assessment. The deliverable further demonstrates that NECOS experimentation needs can be met by well-known experimental testbed environments such as FIBER, Emulab, and 5GTONIC.

Table of Contents

LIST OF FIGURES

CONTRIBUTORS

Participant	Short Name	Contributors
Federal University of Pará	UFPA	Billy Pinheiro, Antônio Abelém
Federal University of Rio Grande do Norte	UFRN	Marcilio Lemos, Augusto Neto
University of Macedonia	UOM	Lefteris Mamatas, Ilias Sakellariou, Sofia Petridou, Polychronis Valsamas, Antonios Tsioukas
CPqD Telecom Research and Development Center	CPqD	Douglas Viroel
Federal University of Goiás	UFG	Sand Luz Corrêa, Kleber Vieira Cardoso, Leandro Alexandre Freitas
University of Campinas	UNICAMP	Christian Rothenberg, Celso Cesila, David Moura
Telefónica Investigación y Desarrollo	TID	Luis M. Contreras
Federal University of Uberlândia	UFU	Rafael Pasquini and Raquel Fialho Lafetá
Universitat Politècnica de Catalunya	UPC	Javier Baliosian and Joan Serrat
University College London	UCL	Stuart Clayman and Alex Galis

REVIEWERS

Reviewer	Short Name	Institution
Christian Rothenberg	UNICAMP	University of Campinas
Marcilio Lemos	UFRN	Federal University of Rio Grande do Norte
Lefteris Mamatas	UOM	University of Macedonia

Achronyms

Acronym	Description
API	Application Programing Interface
C3 PSC	Command, Control and Communications Public Safety Center
CAT	Cache Allocation Technology
CDN	Content Distribution Network
CMC	Chassis Management Controller
CMT	Cache Monitoring Technology
IDC	Itinerant Data Center
KC	Key Characteristic
KPI	Key Performance Indicator
LLC	Last Level Cache
LSDC	Lightweight Software Defined Cloud
MAE	Mean Absolute Error
MEC	Mobile Edge Computing
MQTT	Message Queuing Telemetry Transport
MSE	Mean Squared Error
MSMM	Multi-Slice Metrics Manager
NMAE	Normalized Mean Absolute Error
PoC	Proof of Concept
QoS	Quality of Service
RDT	Remote Diagnostic Technology
SDN	Software Defined Networking
SLA	Service Level Agreement
SMM	Slice Monitoring Manager
vCPE	Virtual Customer Premises Equipment
VLSP	Very Lightweight Network & Service Platform
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
vRAN	Virtual Radio Access Network
XMPP	eXtensible Messaging and Presence Protocol

Executive Summary

This document is associated with WP6, in particular to task 6.1 on Design of the Platform Integration and experiment definition. It focuses on the description of the evaluation environment developed for NECOS platform, also known as the Lightweight Software Defined Cloud (LSDC). In order to validate the LSDC key characteristics, the WP6 activities executed within the first twelve months of the project's life (M1 to M12) focused on the definition of a set of reference implementations addressing specific subsets of requirements of the NECOS platform. Each one of these reference implementations, namely Proof of Concept (PoC), are tightly coupled to the concepts and specifications already stated in previous deliverables, such as the specifications for Telco Cloud and the Mobile Edge Computing (MEC) scenarios developed in the Deliverable D2.1.

Scope

This document is the final result of the task T6.1 Design of the Platform Integration and experiment definition of WP6 of the NECOS project. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [Bradner 1997].

In the context of this document, the NECOS software solution is referenced as Lightweight Slice Defined Cloud (LSDC) and as NECOS platform indistinctly.

In order to keep track of the requirements along validation process, they are identified using the format Req TXX_YY, where (T) identify global requirement (G) or a PoC specific requirement (P), XX identifies the PoC sequence number and YY the requirement sequence number.

1 Introduction

The main objective of this deliverable is to describe the testing environment to validate the NECOS platform, also known as the Lightweight Software Defined Cloud (LSDC). To achieve this goal, the project started defining a set of reference implementations addressing specific subsets of requirements of the NECOS platform. Each one of these reference implementations is called Proof of Concept (PoC), which are tightly coupled to the concepts and specifications already stated in previous deliverables. In fact, Deliverable D2.1 [D2.1] developed specifications for scenarios, which in turn were based on the Telco Cloud and the Mobile Edge Computing (MEC) use cases. These scenarios were the starting point of the above mentioned PoCs. In addition, this document defines the minimal requirements of libraries, operating systems capabilities and the network connectivity between interconnecting federated islands are defined in this deliverable.

Each PoC is derived from a scenario and is constituted by a set of components and Models aligned with the NECOS architecture. In the context of its particular inspiring scenario, each PoC is addressing a set of the NECOS platform requirements, which are in turn supporting one or many of the NECOS key characteristics (KCs), as presented in **Figure 1**.

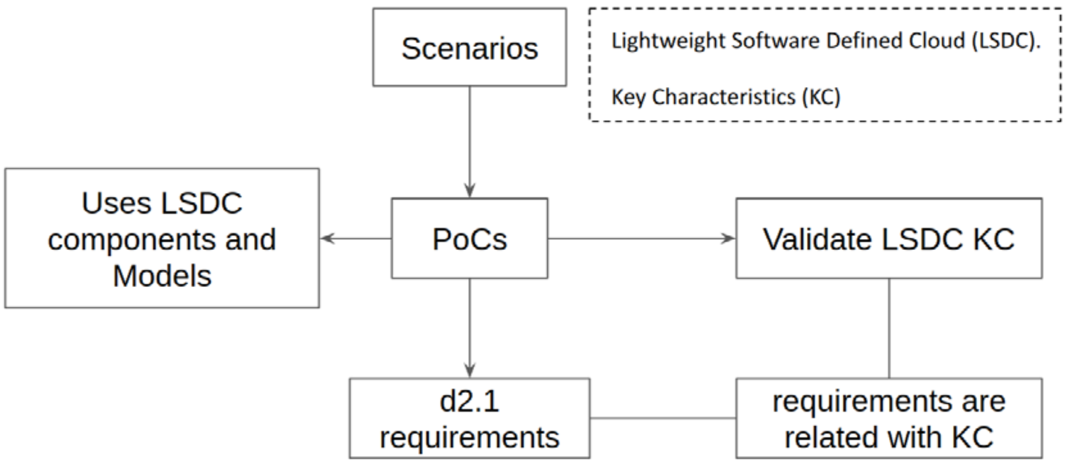
Scenarios
Lightweight Software Defined Cloud (LSDC).
Key Characteristics (KC)
Uses LSDC components and Models
PoCs
Validate LSDC KC
d2.1 requirements
requirements are related with KC

Figure 1. PoC's interaction with NECOS elements

The structure of each PoC was elaborated through collaborative work among the consortium partners. We started with a template with some initial fields to be filled by the different participants. Their answers and comments were used as feedback to generate a more elaborated template and so forth until the final model was reached, which is presented as Appendix A, and articulated around the following fields:

- Objectives - presents and details the main objectives to be achieved by the PoC;
- Hardware and software requirements - presents the infrastructure requirements needed by the PoC;
- Infrastructure for validation - presents the infrastructure that will be used to execute the PoC;
- KPIs - introduces the metrics to validate the PoC.

In addition, after some iterations, we were able to evaluate which PoCs could be continued or merged to be part of the Milestone 6.2 and become part of this deliverable. This exercise was important to

EUB-01-2017

give freedom for all members, allowing them to develop their research at the same time that the technical overlapping between PoCs was mitigated.

1.1 Structure of this document

This document is structured in five sections. Section 1 is this Introduction. Section 2 is the backbone of the different PoCs because it presents the NECOS key characteristics and the mapping of the requirements elicited in Deliverable D2.1 to these key characteristics. Section 3 presents the seven PoCs identified so far making use of the template shown in Appendix A. Section 4 is devoted to describe the three testbeds in scope of the project. Finally, Section 5 summarizes our conclusions and outlook.

1.2 Contribution of this deliverable to the project and relationship with other deliverables

The main objective of the WP6 is to showcase the ability of an integrated platform to serve both as a testbed of the integration of the prototyping results in the project and as a ground for demonstrating the use cases described in WP2.

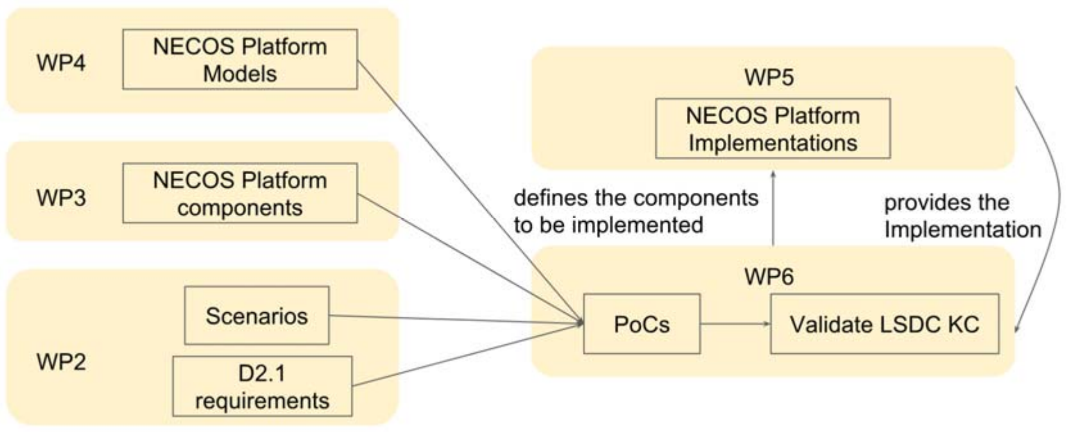
WP4
NECOS Platform Models
WP3
NECOS Platform components
WP2
Scenarios
D2.1 requirements
WP5
NECOS Platform Implementations
defines the components to be implemented
provides the Implementation
WP6
PoCs
Validate LSDC KC

Figure 2. Deliverables and work packages relationship

This deliverable is the first version of this integrated platform that has to be further elaborated and concluded in Deliverable D6.2. The first input for D6.1 is Deliverable D2.1 that constitutes the starting point of our test cases. In fact, the scenarios identified in D2.1 have been the origin of the PoCs. In addition, D6.1 is also getting inputs from Deliverable 3.1 [D3.1] and Deliverable 4.1 [D4.1]. These two deliverables specify the architecture and the APIs to drive the NECOS implementations. Therefore, the PoCs specified within D6.1 are aligned with both the architecture and APIs. Finally, D6.1 is the specification document of the implementations that will be conducted in WP5 and reported in Deliverable D5.1. These implementations will be subsequently tested and the results reported, among other information, within D6.2. These interactions are summarized in **Figure 2**, showing explicitly the connections of WP6 with the other work packages of the project.

2 NECOS Platform Validation Plan

Deliverable 2.1 [D2.1] defines the four key characteristics of LSDC, the novel approach for automating the process of configuring networked clouds by creating network cloud slices across all the resources in a set of federated resource domains. The four main characteristics are the core of the NECOS platform were chosen as drivers of the validation process in D6.1.

2.1 NECOS key characteristics

1) **Slice as a Service model:** It empowers **a new service model** – the Slice as a Service, by dynamically mapping service components to a slice. The enhanced management for the infrastructure creates slices on demand and slice management takes over the control of all the service components, virtualized network functions and system programmability functions assigned to the slice, and (re)configure them as appropriate to provide the end-to-end service.

2) **Easy and Flexible Slice Adaptation:** It enables easy reconfiguration and adaptation of logical resources in a cloud networking infrastructure, to better accommodate the QoS demand of the Slice, through the use of software that can describe and manage various aspects that comprise the cloud environment.

3) **Unified Software-based Management:** It allows each aspect of the cloud environment – from the networking between virtual machines to the SLAs of the hosted applications – to be managed via software. This reduces the complexity related to configuration and operation of the infrastructure, which in turn eases the management of the cloud infrastructure.

4) **Multi-Domain Slice Operation:** The LSDC platform will offer the ability to a specific cloud provider to federate his own infrastructure with other cloud providers with different configurations in order to realize virtualized services through the use of the Slice as a Service concept. The users of the LSDC APIs and platform will be able to create virtual services that can span the merged cloud infrastructure offered by different cloud providers. This concept is not purely technical, it can also encompass business, cultural, geographical or in any other domain.

2.2 Mapping of D2.1 requirements into LSDC key characteristics

In order to define how the LSDC key characteristics are related to the requirements from the scenario defined in Deliverable 2.1 [D2.1], we provide a questionnaire (Appendix B) for the project's members. The questionnaire was applied for each scenario. For each scenario, four questions arise, directly related to each LSDC key characteristic (KC). To answer the questions, we take one key phrase that represents the KC and relate it to the requirements. These are the key phrases used by KC:

- kc1: slice as a service paradigm;
- kc2: easy and flexible adaptation;
- kc3: software-based management;
- kc4: multi-domain slice operation.

The KC is evaluated regarding its relationship to each requirement. When a match is found, the requirement is placed in the KC list, considering that it MUST be present. However, if we find out that more than one requirement fulfills the key phrase, it becomes optional, i.e. it COULD be present.

Table 1. Mapping of D2.1 requirements into LSDC key characteristics

Scenario	LSDC KC 1	LSDC KC 2	LSDC KC 3	LSDC KC 4
5G Networks	[{RF.vRAN.3}]	[{RF.vRAN.3}]	[{RF.vRAN.1},{RF.vRAN.2}, {RF.vRAN.3}]	None
vCPE	[{RF.vCPE.1}]	[{RF.vCPE.2}, {RF.vCPE.6}]	[{RF.vCPE.1, RF.vCPE.2}]	None
Touristic Services	[{RF.Touristic(CD).1)}]	[{RF.Touristic(CD).3, RF.Touristic(CD).5}]	[{RF.Touristic(CD).1}, {RF.Touristic(CD).2}, {RF.Touristic(CD).5}]	None
Emergency	[{RF.emergency.1}]	[{RF.emergency.2}]	[{RF.emergency.4}]	[{RF.emergency.4}]

Table 1, summarizes the mapping of KC with the requirements. This table should be read using the following understanding:

- [{A,B}] : means that A AND B are necessary together.
- [{A},{B}]: means that A OR B are necessary.

The reasons for the relationship between the requirements and the KC are listed following organized by KC.

