Deliverable 3.2

Publication of first version of service catalogues to vertical consumers

Editor:	Christos Tranoris, University of Patras
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	30th June 2019
Actual delivery date:	9th July 2019
Suggested readers:	ICT-19 projects, industry verticals seeking to conduct Experimentation on 5G systems
Version:	1.3
Total number of pages:	102
Keywords:	5G, Service specification, Service catalogue

Abstract

This deliverable contains details of implementation towards the first operational readiness of 5G-VINNI facilities for provisioning network slices as services for industry verticals. The deliverable contains details about how each facility: i) supports the different envisaged 5G-VINNI roles, ii) what are the services and entry points for the supported roles, iii) how these roles access the facilities services and catalogues, iv) how each facility supports the Lifecycle Management of Network Slice Services, v) what are the interfaces exposed to be used by the service orchestration layer, vi) how each facility will support day-to-day operations and issue management for verticals and vii) which are the initial internal tests planned by each facility to verify service operations

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain 5G-VINNI consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU): All 5G-VINNI consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP): All 5G-VINNI consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE): All 5G-VINNI consortium parties have agreed to full publication of this document. However this document is written for being used by <organisation / other project / company etc.> as <a contribution to standardisation / material for consideration in product development etc.>.

In case of Consortium confidential (CO): The information contained in this document is the proprietary confidential information of the 5G-VINNI consortium and may not be disclosed except in accordance with the consortium agreement.

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

Neither the 5G-VINNI consortium as a whole, nor a certain part of the 5G-VINNI consortium, warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, accepting no liability for loss or damage suffered by any person using this information.

The EC flag in this document is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-VINNI receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815279.

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP3: 5G VINNI End-to-End Facility Readiness and Operation
Number and title of task(s)	T3.2: Service orchestration readiness for each E2E facility
Document title	Publication of first version of service catalogues to vertical consumers
Editor: Name, company	Christos Tranoris, University of Patras
Work-package leader: Name, company	Sami Kaavisari, Nokia

Copyright notice

Executive summary

The implementation and deployment of the 5G-VINNI facility includes a phase where the operational readiness is defined, agreed and realized across all facility sites. The focus herein is on provisioning network slices as services for industry verticals.

This document should be consulted by architects, network engineers and other experts that implement and deploy a 5G experimentation facility with the objective to be compliant with 5G-VINNI. In particular this document will help understanding how each 5G VINNI facility site deals with the following characteristics:

- Identification of roles for each facility site and how they are supported
- Definition of the services and entry points for the supported roles
- Definition of the processes on how each role accesses the facility services and catalogues
- Description on how each facility site supports the Lifecycle Management of Network Slice Services
- Description of the Northbound interfaces exposed for use by the service orchestration layer and later by the testing services
- Description on the processes and services of how each facility site will support infrastructure operations and issue management for verticals
- Which are the internal tests planned or executed by each facility site to verify service operations?

The operational readiness of the facility is covering the following aspects:

- Support for the actor roles defined by 5G-VINNI and which are in alignment of the 3GPP actor role model.
- Identification of the interaction model and definition of the supporting services for each actor role that interacts with the 5G-VINNI facility
- Identification of the entry point for interaction with the facility and the facility sites.
- Implementation of all configuration that is necessary in the different orchestration layers at the facility sites for 3GPP SA5 Network Slice Instance(s) life-cycle management use cases:
    - Definition and on-boarding of VNF Descriptors (VNFD), Network Service Descriptors (NSD) to the different service catalogues
    - Establishing orchestration flows for service creation, activation, modification, deactivation and decommissioning
    - Configuring service repository data models and configuration data
- Technical pre-testing of the orchestration flows
- Publishing/exposing slice service catalogues through the service orchestration layer to service consumers such as the test platform and other verticals

This document is the second document in a series of documents that describe the technical and governance aspects of the operationalisation of the 5G-VINNI facility. The first document provides *Specification of services delivered by each of the 5G-VINNI facilities (D3.1).* Future documents will cover *Publication of service catalogues including E2E services across multiple operator domains, Publication of service specification governance model,* as well as *service orchestration and life-cycle management.*

List of authors

Company	Author
UoP	Christos Tranoris
Fraunhofer	Corici Marius-Iulian , Briedigkeit Thomas, Chowdhury Ananya, Szabó Zsolt
BT	Paul Muschamp
Samsung	Carl Williams
Altice Labs	Carlos Parada, José Bonnet, Jorge Carapinha
TID	Jose Ordonez-Lucena, Sonia Fernández
Telenor	Andres Gonzales, Pål Grønsund, Min Xie
Nokia	Rodrigues, Joao A., Ghosh, Tirthankar
UC3M	Carmen Guerrero, Adrián Gallego
HWDU	Artur Hecker, Xun Xiao , Osama Abboud

Table of Contents

List of figures

List of tables

Abbreviations

3GPP	3rd Generation Partnership Project
5G	Fifth Generation (mobile/cellular networks)
5G-PPP	5G Public Private Partnership
5GC	5G Core
5QI	5G Quality Indicator
AGV	Automated Guided Vehicle
API	Application Programming Interface
AR	Augmented Reality
B2B	Business to Business
B2B2X	Business to Business to Everything
B2C	Business to Consumer
B2H	Business to Household
BBF	Broadband Forum
BSS	Business Support Systems
CI/CD	Continuous Integration / Continuous Development
CN	Core Network
CSC	Communication Service Customer
CSP	Communication Service Provider
CU	Central Unit
DL	Downlink
DU	Distributed Unit
E2E	End-to-End
eMBB	Enhanced Mobile Broadband
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
GST	Global Slice Template
HD	High Definition
IETF	Internet Engineering Task Force

IMS	IP Multimedia Subsystem
KPI	Key Performance Indicator
L2VPN	Layer 2 Virtual Private Network
L3VPN	Layer 3 Virtual Private Network
MANO	Management and Orchestration
MEF	Metro Ethernet Forum
mIoT	Massive Internet of Things
mMTC	Massive Machine-Type Communications
MVNO	Mobile Virtual Network Operator
NaaS	Network as a Service
NBI	Northbound Interface
NEST	Network Slicing Task Force
NETCONF	Network Configuration Protocol
NFV	Network Functions Virtualisation
NFV-NS	NFV Network Service
NFVIaaS	NFVI as a Service
NGMN	Next Generation Mobile Networks
NOP	Network Operator
NR	New Radio
NSA	Non standalone
NSaaS	Network Slice as a Service
NSD	Network Service Descriptor
NSI	Network Slice Instance
NSSAI	Network Slice Selection Assistance Information
NST	Network Slice Template
ODA	Open Digital Architecture
ONAP	Open Network Automation Platform
OSM	Open Source MANO
OSS	Operation Support Systems