- Key Characteristic 1:
  - {RF.vRAN.3} could be used because it explicitly defines On-demand slice;
  - {RF.vCPE.1} could be used because it explicitly defines On-demand slice;
  - {RF.Touristic(CD).1} could be used because it explicitly defines Slice and slice-resource management;
  - {RF.emergency.1} could be used because it explicitly deals with slice creation, which is necessary to fulfill KC 1;
- Key Characteristic 2:
  - {RF.vRAN.3} could be used because it explicitly defines the adaptation of the slices on demand;
  - {RF.vCPE.6} could be used because it explicitly defines that the slice should be able to adapt to workload changes or {RF.vCPE.2} could be used because the slice must be manageable. We are assuming that management, in the cloud, also means reconfiguration if is necessary;
  - {RF.Touristic(CD).3, RF.Touristic(CD).5} need to be used together to fulfill the KC2 because they explicitly define "load-balancing of new bursts of user requests" and "able to request additional resources to be included in its slice" respectively.
  - {RF.emergency.2} could be used because it explicitly defines slice on-demand update.
- Key Characteristic 3:

  - {RF.vRAN.1} or {RF.vRAN.2} or {RF.vRAN.3} could be used considering implicitly that a software will provide the management for SLA, Accountability, and On-demand slice respectively.
  - {RF.vCPE.1, RF.vCPE.2} need to be used together to fulfill the KC3 because the RF.vCPE.1 provide management via software when the slice is created or removed and the RF.vCPE.2 provide management via software when the slice is running;
  - {RF.Touristic(CD).1} could be used since it explicitly defines management of slices via software or {RF.Touristic(CD).2} could be used because it deals with automated deployment management via software;
  - {RF.emergency.4} could be used because it explicitly defines Orchestration what is made via software.

- Key Characteristic 4:
  - {RF.emergency.4} could be used because it explicitly deals with provision of orchestration capabilities under different domains.

Looking into Table 1 it is possible to show that if we want to prove that the LSDC key characteristic 1 is present in a specific PoC, this PoC must implement the [{RF.vRAN.3}] in the 5G scenario or the [{RF.vCPE.1}] in vCPE scenario or [{RF.Touristic(CD).1)}] in Touristic scenario or [{RF.emergency.1}] in Emergency scenario. The same reasoning can be applied to the other key characteristics.

3 Description of the Project Testing Environments

This section provides the necessary software and hardware requirements to create the experimental environment for each proof of concept, which is subdivided into Global Requirements and Specific Requirements. Global Requirements accommodate the requirements needed by all PoCs. Finally, Specific Requirements are presented for each PoC to reflect their corresponding peculiarities. Also, a description of each PoC and the target validation infrastructure is provided.

The PoCs are namely: The VIM-on-demand paradigm over the NECOS platform (PoC01), Content Distribution Service over the NECOS platform (PoC02), Lightweight Network Slice monitoring framework in a multi-domain environment (PoC03), Itinerant Data Center (PoC04), Service Orchestration using Machine Learning Techniques (PoC05), Service Isolation in a Shared Infrastructure (PoC06), and WIreless Slicing sErvices (PoC07).

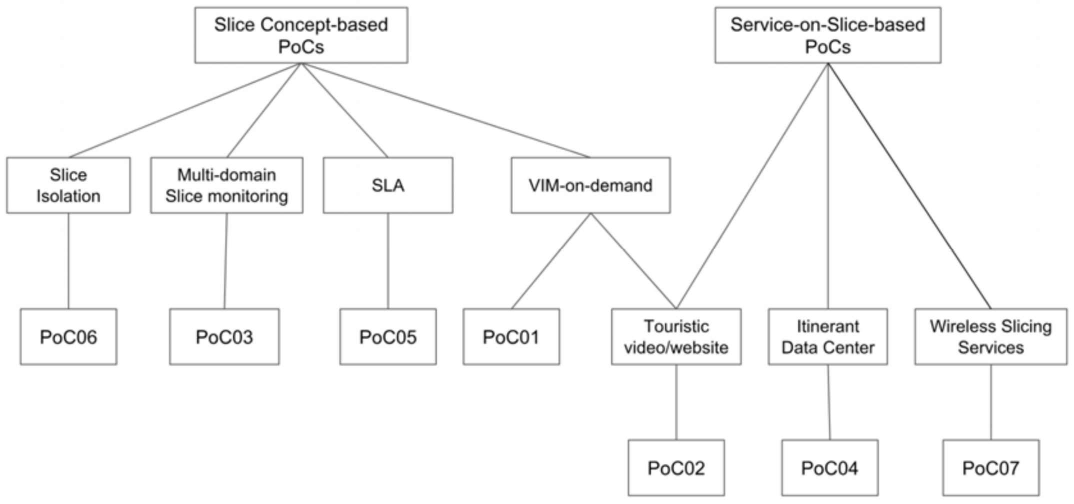

Figure 3. Overview of PoCs relationships

Figure 3 introduces the PoCs that are described in this section. Five of them are Slice-concept oriented, which means that their main target is to demonstrate a specific slice concept inside the NECOS platform. The remaining PoCs are Service-slice oriented, which means that their goal is to demonstrate services running in slices offered by the NECOS platform.

The NECOS platform components used in the PoCs are either developed from scratch or based on existing software components. Implementations from scratch usually implies fewer dependencies and requirements from external software and libraries, while, in general, implementation based on re-using solutions contribute to an increase of requirement. In this deliverable, we identify the cases based on existing open-source software, whereas deliverable 5.1 [D5.1] provides further details on the prototype implementation.

3.1 Global PoC requirements

3.1.1 Software requirements

- **[Req G00_01]** Automatic deployment: An LSDC experimenter MUST be able to configure the allocated resources. The customization items available should include hardware and software configurations, such as turning on/off some devices, installing software and changing OS parameters. Automatic infrastructure deployment tools like Ansible could be used, as presented in the Milestone M6.1 [M6.1];
- **[Req G00_02]** Hypervisors: The infrastructure SHOULD include virtualization software to enable the sharing of the physical infrastructure (computational and networking);
- **[Req G00_03]** Public IP addresses: The infrastructure MUST include publicly routable IP addresses on the Internet in order to provide interconnection between LSDC components hosted in different domains to enable cloud federation experiments. We mainly use IPv4 addresses in the NECOS PoCs.

3.1.2 Hardware requirements

- **[Req G00_04]** Computational Resources: The infrastructure MUST include the computational resources necessary to build a cloud, like virtual machines to hold servers and clients;
- **[Req G00_05]** Network Connectivity: The infrastructure MUST be interconnected by high performance network links;
- **[Req G00_06]** Network Programmability: The infrastructure MUST provide means to access open programmable network interfaces and network virtualization support.

3.2 Proof of Concepts (PoCs)

We now elaborate on the seven Proof of Concepts (PoCs) implemented in the context of the NECOS project. For each PoC, we provide: (i) a general description; (ii) a use-case diagram; (iii) its software and hardware requirements; (iv) the LSDC requirements to be validated from the PoC; (v) the particular validation infrastructure; and (vi) the Key Performance Indicators (KPIs) considered in the PoC.

3.2.1 PoC01: The VIM-on-demand paradigm over the NECOS platform

The main purpose of this PoC is to validate the architectural concepts of the DC Slice Controller, more specifically, the concepts of cloud slicing, bare metal allocation, and the VIM on-demand paradigm. For this purpose, we create two DC slice parts in the FIBRE testbed located at UFG and deploy a complete OpenStack cluster in each part. For each slice part, we allocate two physical servers (bare metal) where OpenStack components (Controller and Compute nodes) are being deployed on demand. The goal is to illustrate that: 1) the two slices are completely isolated from each other; and 2) the time taken to deploy an on-demand VIM is affordable, even for sophisticated VIMs such as OpenStack. In this first release of the DC Slice Controller, the following operations are being exercised:

- ***request_slice (Slice Specification, Time): Slice Part ID***
- ***activate_slice(Slice Part ID)***
- ***remove_slice (Slice Part ID)***

These operations are defined in the deliverable 4.1 [D4.1]. The ***request_slice*** operation returns the ***Slice Part ID*** to be used in the future for the slice removal.

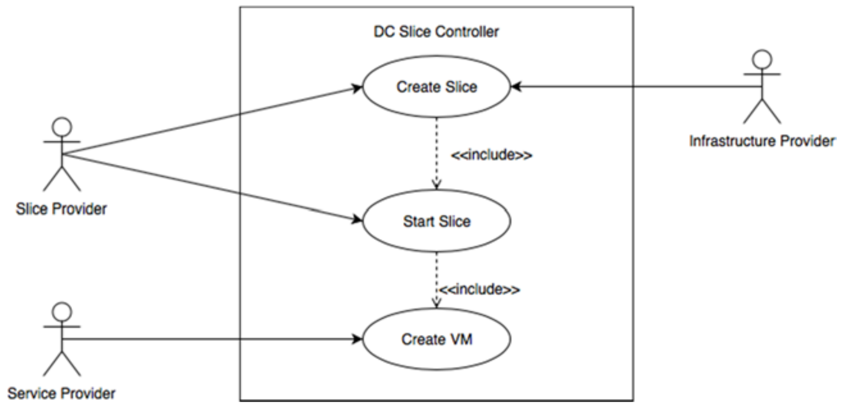

Figure 4. Use case diagram of PoC01

Figure 4 presents the use case diagram for the slice creation. The Slice Provider requests the slice creation from the Infrastructure Provider, passing it all the slice requirements, including: the slice name, slice owner, period of time the slice should be valid, the list of desired bare metal resources and their features, and the VIM type to be deployed. Inside the Infrastructure Provider, an instance of the DC Slice Controller allocates the specified resources, creates the slice inside the data center, and activate the slice. In this last step, the chosen VIM is deployed over the slice and the VIM entrypoint is returned to the Slice Provider. The Service Provider then is able to deploy its Virtual Machines (VMs).

3.2.1.1 Software requirements

- **[Req P01_01]** Message Queuing Protocol: The infrastructure MUST provide Advanced Message Queuing Protocol (AMQP) >= 1.0, so that the DC Slice components can communicate to each other as well to the resource control facilities.
- **[Req P01_02]** Network Hypervisor: The infrastructure MUST provide a network virtualization component, like the FlowVisor, to allow the virtualization of the network inside the data center.
- **[Req P01_03]** Virtual Switches: The infrastructure MUST provide solutions to create virtual switches, like Open vSwitch.
- **[Req P01_04]** Disk image loader: The infrastructure MUST provide a disk image generation and distribution system, such as Frisbee, so that entire VIM images can be saved and loaded.
- **[Req P01_05]** Libraries: The following libraries MUST be included:
  - Ruby support >= 2.3;
  - SQLite >= 3.25;
- **[Req P01_06]** Operating System Capabilities: The following Operating System Capabilities MUST be included:
  - Ubuntu Server 18.04
  - Kernel version >= 4.15.0-32-generic

3.2.1.2 Hardware requirements

- **[Req P01_07]** Regular nodes: the infrastructure MUST provide physical nodes that are able to be powered on/off and booted through the network, i.e., we use the Icarus nodes. These nodes play the role of the bare metal resources.
- **[Req P01_08]** Server nodes: the infrastructure MUST provide server nodes where the DC Slice Controller Components can be instantiated and the VIM images can be stored.
- **[Req P01_08]** SDN Switches: The infrastructure MUST provide SDN Switches with OpenFlow enabled to allow the network virtualization inside the data center.

3.2.1.3 LSDC requirements to be validated

This PoC addresses two requirements from the 5G Networks scenario, namely RF.vRAN3 and RN.vRAN1 which are used to validate the first three LSDC key characteristics, as presented in **Figure 5**.

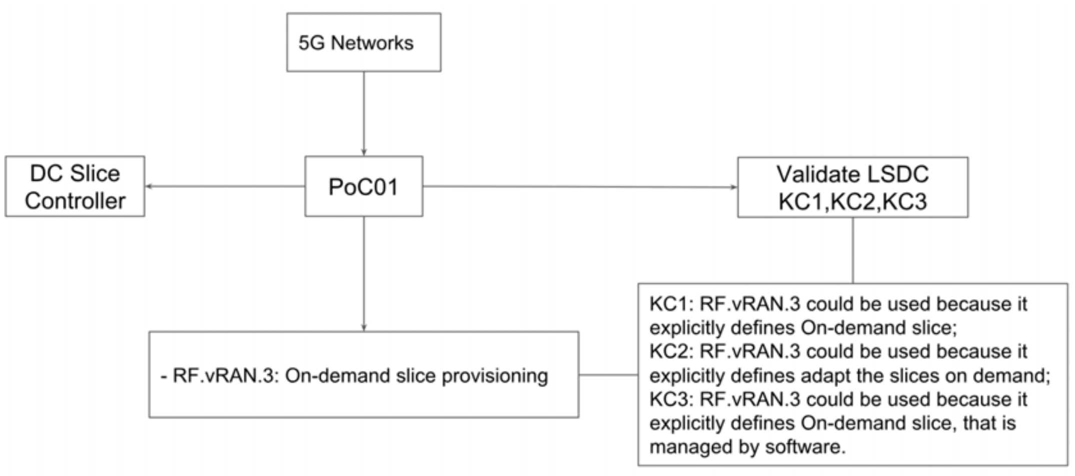
5G Networks
DC Slice Controller
PoC01
Validate LSDC KC1,KC2,KC3
- RF.vRAN.3: On-demand slice provisioning
KC1: RF.vRAN.3 could be used because it explicitly defines On-demand slice;
KC2: RF.vRAN.3 could be used because it explicitly defines adapt the slices on demand;
KC3: RF.vRAN.3 could be used because it explicitly defines On-demand slice, that is managed by software.

Figure 5. PoC01 relationships to NECOS scenarios, elements, characteristics, and requirements

3.2.1.4 Validation infrastructure

We use the FIBRE testbed located at UFG to run the PoC. The DC Slice Controller Components are being installed and instantiated on a server and use four Icarus nodes in the experiment, as illustrated in Figure 7. All the Icarus nodes have the same hardware configuration, and thus allow images generated on one node to be compatible with the others. Each Icarus node has an Arduino, called Chassis Management Controller (CMC), that is connected to the CMC network. The CMC is used to turn on/off the nodes, as well as to check the status (on/off) of the nodes. Each Icarus node is also connected to the Control network, from which the VIM images are installed and from which the node communicates to the other nodes. For each slice, we allocate two Icarus nodes, where the VIM components (Controller and Compute nodes) are being deployed.

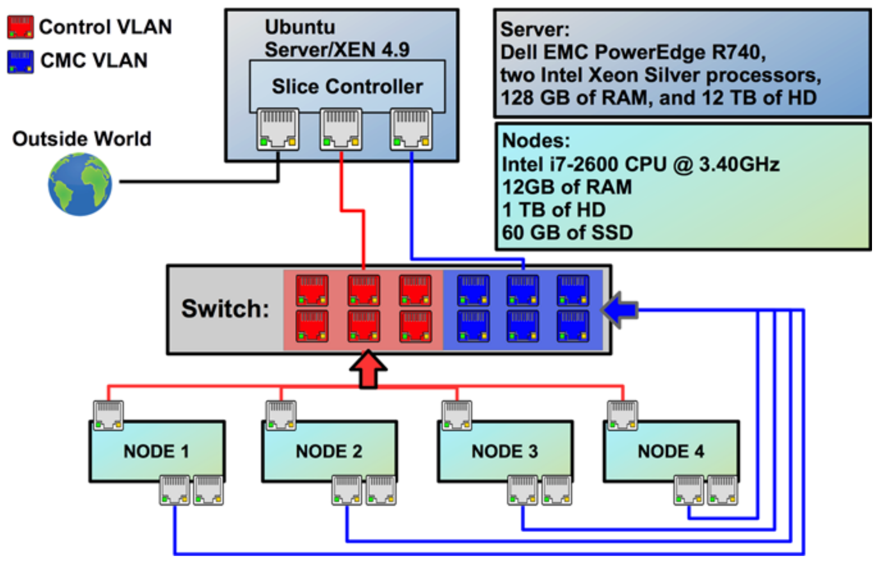

Figure 6. Target validation infrastructure

Figure 6 shows the connections of the Server with the Icarus nodes through the switch. The validation infrastructure is composed by the following equipment:

- Dell EMC PowerEdge R740 Server, equipped with two Intel Xeon Silver 4114 processor, 128 GB (8x 16GB RDIMM, 2666MT/s, Dual Rank) of RAM, and 12 TB of HD;
- 4 Icarus Nodes: Intel i7-2600 CPU Processor @ 3.40GHz, 12 GB RAM, HD 1 TB, SSD: 60 GB;
- 1 Switch Controller: Brocade (CES 2024C);

3.2.1.5 PoC KPIs

We use the following KPIs to quantify how the NECOS platform requirements are met in the PoC:

- Slice Isolation: measure isolation (slice) using two instances of OpenStack in the same IP address range. To do this, we will run the OpenStack dashboard and display that each controller node created can only access its own compute nodes; This allows to show that the resource slices created are isolated from one another.
- Slice Deployment Performance: we measure the time in seconds to deploy a slice, using different types of VIMs (i.e., demonstrating the VIM on-demand paradigm).

3.2.2 PoC02: Content Distribution Service over the NECOS platform

The performance of touristic video or website content provisioning can be significantly improved by utilizing edge clouds that host dynamic content caches near the end-users. Content caches are enablers for elasticity since they can be manipulated on demand based on the content popularity. This PoC investigates a novel elastic Content Distribution Network (CDN), where popular content is provided to the end-users through content caches deployed at the network edge. The content is hosted from Unikernel-based lightweight Virtual Machines.

We assume a number of web clients that periodically retrieve content from several web servers using diverse virtualization technologies (e.g., Unikernels, Containers or regular VMs). The end-users request large quantities of content and stress the infrastructure. For example, in cases of viral video content, the corresponding web servers struggle to operate efficiently. The end-users may suffer from wireless connectivity issues (e.g., due to mobility) and/or delays due to the overloaded cloud.

This PoC demonstrates how the proposed CDN facility can exhibit elasticity to match different scalability requirements, in terms of content requirements or number of supported users. The CDN exploits the novel NECOS slice-as-a-service capabilities to implement different cloud hierarchies (e.g., a core cloud hosting the content and an edge cloud hosting the content caches).

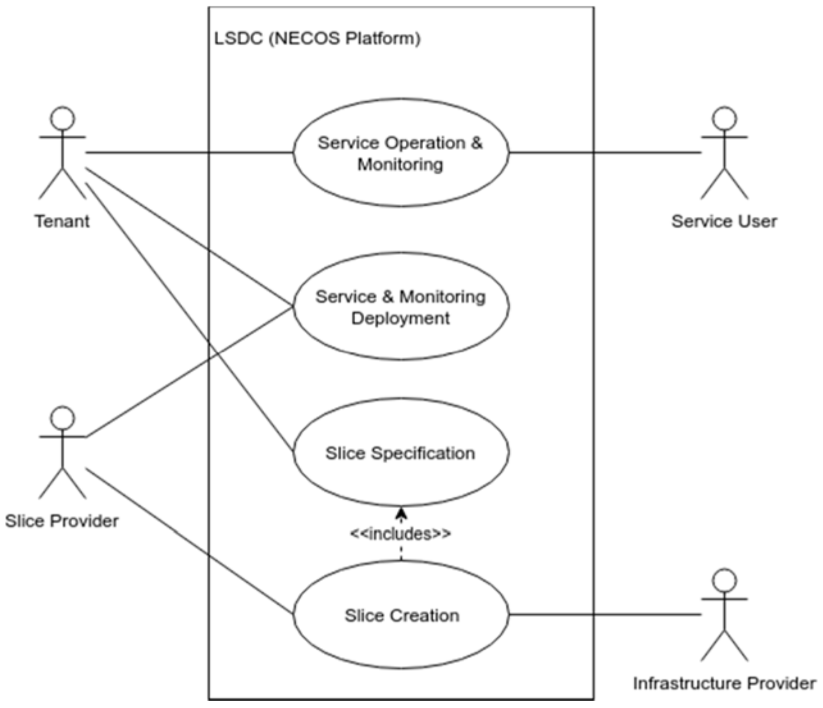

Figure 7. Use case diagram of PoC02

Figure 7 depicts the use case diagram regarding the specific PoC. The Slice Tenant, in our case the Touristic Service Center that aims to offer CDN services over the requested slice, initiates the slice creation process by providing an initial slice specification (or alternatively slice requirements or service requirements). The Slice Provider (LSDC Provider) creates the slice conforming with the corresponding specifications in cooperation with the Infrastructure Providers (WAN/DC Resource providers). The information on the created slice is passed from the Slice Provider to the Tenant, who deploys its service. The Tenant is responsible for the Service Operation and constantly monitors its execution to identify possible situations where elasticity operations should be taken place (e.g., add new content caches or physical servers). The deployed services are being consumed by the Service User.

3.2.2.1 *Software requirements*

- **[Req P02_01]** Slice monitoring tools: The infrastructure MUST provide monitoring tools like Collectd[1], influxdb[2], and Grafana[3] to perform the slice monitoring;
- **[Req P02_02]** Lightweight cloud technologies: The infrastructure MUST provide Lightweight cloud technologies like MirageOS[4], RumpKernel[5], ClickOS[6], and Docker[7];
- **[Req P02_03]** Operating System Capabilities: The following Operating System Capabilities MUST be covered:
    - Linux OS with a XEN-enabled kernel

3.2.2.2 *Hardware requirements*

- **[Req P02_04]** Network Resources: The infrastructure MUST provide layer 3 switches to isolate different slice deployments through VLANs;
- **[Req P02_05]** Server Resources: The infrastructure MUST provide physical nodes that are able to host lightweight Virtual Machines for the edge cloud (i.e., Mini PCs) or the core cloud (i.e., high-end servers)

3.2.2.3 *LSDC requirements to be validated*

This PoC addresses 8 requirements from the Touristic scenario, namely RF.Touristic(CD).1, RF.Touristic(CD).2, RF.Touristic(CD).3, RF.Touristic(CD).4, RN.Touristic(CD).1, RN.Touristic(CD).2, RN.Touristic(CD).3, and RN.Touristic(CD).4, which are used to validate two LSDC key characteristics (i.e., KC1 and KC3), as presented in **Figure 8**.

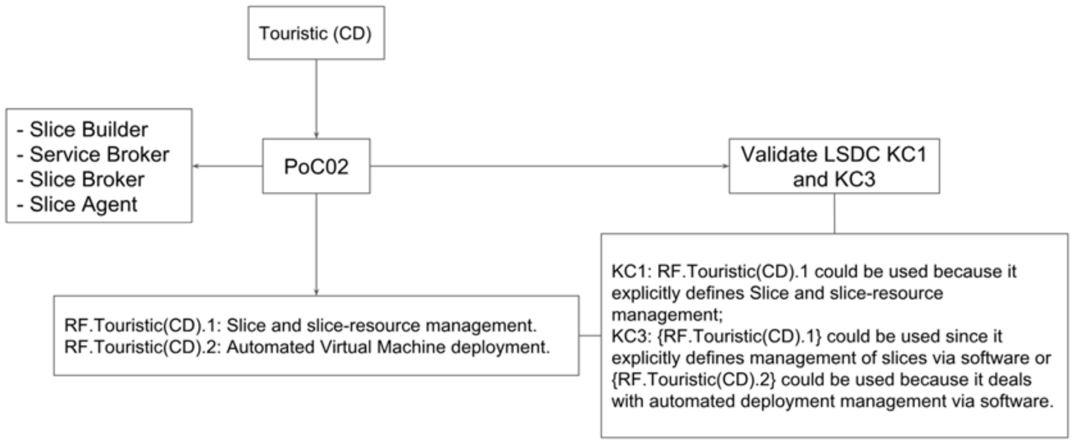
Touristic (CD)
- Slice Builder
- Service Broker
- Slice Broker
- Slice Agent
PoC02
Validate LSDC KC1 and KC3
RF.Touristic(CD).1: Slice and slice-resource management.
RF.Touristic(CD).2: Automated Virtual Machine deployment.
KC1: RF.Touristic(CD).1 could be used because it explicitly defines Slice and slice-resource management;
KC3: {RF.Touristic(CD).1} could be used since it explicitly defines management of slices via software or {RF.Touristic(CD).2} could be used because it deals with automated deployment management via software.

Figure 8. PoC02 relationships to NECOS scenarios, elements, characteristics, and requirements

[1] https://collectd.org/
[2] https://www.influxdata.com
[3] https://grafana.com
[4] https://mirage.io
[5] http://rumpkernel.org
[6] http://sysml.neclab.eu/projects/clickos/
[7] http://www.docker.com

3.2.2.4 Validation infrastructure

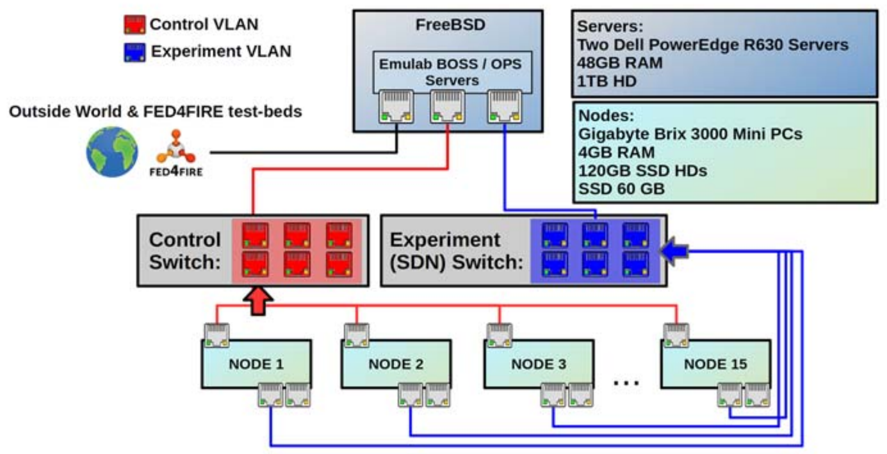

Figure 9. Target validation infrastructure

As shown in **Figure 9**, the infrastructure used for the PoC validation is the Emulab-based testbed at University of Macedonia (UOM) and other federated Emulab testbeds (e.g., via the FED4FIRE scheme). The UOM testbed consists of the following equipment:

- Two Dell PowerEdge R630 servers hosting the testbed software;
- Two L3 switches (one is an SDN switch);
- 15 Gigabyte Brix 3000 Mini PCs with 4GB RAM, 120GB SSD HDs and multiple interfaces (i.e., two fixed and two wireless), including programmable WiFi network cards with the Atheros chipset;
- PDUs to automatically control testbed node power.

The testbed uses separate channels and VLANs for the testbed control (i.e., control VLAN) and the communication of the experiment (i.e, experiment VLAN).

3.2.2.5 PoC KPIs

To demonstrate that the PoC meets the NECOS requirements, we define a number of KPIs, which follow:

- Web client performance
  - network throughput in Mbps;
  - download time in seconds;
  - connection time in seconds.
- Physical resources utilization per server host
  - CPU percentage utilization;
  - Memory allocation in MB;
  - Link throughput in Mbps.
- Slice performance
  - slice deployment time in seconds;
  - slice update time in seconds;

- slice decommission time in seconds.

3.2.3 PoC03: Lightweight Network Slice monitoring framework in a multi-domain environment

This experiment is based on the Command, Control, and Communications Public Safety Center (C3 PSC) scenario [D2.1] to validate a slice monitoring model using Prometheus, an open-source monitoring and alerting system. This monitoring model aims to present a multi-slice monitoring framework, sketched to find a solution for federated cloud environments like the one provided from the NECOS project.

The C3 PSC must monitor the operating situation of the subway lines, with a special focus on out-of-order situations that demand quick response and intervention from a public service first responders' team.

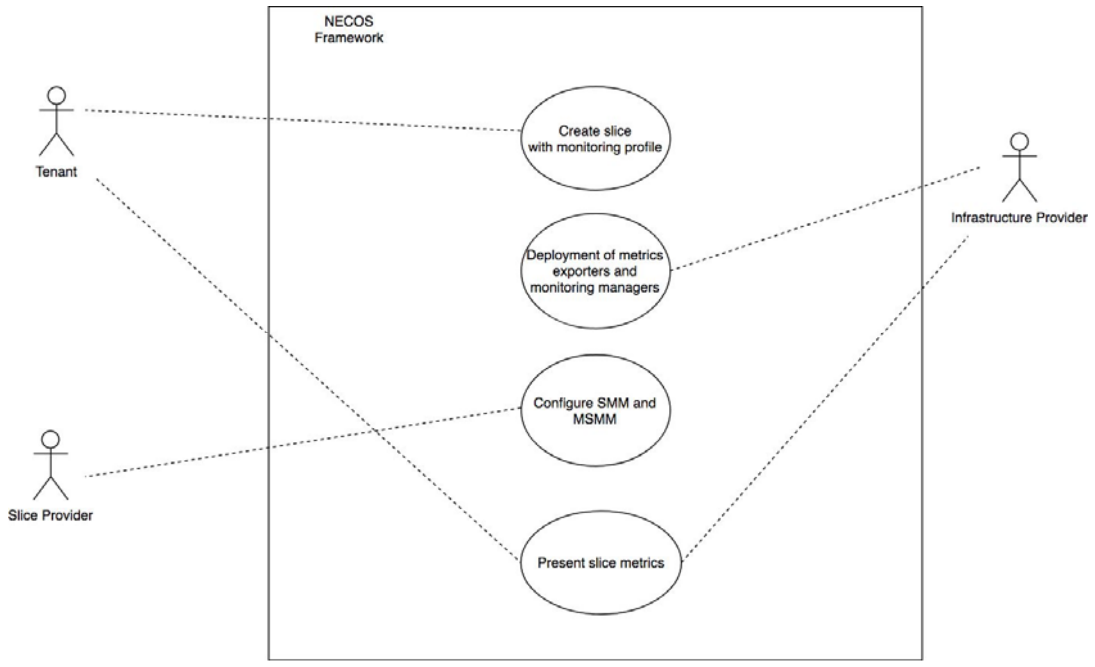

Figure 10. Use case diagram of PoC04

Figure 10 describes how to create the lightweight monitoring framework. The Tenant must request the slice creation to be performed by the Slice Provider using the Slice Resource Orchestrator (1). The corresponding Infrastructure Provider(s) deploys metric exporters related to the Slice DC's (2). In turn, the Slice Providers configure the Slice Monitoring Manager (SMM) and Multi-Slice Metrics Manager (MSMM) . This task is performed by IMA (3). The collected results are presented to the Infrastructure Provider and the Tenant, depending on the adopted monitoring profile (4).