PPDR	Public Protection and Disaster Relief
QoS	Quality of Service
RAN	Radio Access Network
RRH	Radio Remote Header
SA	Standalone
SC	Service Component
SDO	Standards Development Organisation
SLA	Service Level Agreement
SRTP	Secure Real-time Transport Protocol
SST	Slice Service Type
TaaS	Testing as a Service
TCP	Transmission Control Protocol
TMForum	Tele Management Forum
TOSCA	Topology and Orchestration Specification for Cloud Applications
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UHD	Ultra High Definition
UL	Uplink
uRLLC	Ultra Reliable Low Latency Communications
V2X	Vehicle to Everything
VNF	Virtualised Network Function
VNFaaS	VNF as a Service
VNFD	Virtualised Network Function Descriptor
VPN	Virtual Private Network
VR	Virtual Reality
VSB	Virtual Service Blueprint
ZOOM	Zero-touch Orchestration, Operations and Management
ZSM	Zero-touch Network and Service Management

1 Introduction

This deliverable contains details of implementation towards the first operational readiness of the 5G-VINNI facility sites for provisioning network slice as services for industry verticals. The work performed includes:

- Implementation of all configuration that is necessary in the different orchestration layers at the local facility sites for 3GPP SA5 Network Slice Instance(s) life-cycle management use cases:
    - Definition and on-boarding Virtual Network Function (VNF) and Descriptors (VNFD), Network Service Descriptor (NSD) to the different service catalogues
    - Establishing orchestration flows for service creation, activation, modification, deactivation and decommissioning
    - Configuring service repository data models and configuration data
- Technical pre-testing of the orchestration flows
- Publishing/exposing network slice service catalogues through the service orchestration layer to service consumers such as the test platform and other verticals

The deliverable contains details about how each 5G VINNI facility sites address the following aspects:

i) Identification of specific roles for each facility site and how they are supported
ii) Definition of the services and entry points for the supported roles
iii) Definition of the processes and entry points on how each role accesses the facility site services and catalogues
iv) Description on how each facility site supports the Lifecycle Management of Network Slice Services
v) Description of the interfaces exposed to be used by the service orchestration layer and later by the testing services
vi) Description on the processes and services of how each facility site will support day-to-day operations and issue management for verticals and
vii) Which are the initial internal tests planned by each facility site to verify service operations

The document is structured as follows:

- Chapter 2 presents how each 5G VINNI facility site supports 3GPP roles beyond those that are customer-facing
- Chapter 3 describes the 5G-VINNI service catalogues of each facility site as well as how each facility sites serves the supported roles via different service entry interfaces
- Chapter 4 describes facility site approaches on Network Slice Service management, which models and specifications are supported by their deployed services and tools and how the provisioning of slices will be performed by the 5G-VINNI experimenters
- Chapter 5 describes how the facility sites provide access to their customers for accessing the catalogue management and the requests to deploy artifacts
- Chapter 6 describes facility sites approaches for supporting the operations and issue management

A set of Annexes are provided with descriptions of the roles and technical pre-testing.

2 Supported Roles of 5G-VINNI facilities

2.1 Defined Roles

As presented in 5G-VINNI deliverable 3.1 “Specification of Services Delivered by each of the 5G-VINNI facilities” [D3.1] and shown in Figure 2-1, the 3GPP roles in 5G-VINNI that are relevant from the customer-facing perspective are Communication Service Provider (CSP) and Communication Service Customer (CSC). 5G-VINNI facility takes the role of CSP, while a 5G-VINNI customer behave as CSC. The role of Network Operator (NOP) is taken by the operator of each 5G-VINNI facility site, in charge of providing the communication capabilities required to support the network slice services. This includes the design, deployment and operation of network slice services at the E2E infrastructure, and hence falls within the resource-facing perspective.

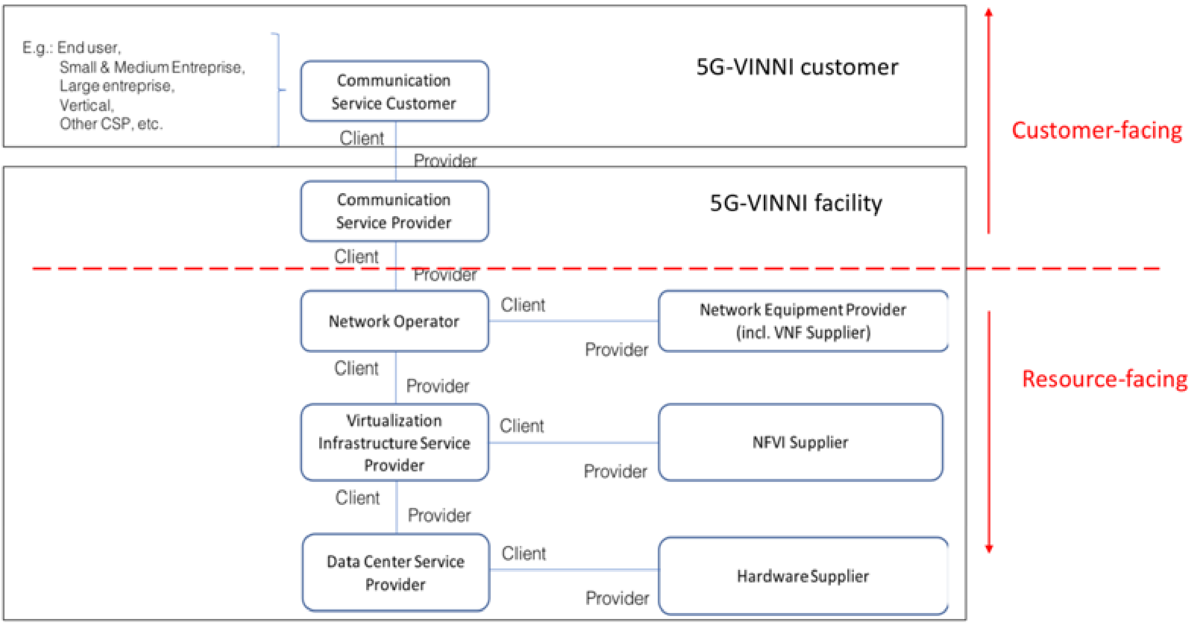


Figure 2-1 3GPP roles and their impact in the resource-facing and customer-facing perspectives

The services that are relevant for 5G-VINNI are the following:

- network slice services -> *a network slice delivered by CSP to a CSC as a communication service, under the NSaaS delivery model.* Network slice service is the customer-facing service view of a network slice
- network service -> *what NOP provides to a CSP to build up a network slice.* Network service is (part of) the resource-facing service view of a network slice.