3.2.3.1 *Software requirements*

- **[Req P03_01]** Streaming server: the infrastructure MUST provide a video server solution like VLC[8] Server to host the manifest files and video/audio tracks;
- **[Req P03_02]** Metrics exporters: the infrastructure MUST provide a library/server solution like Exporter to export existing metrics from third-party systems, as Prometheus metrics;
- **[Req P03_03]** Streaming client: The infrastructure MUST provide video client solution like VLC Client to playback video/audio tracks;
- **[Req P03_04]** Slice Monitoring Manager: the infrastructure MUST provide a system monitoring and alerting toolkit like Prometheus[9] to get a multi-dimensional data model with time series data identified by metric name and key/value pairs, as well as a flexible query language to leverage this dimensionality. Moreover, it requires a visual platform like Grafana to provide event monitoring and time series analytics.
- **[Req P03_05]** Operating System Capabilities: The following Operating System Capabilities MUST be included:
    - Debian 9 x64 Minimal;
    - KVM support.

3.2.3.2 *Hardware requirements*

- **[Req P03_06]** Mobile Devices: The infrastructure MUST include mobile devices (e.g., smartphones, tablets) intended to run applications that give end-users access to experimental services available on the combined wired/wireless substrate.

3.2.3.3 *LSDC requirements to be validated*

This PoC addresses 3 requirements from the Emergency scenario, namely the RF.emergency.1, RF.emergency.3, and RF.emergency.4, which are used to validate 3 LSDC key characteristics (i.e., KC1, KC3 and KC4), as presented in **Figure 11**.

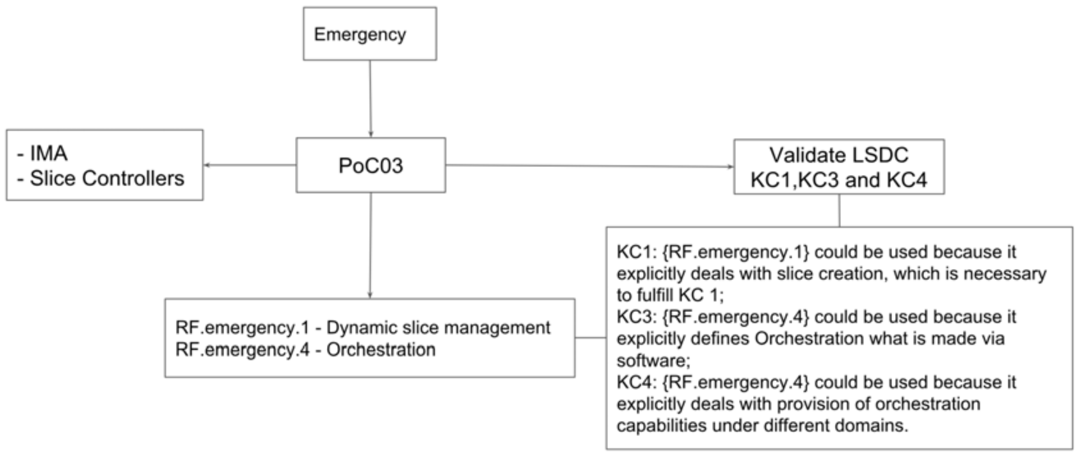
Emergency
- IMA
- Slice Controllers
PoC03
Validate LSDC KC1,KC3 and KC4
RF.emergency.1 - Dynamic slice management
RF.emergency.4 - Orchestration
KC1: {RF.emergency.1} could be used because it explicitly deals with slice creation, which is necessary to fulfill KC 1;
KC3: {RF.emergency.4} could be used because it explicitly defines Orchestration what is made via software;
KC4: {RF.emergency.4} could be used because it explicitly deals with provision of orchestration capabilities under different domains.

Figure 11. PoC03 relationships to NECOS scenarios, elements, characteristics, and requirements

[8] https://www.videolan.org/vlc/
[9] https://prometheus.io/

3.2.3.4 *Validation infrastructure*

This testbed is envisioned to represent an emergency use case on a subway train station, preventing Subway Incidents. The current VM resources are summarized in Table 2:

Table 2. Summarized VM resources

Machine	Role	vCPU	Memory	Applications
Slice1-Server	Streaming Server	1	1 GB	Exporters + VLC
Slice2-Server	Streaming Server	2	2 GB	Exporters + VLC
Slice3-Server	Streaming Server	3	3 GB	Exporters + VLC
Host1-Client	Streaming Client	2	1 GB	Exporters + VLC
Host2-Client	Streaming Client	2	1 GB	Exporters + VLC
Host3-Client	Streaming Client	2	1 GB	Exporters + VLC
SMM1	Slice Monitoring Manager	2	1 GB	Prometheus + Grafana
SMM2	Slice Monitoring Manager	2	1 GB	Prometheus + Grafana
SMM3	Slice Monitoring Manager	2	1 GB	Prometheus + Grafana
MSMM	Multi-Slice Metrics Manager	2	1 GB	Prometheus + Grafana

Figure 12 shows the relationship between the server, the client, and the monitoring system. In this test measurement scenario, three slices are running at the same time and are monitored by slice monitoring managers (SMM) that are gathered by the Multi-slice metrics manager (MSMM). With this approach, we envision to confirm MSMM operation, verifying that the metrics are collected on a separate per slice basis, with a correct and simple presentation. Here, MSMM collects three performance metrics (CPU, memory and bandwidth), to monitor an adaptive video streaming service deployed on each slice.

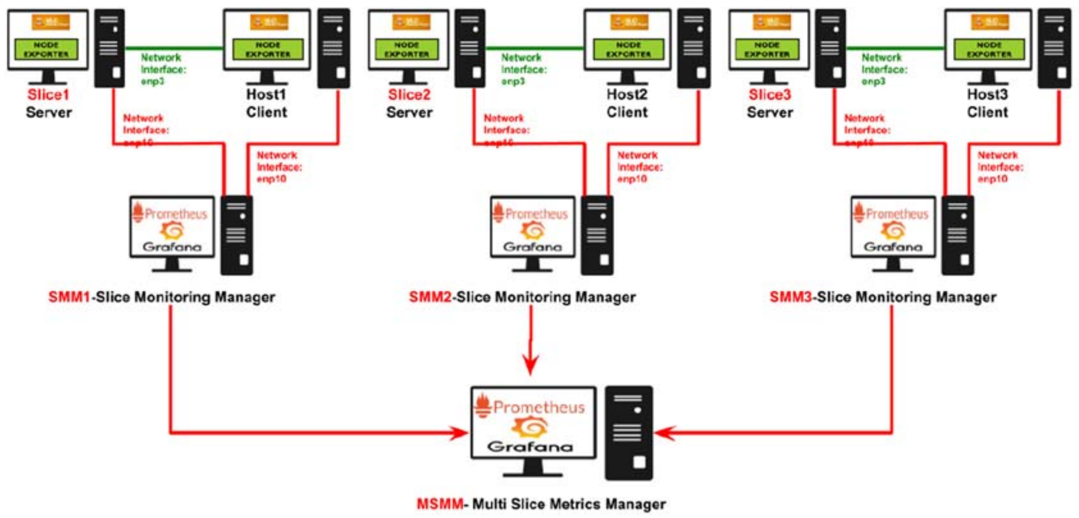

Figure 12. Target validation infrastructure

In **Figure 12**, three independent video streams are delivered in distinct slices. Three client hosts are created (one per slice) to receive the video in the tests, and the Slices are running at the same time. The green link represents the adaptive video streaming interaction between the server and client, while the red link represents the monitoring solution interface.

3.2.3.5 PoC KPIs

We quantify the level of PoC conformance with the NECOS platform requirements using the following KPIs:

- Dynamic slice management monitoring capability to present alerts that capture system dynamics (metrics: ratio slice alerts/events of interest);
- Timely slice management monitoring capability to present timely alerts (metrics: average time interval, in seconds, between alerts and events of interest);
- Multi-domain monitoring capability (Ratio {# of metrics gathered from multiple domain environments/ Total of metrics from distinct environments).

3.2.4 PoC04: Itinerant Data Center (IDC)

This PoC uses the Emergency Scenario [D2.1] as a base and evaluates the time to deploy a new service over an itinerant data center. This itinerant data center is composed of one small computer, one wireless router, and one energy supply. For now, we are discussing the use of video services (read-only) and questionnaire capture services (read-write).

The IDC is the Fog Computing concept [Mouradian 2018] applied as a service. The main idea is to have an itinerant data center that could host microservices on demand, allowing the service coverage of islands without Internet access, thus, islands that normally did not reach the core network to access the services. These services (microservices) need a slice allocation, deployment of the service over the slice and the migration/replication of the service in the itinerant data center equipment.

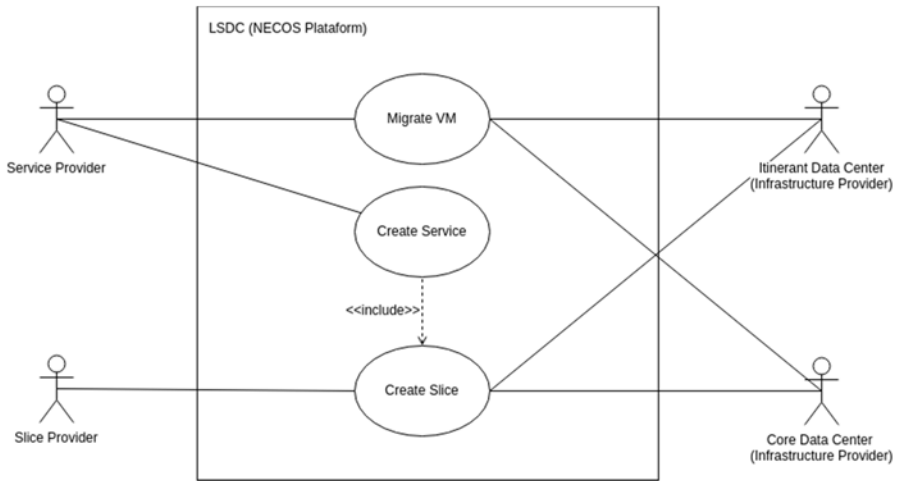

Figure 13. PoC4 Use Case Diagram

Figure 13 shows how the IDC service is being created. The Service Provider must request the slice creation, which is performed by the Slice Provider using the Slice Resource Orchestrator. The Service

Provider must insert the service definition to be handled by the Slice Provider in order to receive the slice definition used to select the resources.

This PoC is not using the Marketplace (as defined in deliverables D3.1 and D4.1), thus the agreement between the Slice Provider and the Infrastructure Provider is performed out of band, i.e., the slice provider admin must manually provide the set of resources to be used, based on an existing agreement. Once the Slice is created in the Core Data Center and the service is instantiated, it is possible to move/replicate part of the service to the Itinerant Data Center, applying the Fog Computing concept enabled by novel capabilities of LSDC.

3.2.4.1 *Software requirements*

- **[Req P04_01]** Streaming server: The infrastructure MUST provide a video server solution like DASH[10]/Apache2 to host the manifest files and video/audio tracks.
- **[Req P04_02]** Libraries: The following libraries MUST be included:
  - HUG[11];
  - Falcon[12];
  - Python support >= 3.5.
- **[Req P04_03]** Operating System Capabilities: The following Operating System Capabilities MUST be included:
  - Unix-like

3.2.4.2 *Hardware requirements*

- **[Req P04_04]** SDN Switches: The infrastructure MUST provide SDN Switches with OpenvSwitch[13] enabled;
- **[Req P04_05]** Wireless Routers: The infrastructure MUST provide Wireless routers to create an access network for the service.

3.2.4.3 *LSDC requirements to be validated*

This PoC addresses 2 requirements from the Emergency scenario, namely the RF.emergency.1 and RF.emergency.4 requirements, which are used to validate 3 LSDC key characteristics, as presented in **Figure 14**. In addition, a new requirement (RF.emergency.5- Offline slice execution, when it loses the connection with the core network.) to be further elaborated in D2.2 is presented.

[10] https://www.encoding.com/mpeg-dash/
[11] http://www.hug.rest/
[12] https://falconframework.org/
[13] https://www.openvswitch.org/

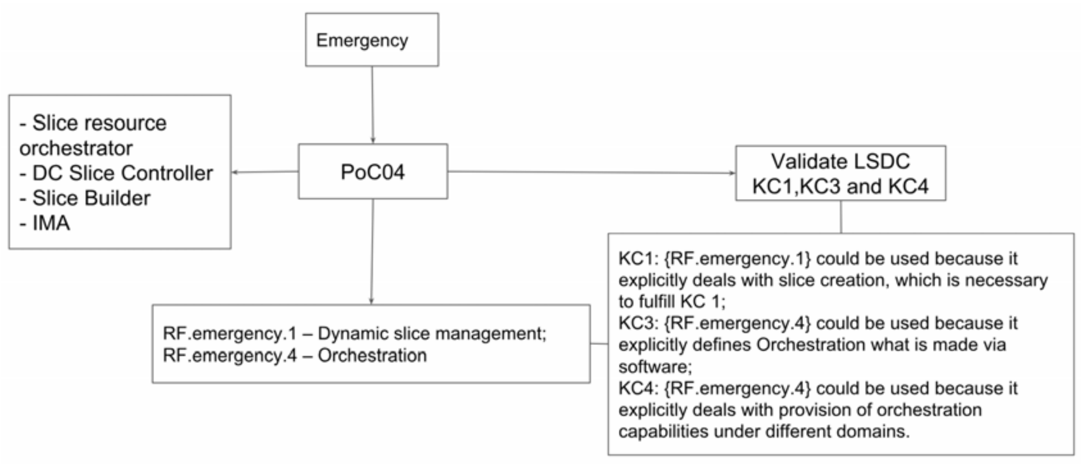
Emergency
PoC04
- Slice resource orchestrator
- DC Slice Controller
- Slice Builder
- IMA
Validate LSDC KC1,KC3 and KC4
RF.emergency.1 – Dynamic slice management;
RF.emergency.4 – Orchestration
KC1: {RF.emergency.1} could be used because it explicitly deals with slice creation, which is necessary to fulfill KC 1;
KC3: {RF.emergency.4} could be used because it explicitly defines Orchestration what is made via software;
KC4: {RF.emergency.4} could be used because it explicitly deals with provision of orchestration capabilities under different domains.

Figure 14. PoC04 relationships to NECOS scenarios, elements, characteristics, and requirements

3.2.4.4 Validation infrastructure

The PoC uses the FIBRE[14] testbed for validation and testing. The tests carried out with this PoC use a virtual resource infrastructure (e.g., a virtual topology of OpenFlow switches and nodes) to emulate an Edge Computing Environment. An OpenFlow Controller controls the scenario for the PoC's validation, e.g., providing packet flow switching and connectivity among different data centers.

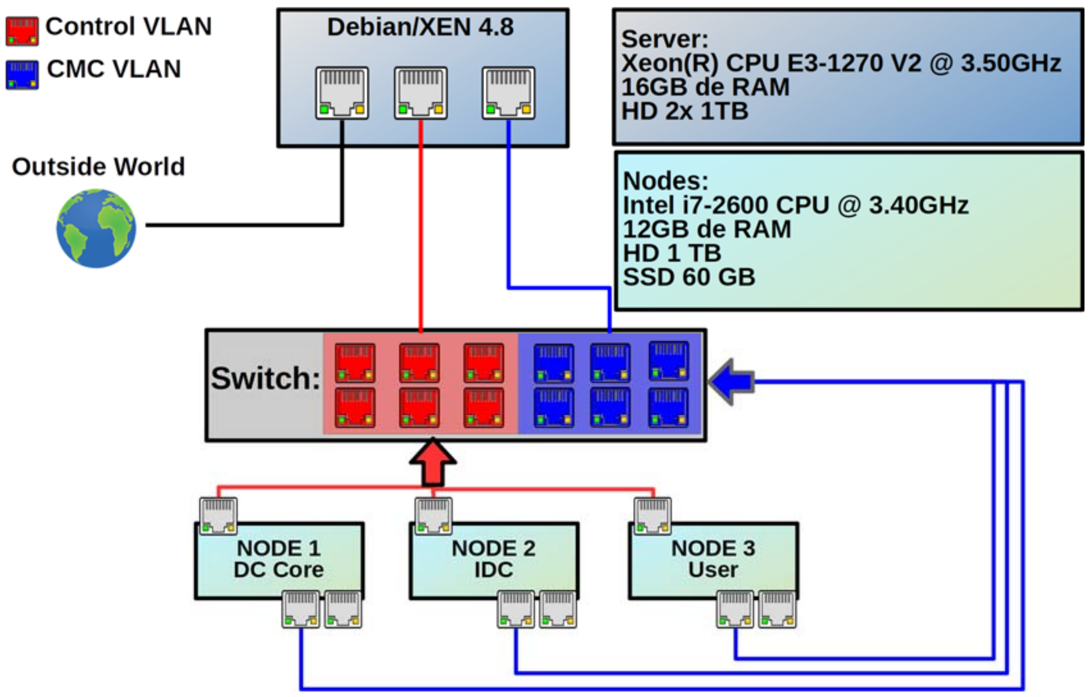

Figure 15. Target validation infrastructure

Figure 15 shows the OpenFlow switch used to create the virtual links connecting the DC core (running the DC slice controller), the IDC (running the DC slice controller) and the User (Running the client) that

[14] https://fibre.org.br/

are placed in Icarus nodes. The server hosts a VM with the Slice Resource Controller, the Slice Builder, and the IMA.

3.2.4.5 PoC KPIs

We define the following KPIs to quantify how the NECOS platform requirements are being met in the PoC:

- Support of slice creation and scaling operations on-demand: Provisioning time in seconds.
- Support for migrating virtual machines to the itinerant data center and virtual machine isolation: VM migration time in seconds.
- Allowing for offline slice execution on the itinerant data center: Percentage of service time availability during the offline period.

3.2.5 PoC05: Service Orchestration using Machine Learning Techniques

This PoC demonstrates the capability of a service orchestrator to detect SLA conformance/violation, performing a root cause analysis in the case of SLA violation and, therefore, defining the required solution to bring the service back to SLA conformance.

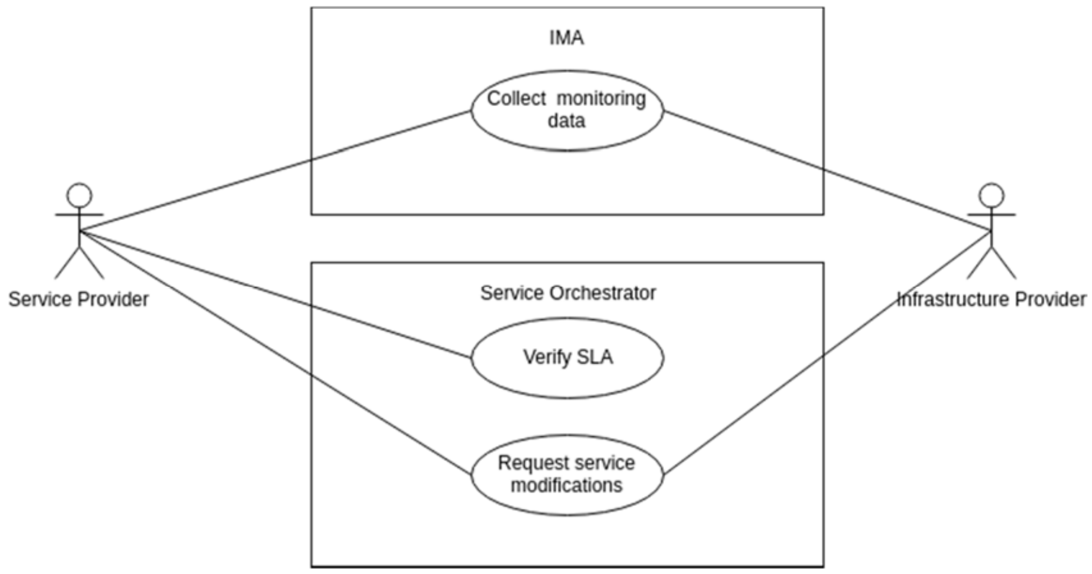

Figure 16. PoC5 Use Case Diagram

The Infrastructure Provider is generating monitoring data during the slice lifecycle and such monitoring data is accessible to the Service Provider via an IMA module present at the Slice Provider. The Service Provider collects monitoring data from the IMA and uses such monitoring data from its slice at its Service Orchestrator, as we can see in **Figure 16**. The Service Orchestrator processes the data to verify if the SLA is under conformance or violation. In case of violation, it uses the data to perform an RCA (root cause analysis) and, finally, to specify what are the required service modifications to bring the service back to SLA conformance.

3.2.5.1 Software requirements

- **[Req P05_01]** DHT solution: The infrastructure MUST provide a DHT solution like Cassandra acting as a distributed database with replication support;
- **[Req P05_02]** Libraries: The following libraries MUST be included:
    - Python >= 2.7
    - JAVA >= 8

3.2.5.2 Hardware requirements

- **[Req P05_03]** Cloud Datacenter Features: The infrastructure MUST provide Virtual Machines in which components of the service will be instantiated;
- **[Req P05_04]** WAN Operator Capabilities: The infrastructure MUST provide OpenFlow-enabled switches to provide communication in between the slice parts;
- **[Req P05_05]** Edge Cloud Support: The infrastructure MUST support containers for the provisioning of service instances closer to the end user/consumers.

3.2.5.3 LSDC requirements to be validated

This PoC addresses 3 requirements from the Touristic scenario, namely the RF.Touristic(CD).1, RF.Touristic(CD).4, and RN.Touristic(CD).3 requirements, which are used to validate 2 LSDC key characteristics (i.e., KC1 and KC3), as presented in **Figure 17**.

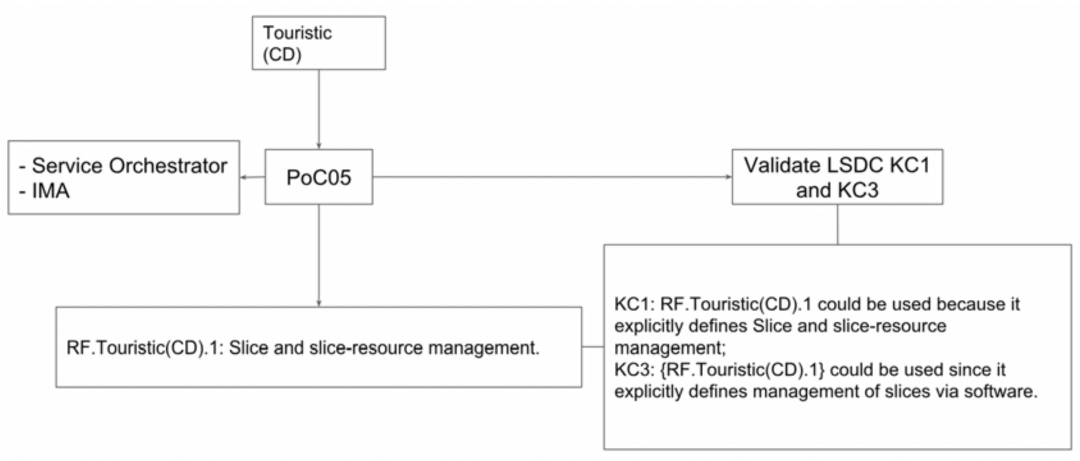
Touristic (CD)
- Service Orchestrator
- IMA
PoC05
Validate LSDC KC1 and KC3
RF.Touristic(CD).1: Slice and slice-resource management.
KC1: RF.Touristic(CD).1 could be used because it explicitly defines Slice and slice-resource management;
KC3: {RF.Touristic(CD).1} could be used since it explicitly defines management of slices via software.

Figure 17. PoC05 relationships to with NECOS scenarios, elements, characteristics and requirements

3.2.5.4 Validation infrastructure

The testbed used to perform this experiment is composed of three servers running OpenStack Queens, 10 raspberry Pi's running Kubernetes[15] and four OpenFlow switches. Such equipment is necessary to emulate a slice with resources scattered in cloud (OpenStack), WAN (OpenFlow) and Edge/Fog (Kubernetes) features.

[15] http://kubernetes.io/

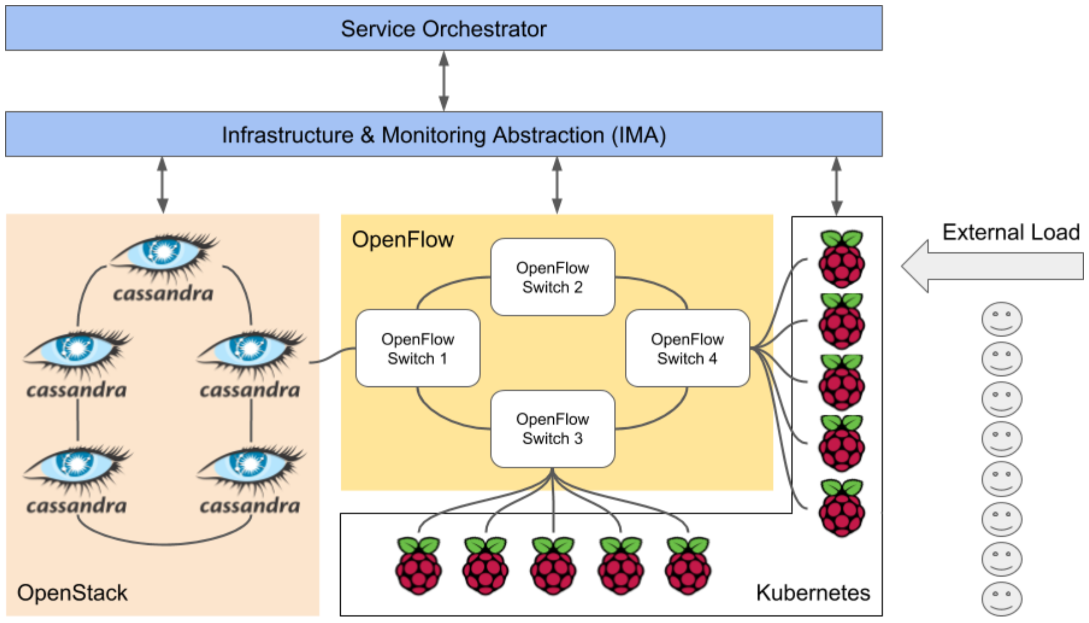

Figure 18. Target validation infrastructure

Figure 18 shows the front-end application running in Kubernetes which persists data in a Cassandra cluster running in our OpenFlow-enabled cloud. All the data is pushed through the OpenFlow network. The testbed also allows other services on top of our infrastructure, and we are currently developing a DASH-based video-on-demand service for future experimentations. By controlling the load patterns that clients put on our services, we collect data from the infrastructure to train machine learning classifiers and regressors, i.e., to infer if the SLAs are under conformance or violation. The methods we apply are service-agnostic.

3.2.5.5 PoC KPIs

For measuring how the NECOS platform requirements are being met in the PoC, we define the following KPIs:

- Number of application users serviced within an area to demonstrate the scalability aspect;
- Accuracy of prediction of service demands and resource utilization for the intelligence aspects of multi-domain orchestration, using, for example, Normalized Mean Absolute Error (NMAE), Mean Absolute Error (MAE), or Mean Squared Error (MSE).

3.2.6 PoC06: Service Isolation in a Shared Infrastructure

As presented in D3.1, the slicing operation modes define alternative approaches to resource partitioning, depending on the level of resource management the slice requires. The slice operation mode 1 allows resource partitioning at VIM level and the direct Resource Orchestration interaction with a VIM in order to manage the allocated resources. In a shared infrastructure created for multi-tenancy, the partitioning of compute resources can be made at CPU level, where two or more tenants share cores from a single CPU chip. In these scenarios is important to guarantee the complete isolation among tenants to avoid performance penalties caused by noisy neighbors that can consume non-isolated shared resources.

This PoC exercises the partitioning of resources at CPU level and the impact caused by noisy neighbors in a multi-tenancy environment. The proposed solution leverages the concept of compute-slicing to resources like the CPU's Last Level Cache (LLC) and the Memory Bandwidth channel. The benefits of this approach is the full isolation between tenants resource to avoid the performance impact commonly caused by memory-hungry applications, sharing the same environment. The PoC covers the development of components to control and monitor those shared resources and provide an optimal resource allocation along with the NECOS Platform, providing constant monitoring and dynamic resizing of slicing resources, as needed. The PoC goal is to achieve two objectives:

1. The isolation of compute resource by introducing components capable of monitoring and allocating LLC resources;
2. Dynamically adapt the Cache allocation based on performance indicators enhancing the slice performance.