2.2 Roles supported per 5G-VINNI facility

The tables below provide a high-level description of how each 5G-VINNI facility site supports the 3GPP roles beyond the customer-facing roles CSP and CSC. Details of each role can be found at the corresponding facility Annex.

Table 2-1 Communication Service Customer

	Communication Service Customer
Norway	For all facilities, industry verticals are considered as CSC and consume the network slice services. In the case of 5G-VINNI facilities the selected service delivery model is NSaaS and the deliveries are network slice services (instances).
UK	
Greece	
Spain	

Portugal	
Germany, Munich	
Germany Berlin/Luxemburg	

Table 2-2 Communication Service Provider

	Communication Service Provider
Norway	Telenor
UK	BT
Greece	Network Architectures and Management group, UoP
Spain	5TONIC
Portugal	Altice Labs
Germany, Munich	Huawei
Germany Berlin/Luxemburg	5G-Playground Berlin

Table 2-3 Network Operator

	Network Operator
Norway	Telenor
UK	BT
Greece	Network Architectures and Management group, UoP
Spain	TID supported by UC3M
Portugal	Altice Labs
Germany, Munich	Huawei
Germany Berlin/Luxemburg	SES in Luxemburg and Fraunhofer FOKUS in Berlin

Table 2-4 Network Equipment Provider

	Network Equipment Provider
Norway	Ericsson, Huawei, Palo Alto Networks, Keysight, Cisco
UK	Samsung, Palo Alto Networks, Keysight
Greece	Limemicro, INTRACOM
Spain	Ericsson Spain, Keysight
Portugal	Fraunhofer Fokus, OpenAirInterface Software Alliance, Altice Labs
Germany, Munich	Huawei
Germany Berlin/Luxemburg	External third equipment providers

Table 2-5 Virtualization Infrastructure Service Provider

	Virtualization Infrastructure Service Provider
Norway	Nokia
UK	Samsung
Greece	Openstack
Spain	Openstack
Portugal	Altice Labs internal process

Germany, Munich	Open source based on Docker
Germany Berlin/Luxemburg	Openstack

Table 2-6 NFVI (Network Functions Virtualization Infrastructure) Supplier

	NFVI (Network Functions Virtualization Infrastructure) Supplier
Norway	Nokia
UK	Samsung
Greece	UoP based on Open source solution
Spain	5TONIC members
Portugal	Altice Labs internal process
Germany, Munich	Open source based
Germany Berlin/Luxemburg	Existing standard common off-the-shelf servers

Table 2-7 Data Centre Service Provider

	Data Centre Service Provider
Norway	Telenor
UK	BT
Greece	Network Architectures and Management group, UoP
Spain	5TONIC
Portugal	Altice Labs internal process
Germany, Munich	Huawei in-house small-scale data centre
Germany Berlin/Luxemburg	Own data centers

Table 2-8 Hardware Supplier

	Hardware Supplier
Norway	Nokia
UK	Samsung
Greece	Patras facility carries out HW selection process
Spain	5TONIC members
Portugal	Altice Labs internal process
Germany, Munich	Huawei Technologies
Germany Berlin/Luxemburg	Based on respecting the specific acquisition rule

3 Service catalogues of 5G-VINNI facilities and Entry interfaces for roles

This section describes the 5G-VINNI service catalogues of each facility site as well as the how each facility site serves the supported roles via different service entry interfaces. These different service entry interfaces might have the form of portals for facilitating users accessing the facility and APIs for programmability access. This section outlines the approach for the service catalogue, for both release 0 and release 1.

3.1 Service Catalogue for Norway and UK

Norway and UK facility sites are based on NOKIA's FlowOne solution. Thus, the service catalogue solution is similar. The only difference is on the implemented slice and NSD compositions.

Below is an overview of the overall architecture of a CSP – depicting participation and integration between the Service Orchestrator (FlowOne), NFVO and VNFM/EMS.

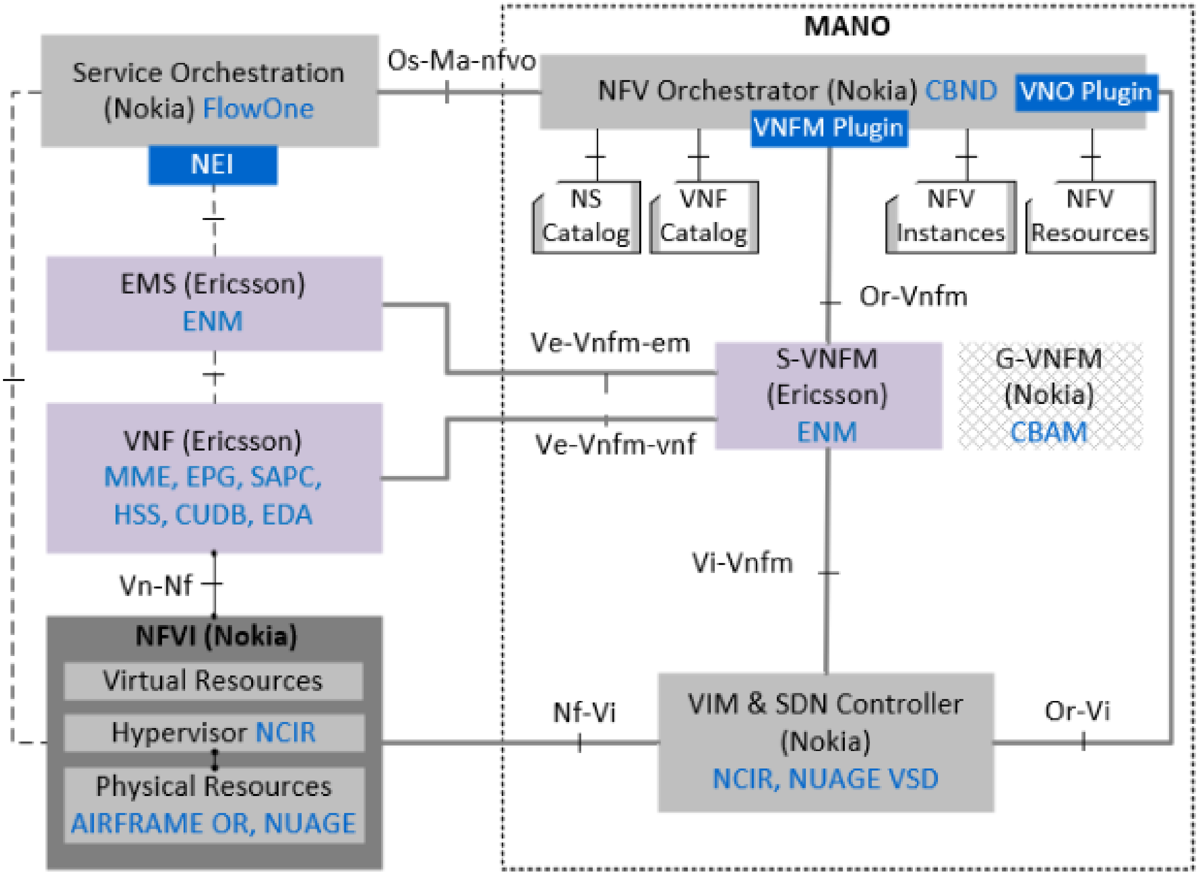


Figure 3-1 Overview of the overall architecture of a CSP in Norway

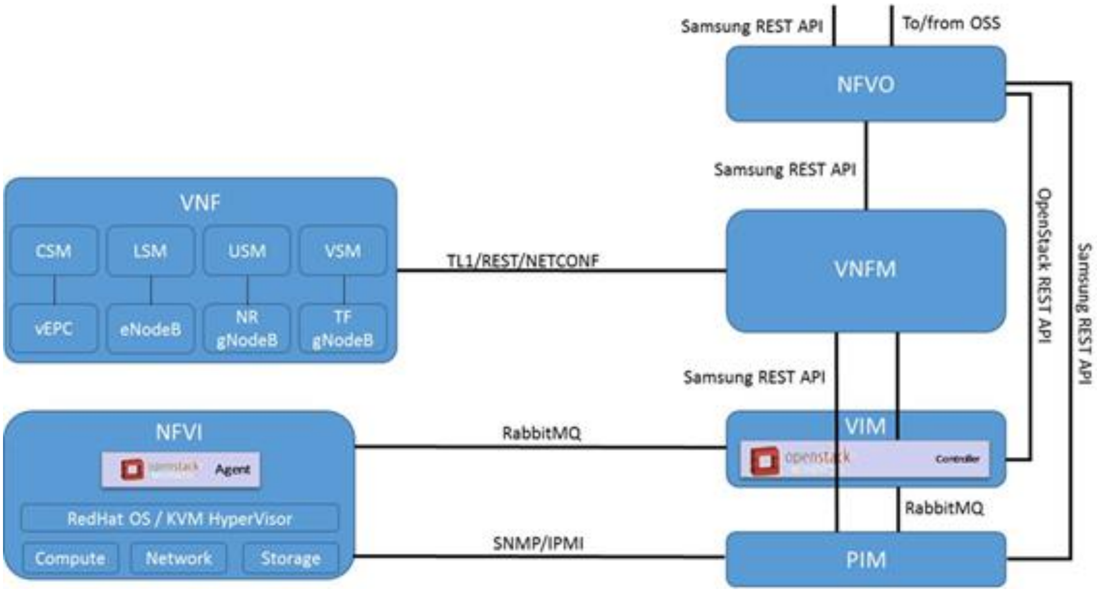


Figure 3-2 Overview of the overall architecture of a CSP in UK

Nokia FlowOne solution is taking care of the E2E service orchestration function. The figure and table below give an overview of the FlowOne components.