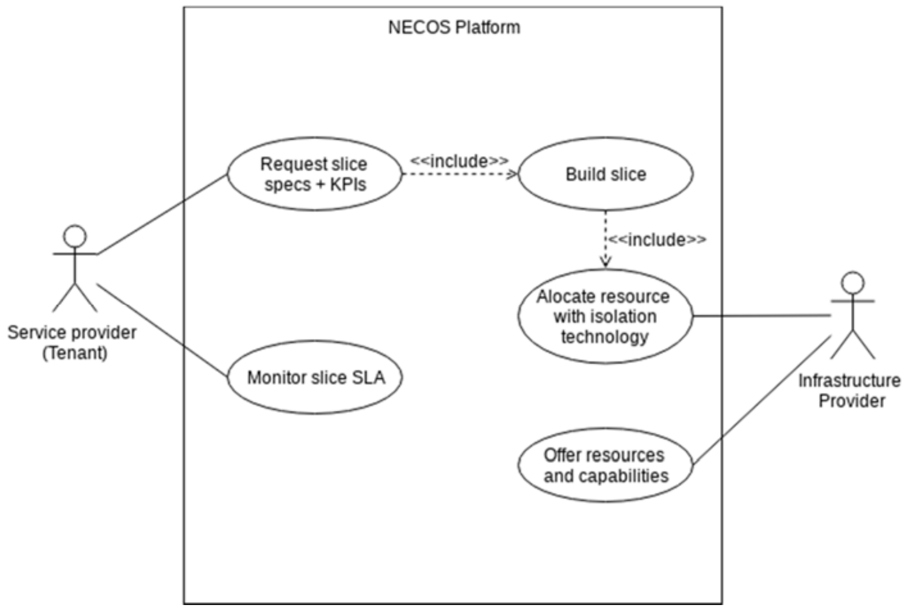
NECOS Platform
Request slice specs + KPIs
<<include>>
Build slice
<<include>>
Alocate resource with isolation technology
Service provider (Tenant)
Monitor slice SLA
Infrastructure Provider
Offer resources and capabilities

Figure 19. PoC06 Use case diagram

Figure 19 describes the interactions between the main actors involved in this PoC and shows on which context the cache allocation feature should be placed. It starts with the service provider (tenant) requesting a slice to the NECOS Platform, providing slice requirements and KPI's. The NECOS Platform triggers a slice creation process that searches for slice parts to fulfill the request. On the other side, the resource provider offers its domain's resources, built in a shared infrastructure that is capable of isolating multiple tenants, by using cache isolation technologies. The resources that belong to the requested slice are allocated in the provider's domain along with isolation technologies. After slice and service instantiation, the service provider can monitor the slice performance through NECOS IMA interfaces.

3.2.6.1 *Software requirements*

- **[Req P06_01]** Prometheus Monitoring System[16] >= 2.4: the infrastructure SHOULD have an efficient monitoring system that can store custom cache-related metrics and provide an API for fast queries based on time series;
- **[Req P06_02]** MoonGen[17]: the infrastructure MUST provide a customizable traffic generator like MoonGen to simulate clients of the virtual Customer Premises Equipment (vCPE) platform and to provide system performance indicators like throughput and latency;
- **[Req P06_03]** Memtester [18] >= 4.0: application used to simulate the behavior of noisy neighbors in the shared infrastructure;
- **[Req P06_04]** Operating System Capabilities: The following Operating System Capabilities MUST be included:
    - Linux OS with kernel version >= 4.10 compiled with INTEL_RDT flag enabled. The cache allocation and monitoring technologies flags must be enabled in kernel boot parameters.
    - Libraries: The following libraries MUST be included:
        - PQoS library[19] version 2.0: this library provides an API for detecting and to configure Intel RDT technologies. The version 2.0 supports newer Linux kernel extensions to program these technologies.

3.2.6.2 *Hardware requirements*

- **[Req P06_04]** The physical infrastructure MUST provide Intel Xeon Processors with support for Intel Resource Director Technology[20] like Cache Monitoring Technology (CMT) and Cache Allocation Technology (CAT). The Memory Bandwidth Monitoring (MBM) and the Memory Bandwidth Allocation (MBA) technologies are not mandatory but they can also be used, if available.

3.2.6.3 *LSDC requirements to be validated*

This PoC addresses 5 requirements from the vCPE scenario, namely the RF.vCPE.4, RF.vCPE.5, RF.vCPE.6, RN.vCPE.1, and RN.vCPE.2 requirements, which are used to validate 1 LSDC key characteristics (i.e., KC2), as presented in **Figure 20**.

[16] https://prometheus.io/
[17] https://github.com/emmericp/MoonGen
[18] http://pyropus.ca/software/memtester/
[19] https://github.com/intel/intel-cmt-cat/tree/master/pqos
[20] https://github.com/intel/intel-cmt-cat

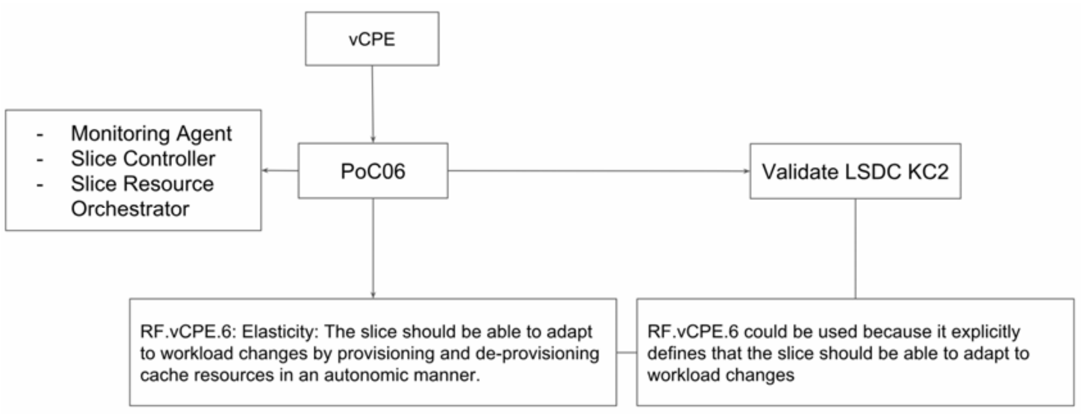
vCPE
- Monitoring Agent
- Slice Controller
- Slice Resource Orchestrator
PoC06
Validate LSDC KC2
RF.vCPE.6: Elasticity: The slice should be able to adapt to workload changes by provisioning and de-provisioning cache resources in an autonomic manner.
RF.vCPE.6 could be used because it explicitly defines that the slice should be able to adapt to workload changes

Figure 20. PoC06 relationships to NECOS scenarios, elements, characteristics, and requirements

3.2.6.4 Validation infrastructure

The testbed used to perform this experiment is composed by one server with Intel® Xeon® processors supporting the Intel Cache Monitoring Technology (CMT) and Cache Allocation Technology (CAT), with at least 8 CPUs cores, 16GB of RAM and 100GB of storage. These resources are being provided at bare-metal level. Another physical server is used to accommodate the software traffic generator, i.e., simulating the traffic from the vCPE clients to the primary server, using high-speed network interfaces.

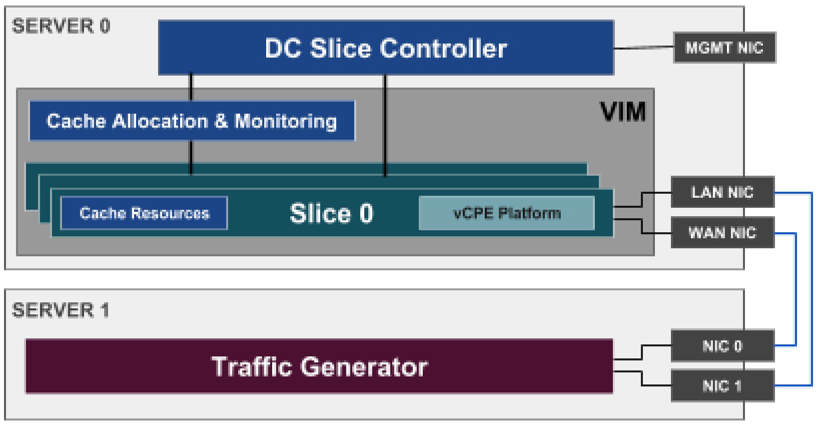

Figure 21. The two-server testbed of the Slice Isolation PoC

Figure 21 shows the connectivity between two servers to create the validation scenario. Server 0 holds all slices' instances represented by cloud slices only, i.e., the minimum requirement to demonstrate this scenario. In this server, that fulfils the hardware requirements needed to create cache allocations, we deploy the Cache Allocation application along with the NECOS Slice Controller, responsible for creating and managing slices within the infrastructure. The server 1 runs the software traffic generator that simulates clients for the vCPE Platform. Both servers are connected using 2 pairs of NICs, simulating the LAN and WAN interfaces of the vCPE platform.

3.2.6.5 PoC KPIs

We use the following KPIs to quantify the level of conformance of the PoC to the NECOS platform requirements:

- Slice throughput, in Mbps, and latency, in seconds, as an indication of performance for the vCPE scenario. The vCPE user experience is affected by these parameters and reflect the quality of the offered service;
- Slice Cache related metrics as indicators of the slice isolation level. These measures are defined in terms of cache usage number, in MB, and cache misses per second.

3.2.7 PoC07: WIreless Slicing sErvices (WISE)

WISE is a promising approach to improve 5G Ultra-dense networking (UDN), turning the WiFi WLAN-sharing technology into a fully softwarized slice-defined architecture. Through WISE-enabled WiFi WLAN-sharing systems, 5G networks are foreseen to complement cellular networks with broadband wireless access, and enable service-oriented WLANs to provide a myriad of differentiated services to hosts and things with high-level isolated and customized end-to-end slices (called deep slicing capability).

This PoC exercises the “NECOSization” of the WISE approach to leverage advanced capabilities (eg., elasticity, federation and mobility) that seek to improve broadband wireless access over 5G UDN scenarios. For evaluation purposes, the PoC verifies how beneficial WISE becomes over FON[21], the *de facto* WiFi WLAN-shared system, through the NECOS-supported slicing capabilities.

[21]FON is the name of world’s largest community WiFi network where its members share their broadband connection in exchange for broadband access to other FON access points throughout the world.

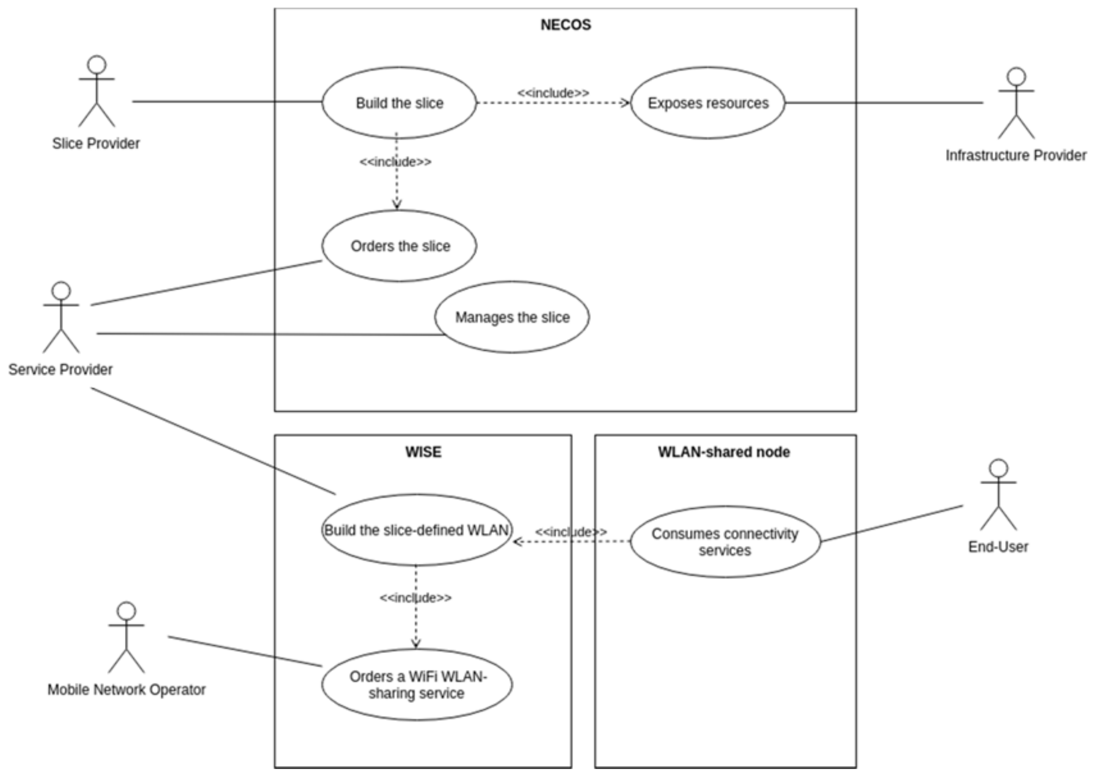

Figure 22. PoC07 Use case diagram

3.2.7.1 *Software requirements*

- **[Req P07_01]** Cloudification: The infrastructure MUST provide cloudification solutions like OpenStack, Docker, Cloudfy, and TOSCA;
- **[Req P07_02]** Switch software: The infrastructure MUST provide switch software that include support for OpenFlow protocol, like Open vSwitch;
- **[Req P07_03]** SDN controller software: The infrastructure MUST provide SDN controller software that include support for OpenFlow protocol, like OpenDaylight;
- **[Req P07_04]** Monitoring software: The infrastructure MUST provide system monitoring software like Elasticsearch[22], Logstash[23], and Kibana[24] to perform the slice monitoring;
- **[Req P07_05]** Operating System Capabilities: The following Operating System Capabilities MUST be included:
  - CentOS 7;
  - Kernel version >= 3.10.0-862.11.6.el7.x86_64.

3.2.7.2 *Hardware requirements*

- **[Req P07_06]** Computing resources: The infrastructure MUST provide off the shelf computer nodes with support for logical partitioning technologies (i.e., virtual machines and containers);

[22] https://www.elastic.co/products/elasticsearch

[23] https://www.elastic.co/products/logstash

[24] https://www.elastic.co/products/kibana

- **[Req P07_07]** Networking resources: The infrastructure MUST provide SDN switches for fog-to-cloud connectivity and wireless routers with OpenWrt enabled for WLAN connectivity.

3.2.7.3 LSDC requirements to be validated

This PoC addresses 4 requirements from the 5G Services scenario, namely the RF.vRAN.1, RF.vRAN.2, RF.vRAN.3, and RN.vRAN.1 requirements, which are used to validate 3 LSDC key characteristics (i.e., KC1, KC2 and KC3), as presented in **Figure 23**.