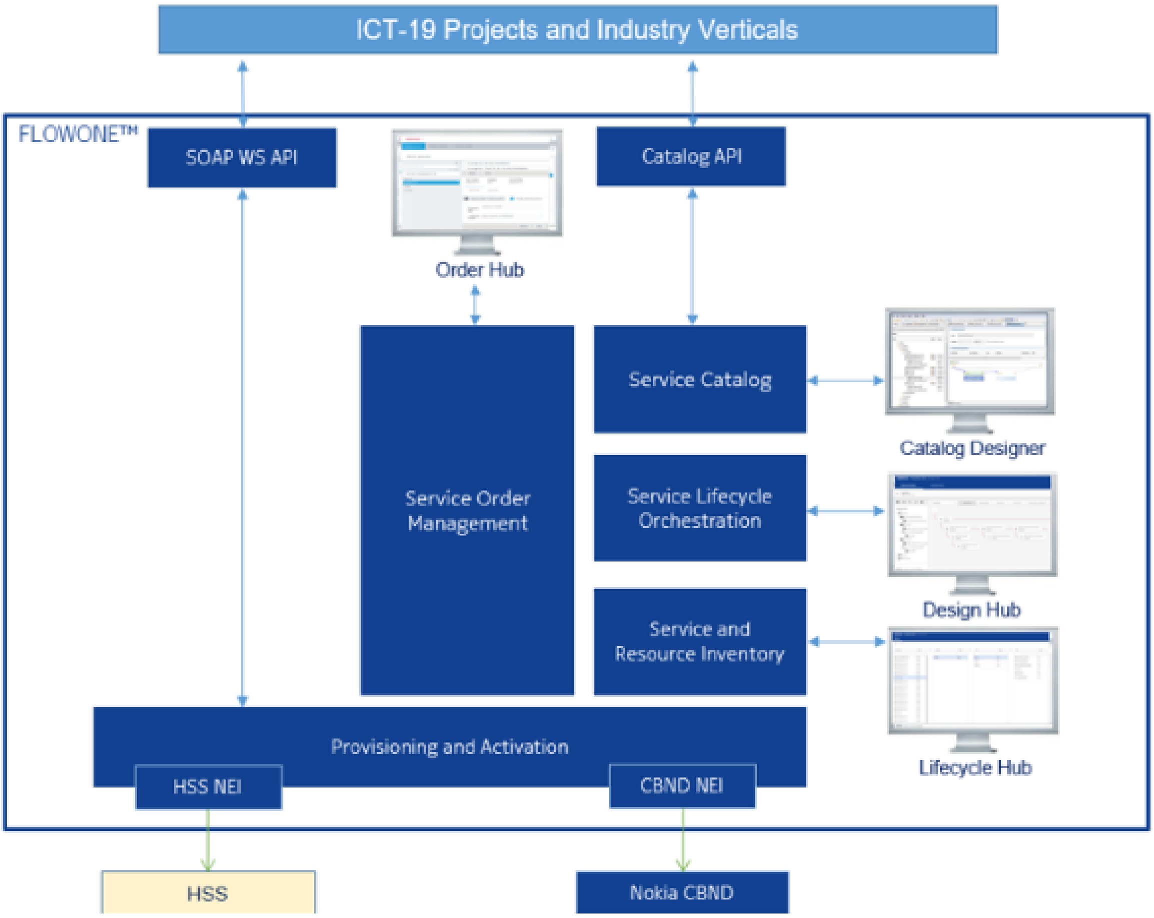


Figure 3-3 Overview of the FlowOne components

Table 3-1 List of FlowOne Components

Component	Description
Design Hub	User interface for designing service specification
Lifecycle Hub	User interface for accessing the service inventory
Order Hub	User interface to handle fallouts/error management
Service Lifecycle Orchestrator	Functionality for managing onboarding of network services and VNF's
Service Catalogue Designer	User interface for designing service specification
Order Management	Nokia Order Management has the capabilities to control and monitor the progress of the service order from receipt to activation, through all the necessary physical an electronic workflow stage.
Service Catalogue	Provides the service definitions and decompositions to more detailed service and resource level specifications
Service and Resource Inventory	In this project, Flowone Service and Resource Inventory (SRI) will be the master for network slice instances and subscriptions
Provisioning and Activation	Provides southbound integration capabilities to CloudBand NFVO and Ericsson HSS.

3.1.1 Service Catalogue

In the Service Catalogue, one can find the Digital Services which are supported by the Service Provider (CSP) and can be purchased by a customer (CSC). The following B2B CFSs[1] will be available for CSCs to order:

- eMBB (enhanced Mobile Broadband)
- mMTC (massive Machine Type Communications)
- URLLC (Ultra Reliable Low Latency Communications)

Once a slice service (of one of the above types), i.e., a B2B CFS[2], is ordered and is live, it becomes available for end users (e.g., mobile phones, IoT devices, etc.) to order and subscribe. The following B2B2C CFSs[3] will also be available for end users (Consumers) to order.

- User Equipment for slice type: eMBB

[1] In the context of Service Catalogue, CFSs are templates of Customer Facing Services — means to order the service. This term is used to be in line with TMF SID. In the context of VINNI, there are two categories of CFSs.

[2] B2B (Business-to-Business) – offered by a CSP to a CSC. E.g., a NSaaS (Network Slice as a Service).

[3] B2B2C (B2B-to-Consumer) – offered by a CSP but, for the Consumer of the CSC. E.g., an User Equipment in a NSaaS.

- User Equipment for slice type: mMTC
- User Equipment for slice type: URLLC

The Service Catalogue (accessible through an API) has the representation of the CFS--RFS decomposition which are linked to a network slice type offering. The following business processes are addressed by the Service Catalogue.

- Create Entity (Service Specification)
- Access to a Catalogue
- Retire a Catalogue entity
- Publish Service/Resource Candidate to Catalogue
- Discover Service/Resource candidate from another catalogue

3.1.2 Definition and on-boarding of VNF, NSD descriptors to the different service catalogues

The E2E orchestrator (Nokia FlowOne) comes with a Service Catalogue. It will on-board NSDs (Network Service Descriptors) from the NFVO. The NSDs shall be TOSCA-based service specifications. Once on-boarded, a solution designer (human) will use the Design Hub to design Slice specifications. The Design Hub provides a 'design canvas' for the purpose. The slice specifications, thus built, are CFSs in the context of this business.

Going forward, just to have a closer look at the term 'on-boarding' and how it refers to different business processes depending on the entities on which it acts upon and the actors. For example, on-boarding of a NSD into the E2E Orchestrator's Catalogue, refers to a design time process. Whereas, on-boarding of VNFs into a NFV Infrastructure (NFVI), refers to the run time process of deployment and/or instantiation of a VNF. And, on-boarding 'User Equipment' would mean activating 5G SIM cards in an up and live Network Slice — that has been already ordered and delivered some time back from now.

The following figure gives an overview of the Norway facility site setup. Notice that how even for instantiating one VNF, such as PCRF, the NFVO shall provide a NSD (i.e., NSD-3, in the below figure) containing just that one VNFD for PCRF. This diagram is depicting the orchestration essentials for the NSA 5G Core, only. For the end-to-end solution, please see the next chapter describing LCM where it is explained along with some elements from the information model (for orchestration).

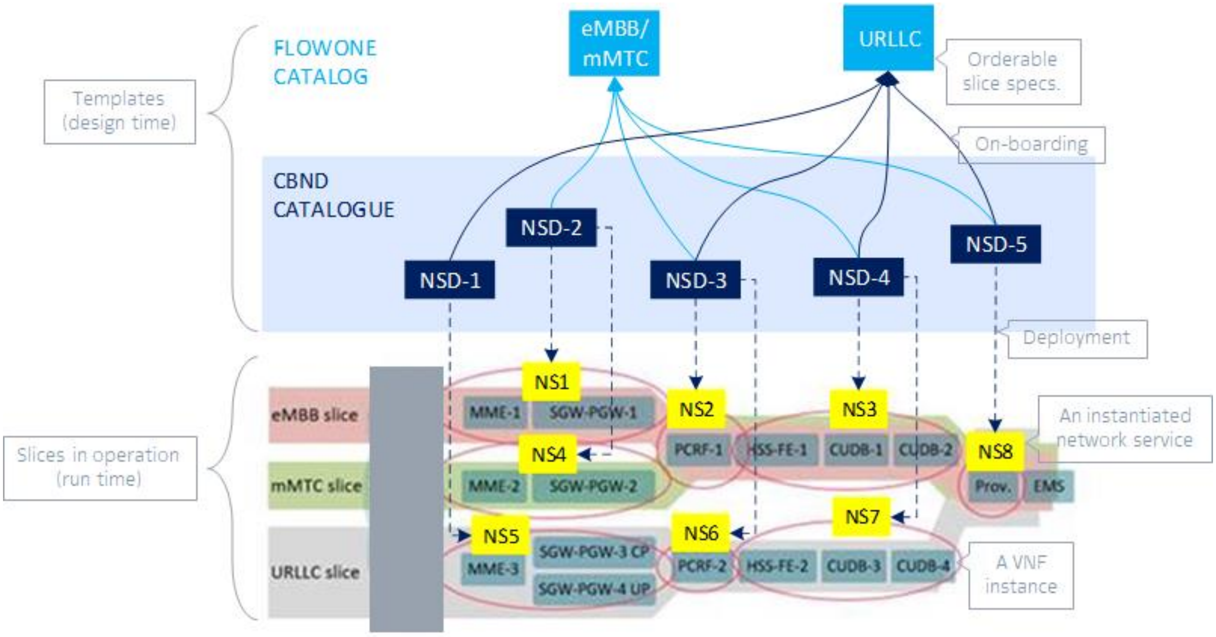


Figure 3-4 Overview of the Norway site setup

For the UK facility site setup, NFVO and core owner is Samsung. The NSD composition, in terms of VNFDs, differs. The following Figure 3-5 gives an overview of the UK facility site setup. Notice how even for instantiating one VNF, such as VNB, the NFVO shall provide a NSD (i.e., NSD-4, in the below figure) containing just that one VNFD for the 5G FWA VNB. This diagram is depicting the orchestration essentials for the NSA 5G Core and 5G RAN and is still subject to confirmation. For the end-to-end solution, please see the next chapter describing LCM where it is explained along with some elements from the information model (for orchestration).

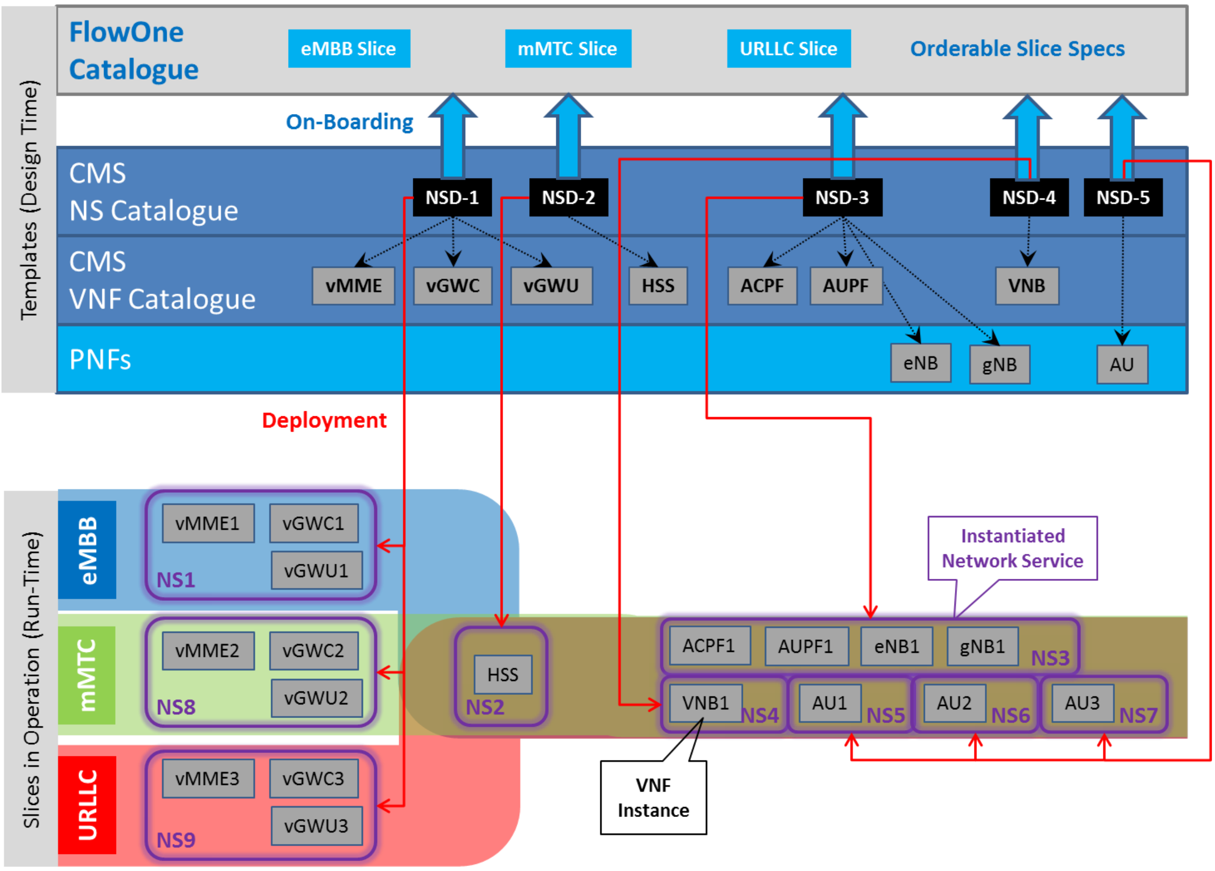


Figure 3-5 Overview of the UK site setup

3.2 Service Catalogue for Greece and Spain

Greece and Spain facilities will use a common solution based on Open Source MANO (OSM) as well as an adaptation of an experimentation portal by the H2020 project 5GinFIRE.

The Service Catalogue is a feature that will be exposed by the facility site portal and will contain various service offerings of the facility site. These service offerings will be delivered under the Network Slice as a Service (NSaaS) model. As stated in D3.1 [D3.1], this model relies on the deployment of NSIs and their delivery towards 5G-VINNI customers, allowing them to build on top their own services and carry out experimentation activities on them.

The Service Catalogue is maintained by the facility site (CSP) via the facility site portal user interface, where the facility site can define its service offerings through 5G-VINNI Service Blueprints (VINNI-SBs).

5G-VINNI Customers can define requirements through service descriptors, which contain specific attributes of a Service Blueprint for the specific customer.

The Service Catalogue will also expose an API based on TM Forums OpenAPIs [OpenAPI]:

– TMF633 Service Catalogue Management API

– TMF641 Service Ordering

Since the facility uses OSM, the Service Catalogue utilizes the SOL005 NBI API of OSM to expose Network Service Descriptors (NDSs) as well as Network Slices Templates (NSTs). The Service Catalogue performs transformations from the ETSI defined models as used by OSM for describing VNF, NS descriptors (SOL001, SOL004, SOL006) and the slicing templates defined by OSM to the TM Forum OpenAPIs models.