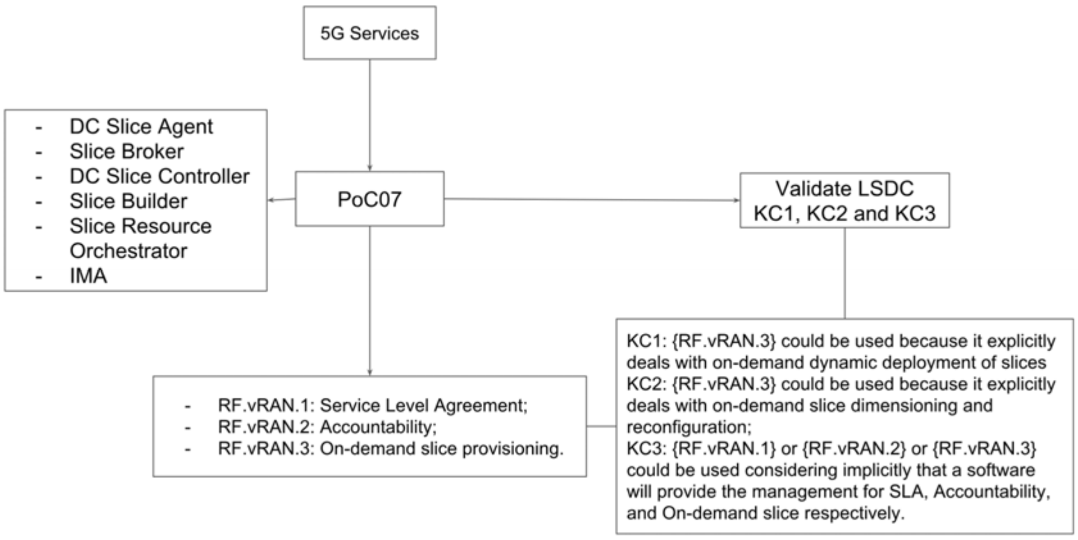
5G Services
- DC Slice Agent
- Slice Broker
- DC Slice Controller
- Slice Builder
- Slice Resource Orchestrator
- IMA
PoC07
Validate LSDC KC1, KC2 and KC3
- RF.vRAN.1: Service Level Agreement;
- RF.vRAN.2: Accountability;
- RF.vRAN.3: On-demand slice provisioning.
KC1: {RF.vRAN.3} could be used because it explicitly deals with on-demand dynamic deployment of slices
KC2: {RF.vRAN.3} could be used because it explicitly deals with on-demand slice dimensioning and reconfiguration;
KC3: {RF.vRAN.1} or {RF.vRAN.2} or {RF.vRAN.3} could be used considering implicitly that a software will provide the management for SLA, Accountability, and On-demand slice respectively.

Figure 23. PoC07 relationships to NECOS scenarios, elements, characteristics, and requirements

3.2.7.4 Validation infrastructure

As depicted in **Figure 24**, the PoC uses the following testbed for its evaluation:

- A WIFi-slicing domain where different customers (e.g., a set of IoT devices communicating with each other over Message Queuing Telemetry Transport (MQTT) protocol, an UDP video streaming, or a mobile application generating TCP traffic) transmit wireless data;
- A Fog-slicing domain, where off-the-shelf fog nodes provide services beyond typical WiFi WLAN, such as attachment/connectivity for mobile devices, locally-deployed custom procedures encapsulated as slice abstractions. The fog node provisions different virtual WLANs (V-WLANs), each associated to particular slice(s). In this way, packets coming from different V-WLANs are subject of processing for the corresponding slice before leaving the network;
- A Network-slicing domain, where multiple virtual networking infrastructures (e.g., nodes, links, interfaces, etc.) are built on top of an SDN-capable backhaul, providing proper fog-to-cloud slicing connectivity;
- A Cloud-slicing domain, where in addition to the WLAN domain, the required computing and storage resources are dynamically instantiated and configured. Moreover, this domain embeds the entire NECOS slice management and orchestration approach.

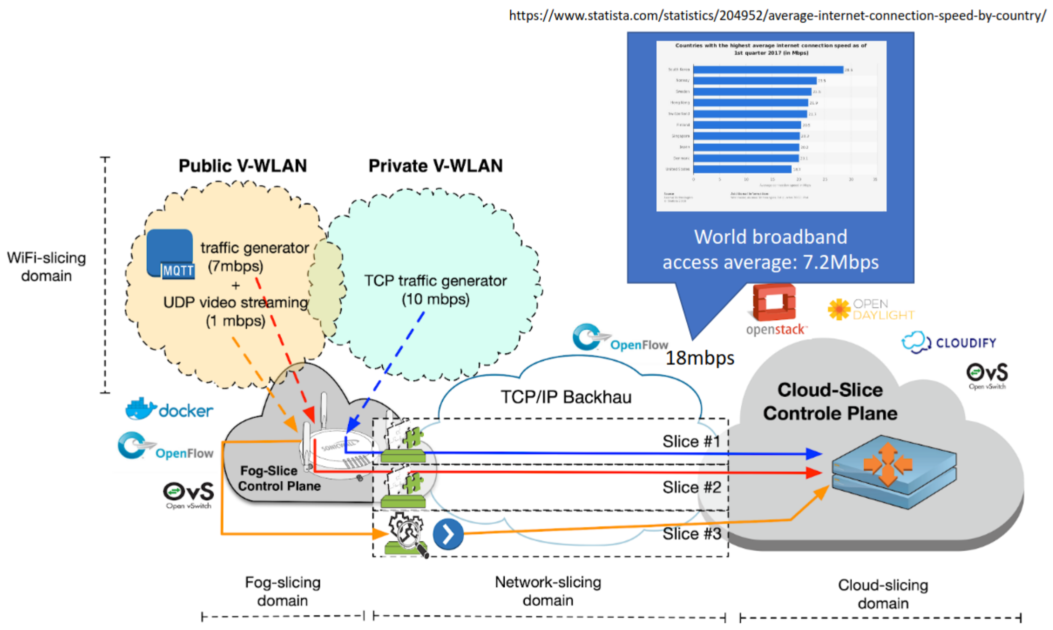
https://www.statista.com/statistics/204952/average-internet-connection-speed-by-country/
World broadband access average: 7.2Mbps
Public V-WLAN
Private V-WLAN
WiFi-slicing domain
traffic generator (7mbps) + UDP video streaming (1 mbps)
MQTT
TCP traffic generator (10 mbps)
docker
OpenFlow
OvS
Fog-Slice Control Plane
openstack
OPEN DAYLIGHT
CLOUDIFY
18mbps
TCP/IP Backhau
Cloud-Slice Controle Plane
Slice #1
Slice #2
Slice #3
Fog-slicing domain
Network-slicing domain
Cloud-slicing domain

Figure 24. PoC07 target validation infrastructure

The validation infrastructure is composed by the following hardware:

- Two Dell PowerEdge R740 servers hosting the testbed software;
- Two low-capacity nodes (i.e., 2 Raspberry PIs);
- Two medium-capacity nodes (i.e., 2 laptops);
- Four MikroTik RouterBoard 951G-2HnD.

3.2.7.5 PoC KPIs

We define the following KPIS in order to measure how the NECOS platform requirements are being met in the PoC:

- Slice performance (e.g., slice deployment time in seconds, slice decommissioning time in seconds);
- Physical resources utilization percentage:
  - CPU: Percentage utilization in seconds;
  - Memory: Utilization in MB;
  - Link: bandwidth utilization in Mbps.
- Service performance (e.g., throughput effect from slicing over classical WLAN-shared in Mbps and delay effect from slicing over classical WLAN-shared in seconds).

4 The NECOS platform over experimentation testbed environments

This section provides an introduction of main testbed environments in scope of the NECOS project, namely FIBRE, Emulab, and 5Tonic While all of them meet the high-level experimentation requirements defined in subsection 3.1, e.g., support for virtualization, bare metal, and network programmability, the best suitability of each PoC to the experimentation environment differs, as covered by the PoC-specific requirements as well as familiarity and accessibility to the testbed environment instantiations.

From the PoCs described in Section 3, four of them require a local infrastructure deployment (i.e., PoC03, PoC05, PoC06 and PoC07) and the remaining three (i.e., PoC01, PoC02 and PoC04) use well-known experimentation testbeds that also allow the replication of the tests performed. Next, we describe the FIBRE and EMULAB testbed environments currently used, and we also describe the 5TONIC testbed that is a testbed target for the second year of the project.

4.1 FIBRE

The FIBRE testbed is research facility constructed in the scope of project funded by the Brazil-EU Coordinated Call in ICT. The FIBRE infrastructure consists of a federation of 11 local testbeds, also called islands or experimentation nodes.

The FIBRE is composed of islands that each have a set of network devices to support experiments in both fixed and wireless technologies. These network resources are connected to an overlay network on the RNP backbone that is comprised of two network separate layers: the first is a control plane, and the second an experimentation plane.

The control plane is responsible for the control communications through a main central node, where the FIBRE is located, and the experimentation plane for the data communication between the network resources used by an experimenter, in the FIBRE facility. A typical FIBRE island is illustrated in **Figure 25**.

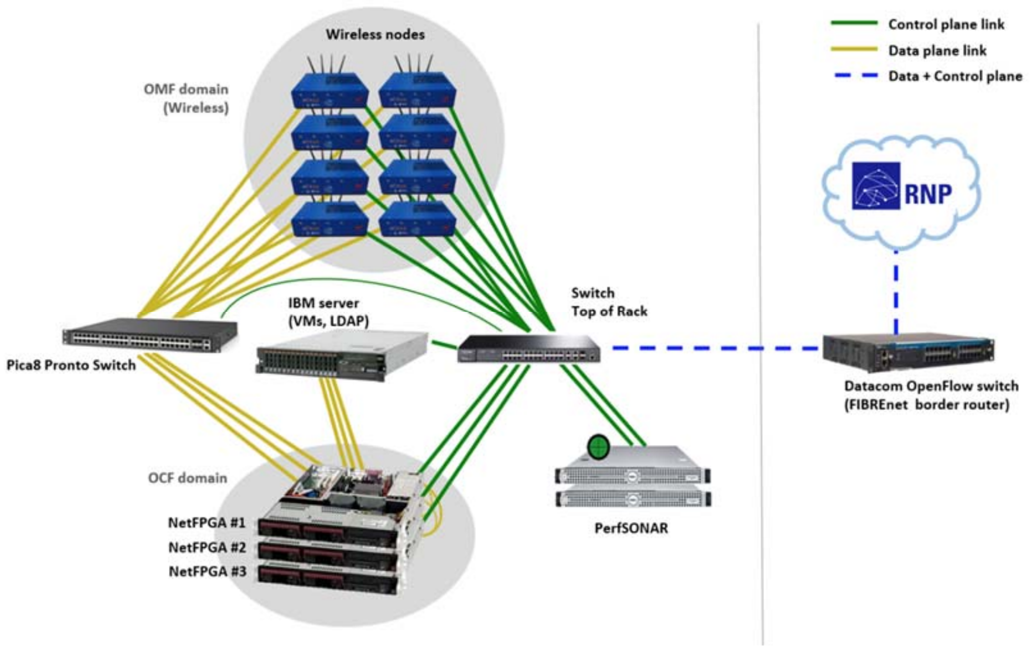

Figure 25. Typical FIBRE island

The FIBRE uses the Control Monitoring Framework[25] (OMF). It was developed in the Ruby language and based on the eXtensible Messaging and Presence Protocol (XMPP), with the focus on controlling and managing network devices. The OMF architecture consists of 3 logical planes:

- Control plane: includes the OMF tools that a researcher uses to describe his/her experiments;
- Measurement plane: includes the OMF tools to instrument an experiment; and
- Management plane: includes the OMF functions and entities to provision and configure the resources.

The FIBRE testbed has been of fundamental importance for conducting our experiments, in particular, those related to the development of NECOS architecture components, such as DC Slice Controller. The WP7 deliverables present contributions that the NECOS project brought to the FIBRE testbed.

4.1.1 LSDC testbeds requirements

The PoCs requirements from section 3, that use FIBRE, are met using the following FIBRE's elements:

- Hardware requirements:
    - The Icarus nodes perform the remote boot, which allows the implementation of bare metal resources and the wireless access points. This device meets the requirements [Req P01_07] and [Req P04_05];
    - The switch Pronto Pica 08, present in the FIBRE testbed, fulfills the requirement [Req P04_04**].**
- Software requirements:

[25] https://www.fed4fire.eu/tools/omf6/

- The CMC service must be used to boot the Icarus node, fulfilling the requirement [Req P01_04];
- The Resource Controller from Frisbee must be used for save the image in Icarus node, in accordance to requirement [Req P01_04].

4.2 Emulab

Emulab is a flexible environment for real experimentation which keeps the simplicity of simulation. It offers time- and space-shared experimentation, e.g., sharing network and physical machines between experimenters. A large number of PCs in racks, combined with user-friendly web-based tools, and driven by scripting languages (e.g., ns-compatible scripts) or GUIs, allow the experimenters to remotely configure and control machines and links down to the hardware level. It also supports emulation of user-defined packet loss, latency, bandwidth, queue sizes. The allocated PCs automatically install custom experimenter images. The primary Emulab installation[26] and the maintenance of the Emulab code is being carried out by the Flux Group, School of Computing, University of Utah. There are tens of other installation throughout the globe, ranging from handful up-to hundreds of nodes. Emulab is widely used by networking and distributed systems researchers and also supports education, i.e., is used to teach classes in the same fields.

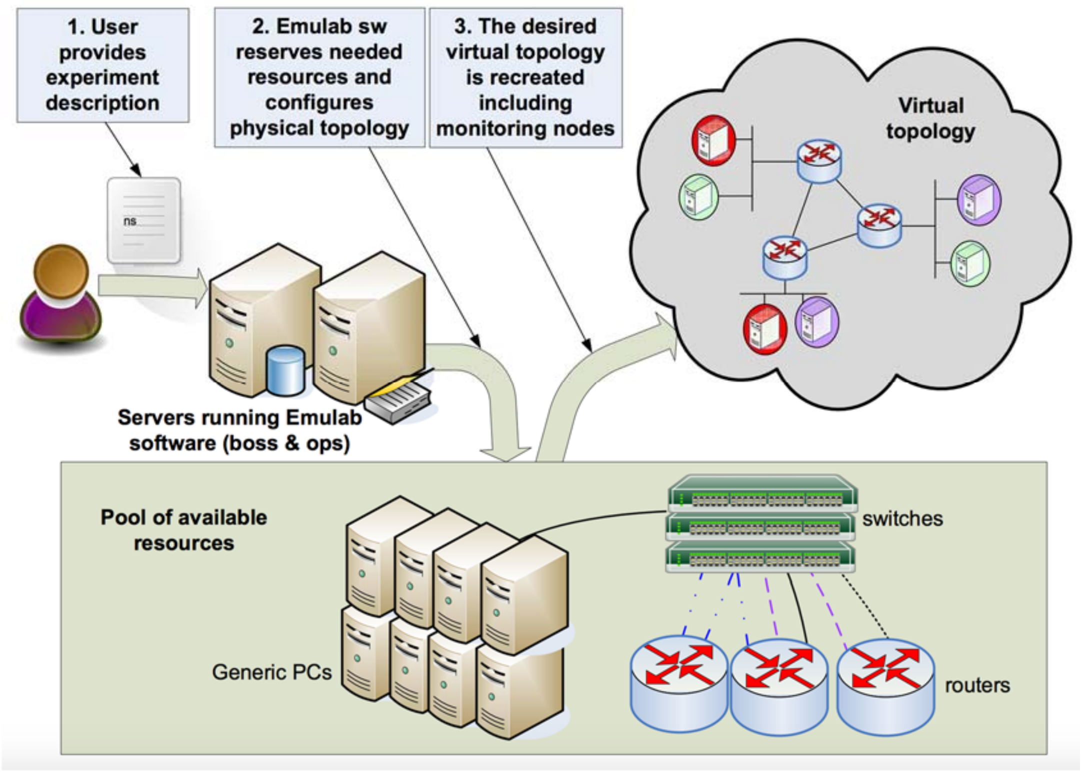
1. User provides experiment description
2. Emulab sw reserves needed resources and configures physical topology
3. The desired virtual topology is recreated including monitoring nodes
Virtual topology
ns
Servers running Emulab software (boss & ops)
Pool of available resources
Generic PCs
switches
routers

Figure 26. Basic Emulab Architecture [Siaterlis 2013]

In **Figure 26**, we show the basic Emulab architecture. The experimenter describes his/her own requirements based on a scripting language, usually in TCL or Python. Such description includes the

[26] http://www.emulab.net

required physical resources, network configuration and hard disk images to install. Two physical servers (i.e., called boss & ops) are running the Emulab software and other required server facilities, e.g., frisbee - the software installing custom images to the physical machines through multicasting, a DNS server, a file server, an e-mail server, etc. The Emulab software installed in boss/ops realizes the experiment through: (i) reserving and allocating the requested physical resources, for a particular time-frame; and (ii) configuring the allocated physical servers and network resources, i.e., creating user accounts, emulating network conditions, deploying VLANs for the isolation between experimenters, etc.