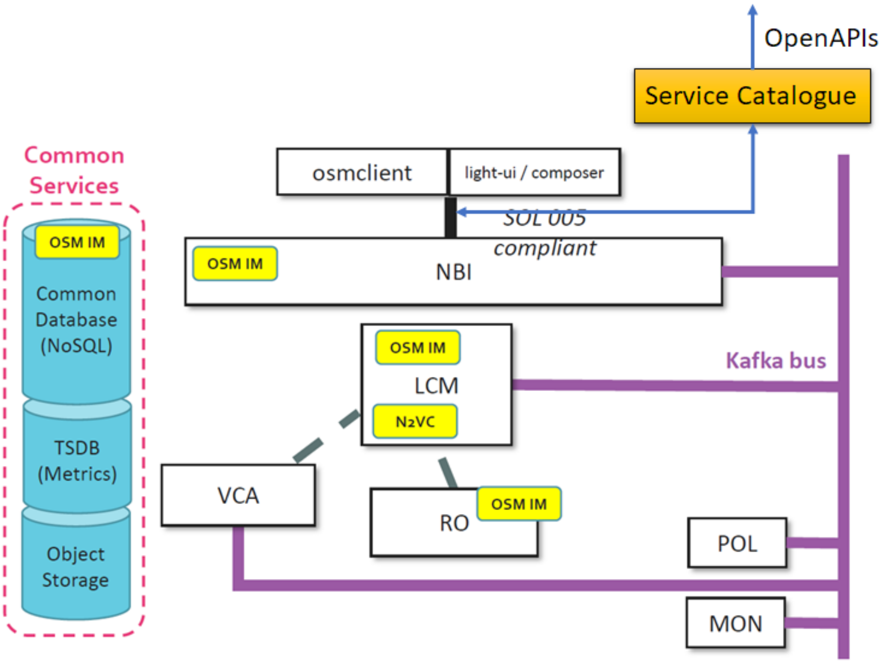


Figure 3-6 Service Catalogue in the OSM architecture

3.2.1 Definition and on-boarding of VNF, NSD descriptors to the different service catalogues

Two roles will be able to upload VFN and NS descriptors and define slices to be onboarded in the facility site, the customer (CSC) and the provider (CSP). CSC as well as the CSP will use the facility portal to maintain and on-board different artifacts. The facility site portal mainly:

- holds information about User data and roles retrieved from other repositories like OSM Keystone
- holds information about VNF, NSD, Slices.
- Utilizes the OSM repository/catalogues via the OSM SOL005 NBI API.
- Categorizes artifacts

The facility site portal implements the RESTful protocols specification for the Os-Ma-nfvo Reference Point. The portal will utilize:

3.2.1.1 NS and VNF Package Management

REST wrapper for the NS and VNF package service via resource Descriptors. Provides methods for onboarding, updating and downloading NS and VNF packages.

3.2.1.2 VNF Descriptor Management

REST wrapper for VNFD descriptor management provides methods for onboarding, updating, querying, and deleting a VNF descriptor through the Individual resource Descriptors and Content.

3.2.1.3 VNF Lifecycle Management

REST wrapper for the VNF lifecycle management service. Provides methods for querying a VNF and retrieving VNF records via resource Descriptors.

3.2.1.4 Network Service Descriptor Management

REST wrapper for the **NSD** service (via resource Descriptors). Provides methods for onboarding, updating, querying, and deleting a network service descriptor.

3.2.1.5 Network Service Lifecycle Management

REST wrapper for network service lifecycle management. Provides methods for instantiating, updating, finding, and terminating a network service (NFV-NS). Also provides methods for creating, updating, listing, and deleting or VNF forwarding graph (VNFFG). This will be implemented via the NS Lifecycle Management interface.

3.2.1.6 Network Slice Template Lifecycle Management

REST wrapper for network slice template lifecycle management. Provides methods for instantiating, updating, finding, and terminating a NST.

3.2.1.7 Hosting of third party VNFs and Images management

3rd party VNFs can by on-boarded through the portal. However, VNF Images are uploaded to the facility site only from the Network Operator that has also access to the VIM. The customer must request the upload of VNF Images and also provide any licensing details.

3.2.1.8 VNF, NSD validation

The facility site portal makes a validation in terms of packaging compliance and modeling. The facility site portal currently supports descriptors following the ETSI proposed YANG models (e.g. SOL001, SOL004, SOL006).

3.2.1.9 VNF upgrade/migration/suspension issues and dependencies with NSDs

Deployed NSDs depend on specific on-boarded VNFs. Currently it is not possible to upgrade these VNFs due to dependencies.

3.3 Service Catalogue for Portugal

SONATA is being developing in the 5GTANGO H2020 project (http://5gtango.eu). All the results of this work are available at GitHub (https://github.com/sonata-nfv) under the Apache2 licence.

In the case of SONATA, the following ways of interacting with the Service Platform (SP) are provided:

- A web-based Graphical User Interface (GUI)[4];
- A command line interface (CLI)[5];
- An API to all the features of the Service Platform[6], which is used by both the GUI and the CLI.

All features available in the GUI and the CLI use the API.

[4] Available at GitHub: https://github.com/sonata-nfv/tng-portal

[5] Available at GitHub: https://github.com/sonata-nfv/tng-cli

[6] Available at GitHub: https://github.com/sonata-nfv/tng-api-gtw

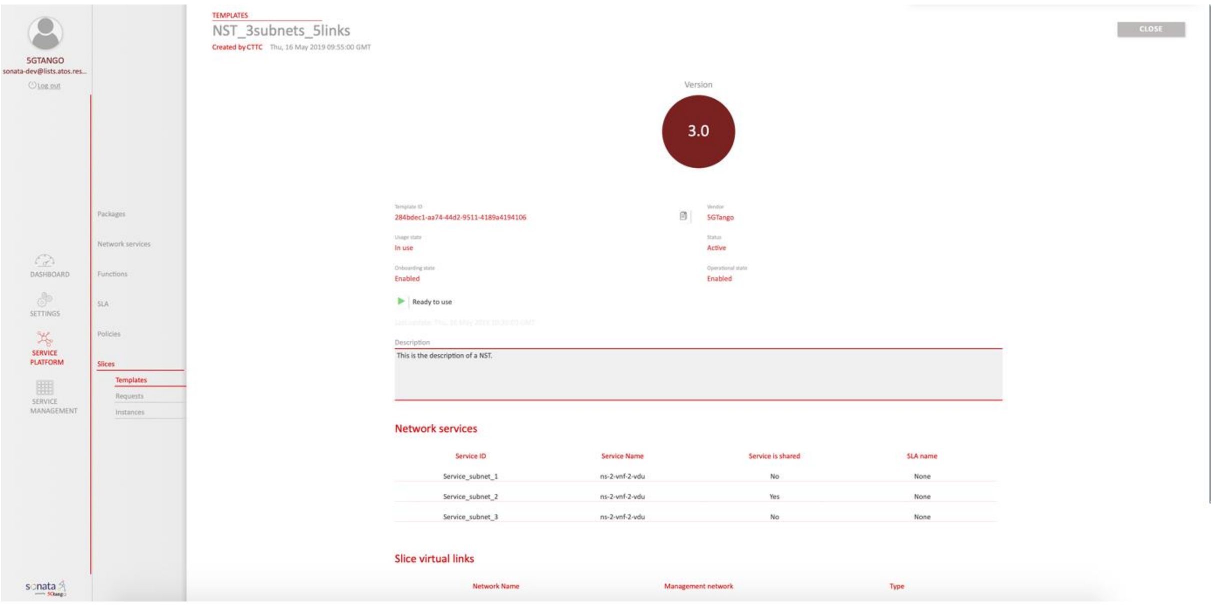
The SONATA Service Platform works in tandem with the Service Development Kit (SDK) and the Validation and Verification platform (V&V, see D2.2 Architecture Design[7]).