University of Macedonia (UOM) maintains a local Emulab installation[27] which is being used in the NECOS project. UOM is in the process to integrate such testbed with a global federation of relevant testbeds called FED4FIRE[28], in the context of UNIC, an open-call project of the FED4FIRE+ H2020 research project. The integrated testbed will be used in a NECOS demonstrator, as a means to realize large-scale and multi-domain slice operation.

4.2.1 LSDC testbed requirements

The UOM Emulab testbed satisfies the requirements for the PoC02: Content Distribution Service over the NECOS platform, including:

- Software requirements: The Slice monitoring tools (Req P02_01), Lightweight cloud technologies (Req P02_02) and the Operating System Capabilities (Req P02_03) are being supported through custom Emulab images.
- Hardware requirements: The Emulab testbed supports all the required hardware facilities, such as: (i) Layer-3 Switches for experimentation and control networks (Req P02_04); and (ii) server resources for edge and core cloud facilities (Req P02_05).

4.3 5tonic

The 5G Telefónica Open Network Innovation Center lab (5TONIC, https://www.5tonic.org/) was created in 2015, by Telefónica I+D (i.e., the research branch of Telefónica) and IMDEA Networks Institute. The creation of this open lab was fostered with a clear vision of setting up an open research and innovation laboratory ecosystem in which industry and academia could come together to boost technology and business innovative experiments and exploratory initiatives. Since then, several leading companies have become members and collaborators of 5TONIC, with a current roster of 10 members and 5 collaborators, as shown in Table 3.

Table 3. 5TONIC Members and Collaborators

Members	Collaborators
• Telefónica	• IFEMA

[27] http://emulab.swn.uom.gr
[28] http://www.fed4fire.eu

• IMDEA Networks • Ericsson • University Carlos III of Madrid • Intel • CommScope • Altran • Cohere Technologies • InterDigital • Red Hat	• ASTI Robotics • Rohde & Schwarz • Luz Wavelabs • Saguna Networks

The 5TONIC lab is on the forefront of technological innovation, showing an extensive track record in past and present European 5G Research Projects. The list of projects using or having used 5TONIC facilities can be checked at https://www.5tonic.org/research/5g-european-projects. It is relevant to highlight that the 5TONIC premises are part of the new 5G end-to-end facilities being funded by the European Commision through the projects 5G-EVE[29] and 5G-VINNI[30]. In addition to that, it is also remarkable as well that other Europe-Brazil innovation initiatives make use of 5TONIC, such as 5G-INFIRE[31] and 5G-RANGE[32].

The 5TONIC site is located at IMDEA Networks premises in Leganés, Madrid area, Spain, but it has access to other locations for the support of different network functions and use cases:

- University Carlos III of Madrid (UC3M) campus both at Leganés and Madrid City Center;
- Telefónica I+D Future Network lab at Almagro Central Office, in Madrid;
- Telefónica headquarters campus Distrito C, in Madrid;
- 5G IFEMA Lab at Feria de Madrid;
- Connection with Telefónica Spain lab at Alcobendas, Madrid area.

Due to its collaborative nature, 5TONIC is not a conventional site and in terms of available infrastructure. In this sense, 5TONIC infrastructure provided by Telefónica (in the different locations provided by Telefónica), as well as common infrastructure and services (e.g., connectivity), is the one that will be available for NECOS. This fact does not prevent that access to these network elements for 5G EVE activities could be agreed with other 5TONIC members and collaborators in a case by case basis, to be explored in the future. This also applies to the establishment of cross-collaboration with other projects using the same facilities.

Permanent infrastructure that is foreseen to be used in the project includes data center infrastructure (with rack space for hosting equipment) and communications infrastructure. In 5TONIC premises Telefonica has at this stage different equipment available, including hybrid OpenFlow switches from

[29] https://www.5g-eve.eu/

[30] https://www.5g-vinni.eu/

[31] https://5ginfire.eu/

[32] http://5g-range.eu/

HP, optical equipment from Huawei, and COTS servers from Dell. This is complemented by the Future Network lab equipment which includes IP/MPLS routers from Juniper and Nokia, and optical equipment from Adva and Huawei.

5 Conclusions

This document organizes and collects the requirements to perform the experimental activities within the NECOS project. The practical validation follows proofs-of-concept approach. Currently, 7 PoCs are undergoing, as described in this deliverable along the requirements of libraries and operating systems capabilities needed. Also, the requirements of public IP addresses to provide interconnection between NECOS components hosted in different domains to enable cloud federation experiments were presented.

In order to provide the validation of the NECOS platform, Table 1 was presented showing the relationship between the LSDC key characteristics and the requirements from the scenarios. Those relationships were discussed in Section 2 and which were correlated in Section 3 with the PoCs.

As a wrap-up, **Figure 27** presents how the seven PoC contribute to the validation of specific LSDC key characteristics. All the four characteristics will be validated by at least one of the PoC, showing the complementarity of the PoCs proposed in this deliverable.

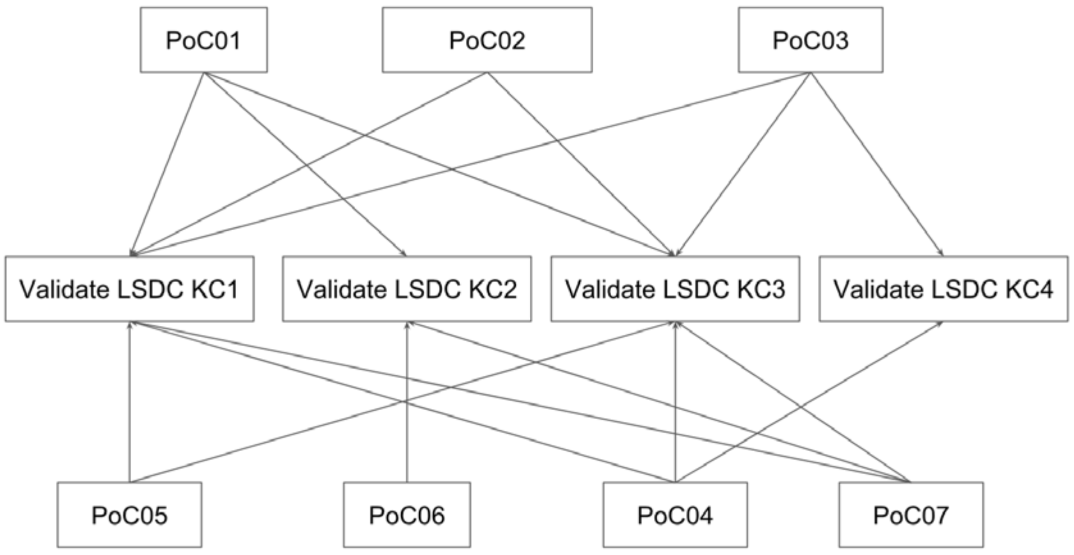

Figure 27. PoCs' validation relationship with the LSDC Key characteristics

The last contribution of this deliverable was a description of the testbeds and how the requirements of the PoCs are met by the specific testbeds. It's important to note that some of the PoC will use a local testbed deployed by NECOS members while others will use testbeds already well known by the community, providing a heterogeneous yet rich test environment to validate NECOS from different perspectives.

References

[Bradner 1997] S. Bradner, “Key words for use in RFCs to Indicate Requirement Levels”, RFC 2119, March 1997.

[D2.1] Deliverable D2.1: Initial definition of use cases, 2018.

[D3.1] Deliverable D3.1: NECOS System Architecture and Platform Specification. V1, 2018.

[D4.1] Deliverable D4.1: Provisional API and Information Model Specification, 2018.

[D5.1] Deliverable D5.1: Architectural update, Monitoring and Control Polices Frameworks, 2018.

[M6.1] Milestone M6.1: Report of Software Tools for Automated Infrastructure Deployment, 2018.

[Mouradian 2018] Mouradian, Carla et al. “A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges.” IEEE Communications Surveys & Tutorials 20 (2018): 416-464.

[Siaterlis 2013] Siaterlis, C., Garcia, A. P., & Genge, B. (2013). On the use of Emulab testbeds for scientifically rigorous experiments. IEEE Communications Surveys & Tutorials, 15(2), 929-942.

Version History

Version	Date	Author	Change record
0.1	06/06/2018	UFPA	Draft ToC
0.2	29/06/2018	UFPA, UPC	Draft ToC amended
0.3	22/08/2018	UFPA	Merging the M6.2 with the D6.1
0.4	01/10/2018	UCL, UFG, UOM, UNICAMP, UFPA, UFU, CPQD, UFRN	Updated with the last PoCs
0.5	17/10/2018	UOM	Internal review from Lefteris Mamatas
0.6	18/10/2018	UNICAMP	Internal review from Christian Rothenberg
0.7	19/10/2018	UFRN	Internal review from Marcilio Lemos
1.0	22/10/2018	UFPA	Consolidated final version

Appendix A

Template proposed for the description of NECOS POCs

For each NECOS scenarios, proposed in D2.1, several Proofs-of-concept (POC) can be realized. The following template is proposed to assist the description of the POCs.

PoC Name	Name
Scenario	
Goal	
Use Case Diagram	
Description of the Use case Diagram	
Inputs	
Outputs	
Actors	
Requirements addressed	
Testbed	
Hardware Infrastructure	
Software Infrastructure	
LSDC Components	
PoC Client	
Actor Actions	
Slice mode	
PoC KPIs	

The fields in the table refer to:

- PoC Name: Name for the PoC.
- Scenario: Title of the scenario.
- Goal: brief description of the PoC goal (what it tries to address).
- Use Case Diagram: Provide the Use case diagram for the PoC. You should put the Actors (already defined), the activities performed by LSDC (try to correlate it with the requirements already defined).
- Inputs: Information to be inserted by the PoC client
- Outputs: Information to be produced by the PoC.

- Actors: list of actors of the PoC
- Requirements addressed: list of requirements to be addressed by the PoC.
- Testbed: Will you use a testbed to perform the PoC? Which one? Why this one? Does this testbed have something special needed by the PoC?
- Hardware Infrastructure: What hardware infrastructure do you need to perform the PoC? (Number and type of computational devices needed).
- Software Infrastructure: What software infrastructure do you need to perform the PoC? Do you need a specific OS? Which one? (This infrastructure software are not part of LSDC, it's needed to deploy and use LSDC).
- LSDC Components: Which LSDC components will be used in the PoC? Which functions will be implemented in each component?
- PoC Client: Does the PoC need a client? How will this client be emulated? How will it consume the service provided in the PoC? (Ex: if you are providing a video service over a slice, the client will be some VLC user or some program emulating many VLC users.)
- Actor Actions: Which action will be performed by each actor in the PoC? (Try to be more specific, avoiding the tenant term and defining the specific tenant role like network provider or slice provider)
- Slice mode: Which slice mode is used in the PoC?
- PoC KPIs: What are the concrete KPIs or criteria for measuring how the main goals and specific requirements are met?

Appendix B

As it was presented in our last face to face meeting at Thessaloniki, we need to define the requirements of D2.1 in order to validate all the five LSDC key characteristics; thus the UFPA team has made a questionnaire.

Since we don't know if a single requirement is enough to validate one LSDC key characteristic, we ask for a minimal set of requirements to validate each singular characteristic.

The questionnaire should be applied to each scenario. For each scenario four questions arise, directly related to each LSDC key characteristic. The four key characteristics of LSDC are as follows:

1) It empowers a new service model – the **Slice as a Service**, by dynamically mapping service components to a slice. The enhanced management for the infrastructure creates slices on demand and slice management takes over the control of all the service components, virtualized network functions and system programmability functions assigned to the slice, and (re)configure them as appropriate to provide the end-to-end service.

2) It enables **easy reconfiguration and adaptation of logical resources** in a cloud networking infrastructure, to better accommodate the QoS demand of the Slice, through using software that can describe and manage various aspects that comprise the cloud environment.

3) It allows each aspect of the cloud environment – from the networking between virtual machines to the SLAs of the hosted applications – to be **managed via software**. This reduces the complexity related to configuring and operating the infrastructure, which in turn eases the management of the cloud infrastructure.

4) The LSDC platform will offer the ability to a specific cloud provider to **federate his own infrastructure with other cloud providers** with different configurations in order to realize virtualized services through the use of the Slice as a Service concept. The users of the LSDC APIs and platform will be able to create virtual services that can span the merged cloud infrastructure offered by different cloud providers. This concept is not purely technical, it can also encompass business, cultural, geographical or in any other domain.

KEY CHARACTERISTIC MAPPING QUESTIONNAIRE
1 - What is the minimal set of requirements to validate the key characteristic "**new service model - the Slice as a Service"** of LSDC?
[**{}** **]**

2 - What is the minimal set of requirements to validate the key characteristic "**easy reconfiguration and adaptation of logical resources in a cloud networking infrastructure"** of LSDC?
[**{}** **]**
3 - What is the minimal set of requirements to validate the key characteristic "**managed via software"** of LSDC?
[**{}** **]**
4 - What is the minimal set of requirements to validate the key characteristic "**federate his own infrastructure with other cloud providers"** of LSDC?
[**{}** **]**