3.3.1 Service Catalogue

The Service Catalogue in SONATA is a center piece of the overall solution.

It accepts packages containing NSDs and VNFDs and Slice Templates (STs), among other kinds of descriptors, which are not relevant here.

STs define which NSs are part of the template and define how these services connect to each other (see next figure).



TEMPLATES

NST_3subnets_5links

Created by CTTC Thu, 16 May 2019 09:55:00 GMT

Version

3.0

Template ID
284bdec1-aa74-44d2-9511-4189a4194106

Vendor
5GTango

Usage state
In use

Status
Active

Onboarding state
Enabled

Operational state
Enabled

Ready to use

Description
This is the description of a NST.

Network services

Service ID	Service Name	Service is shared	SLA name
Service_subnet_1	ns-2-vnf-2-vdu	No	None
Service_subnet_2	ns-2-vnf-2-vdu	Yes	None
Service_subnet_3	ns-2-vnf-2-vdu	No	None

Slice virtual links

Network Name	Management network	Type

Figure 3-7 How the SONATA Portal user can create a Slice Template

An example using the CLI is shown here:

```
$ tng-cli -u http://pre-int-sp-ath.5gtango.eu slice --templates
SLICE UUID                                NAME                VERSION             CREATED
AT          284bdec1-aa74-44d2-9511-4189a4194106    NST_3subnets_5link  3.0
2019-05-16 09:55
```

NST instantiation can then be requested through the same interfaces. An ST instantiation instantiates all services that are part of the template and connects them.

Similarly, an existing ST Instance can be terminated.

3.3.2 Definition and on-boarding of VNF, NSD descriptors to the different service catalogues

In SONATA, NSDs and VNFDs enter the Catalogue within a package, which can be signed, verified, etc. A package (actually, all the assets that enter the Catalogue) is universally identifiable by a trio:

- Vendor: the name of the entity responsible for designing the service;
- Name: a descriptive name of the service;
- Version: a version of the asset.

This trio must be unique within a single Catalogue. Further details are available in the GitHub repository[8] and in particular in the available wiki[9].

[7] 5GTANGO D2.2 Architecture Design: https://www.5gtango.eu/project-outcomes/deliverables/2-uncategorised/31-d2-2-architecture-design.html

Whenever a package is submitted to the SONATA SP, it's (un-)packaging process calls a validator, available in the GitHub repository[10]. This validator has proved to be really useful when building the services and functions for the 5GTANGO tree pilots. It can validate the service and function descriptors' syntax, integrity, topology and also execute custom rules.

SONATA supports deploying both VM- and container-based (docker) functions. VMs must be available in a public repository, and containers in a public docker hub.

As stated above, making NSDs and VNFDs available in the SONATA's SP Catalogue implies the definition of those assets, for which the 5GTANGO project provides an SDK. With this SDK we can build, validate and package all the needed NSDs and VNFDs, even those that are located on different facility sites. As stated above, the VM images of the externally located VNFs must be available in a public repository.

One of the main innovations of the SONATA Service Platform is the ability to on-board Service- and Function-Specific Managers (SSMs/FSMs). SSMSs and FSMs are pieces of code (docker containers) that a service or function can bring into the platform (they're described in their own NSD/VNFD) so that its default behavior is adapted to the specific service or function. With this mechanism, we can very easily build features such as upgrading, migrating or suspending VNFs, given that the concrete implementation of each one of these flows are very specific to each service or function. For example, if we are following a KPI that is affected by that upgrade, migration or suspension, we must take additional actions so that the operation does not interfere (or at least interferes in a controlled way) with that KPI. The generic approach that is common in many other Service Platforms cannot guarantee this.

The 5GTANGO/SONATA team took part in the ETSI PlugTest in early 2018, where several third party-provided VNFs were integrated. The implications of this is to use the 5GTANGO SDK tools to build, validate and package the needed VNFDs and NSDs.

3.4 Service Catalogue for Germany – Munich

The virtual machines and Docker containers are based on the usual Linux based distributions and are orchestrated using a MANO like architecture. No specific interfaces are open for interacting with the Munich facility site, however it is possible to pre-configure and setup the entry points as per the requirements from the customer and ESB vertical.

3.4.1 Service Catalogue

All of the services such as slicing and bandwidth on demand will be pre-configured to suite the specific scenario which is being experimented. Therefore, we do not have a specific definition of the procedure for onboarding services.

3.4.2 Definition and on-boarding of VNF, NSD descriptors to the different service catalogues

Most of the VNFs and network services will be preconfigured to suite the specific scenario which is being experimented. Therefore, we do not have a specific definition of the procedure for onboarding VNFs. No specific interfaces are open for uploading VNFDs and NSDs due to the experimental nature of the Munich facility.

[8] GitHub repository for SONATA Package: https://github.com/sonata-nfv/tng-sdk-package

[9] Wiki for SONATA Package: https://github.com/sonata-nfv/tng-sdk-package/wiki

[10] GitHub repository for SONATA Package Validation: https://github.com/sonata-nfv/tng-sdk-validation

3.5 Service Catalogue for Germany – Berlin, Luxembourg

3.5.1 **Service Catalog**

The Berlin and Luxemburg facility sites are using as basis for the Service Catalog the OpenBaton Release 5 Marketplace [OpenBaton-MP] which includes the open source components which are offered by the facility. The Marketplace is extended by an internal service catalogue which includes the specific 5G components, especially different customizations of the Open5GCore for the core network as well as different application enablers.

3.5.2 **Definition and on-boarding of VNF, NSD descriptors to the different service catalogues**

In this section the building and the on-boarding of VNF packages and of Network Service Descriptors is described. The specific functionality is following directly the OpenBaton mechanisms [OpenBaton] as the facilities are based on OpenBaton for orchestration.

3.5.2.1 NS and VNF Package Management

The OpenBaton NFVO supports two different formats for VNF Packages:

- TAR archive following the ETSI NFV specification for VNF Descriptors and Packages
- CSAR archive following the TOSCA simple profile for NFV specification.

For the VNF packages, OpenBaton uses a set of configurations and software images which are required in order to have enough knowledge to run the software. It includes:

- VNF Descriptor: containing all the information required by the NFVO for deploying the VNF (more information available at the VNF Descriptor page).
- Image: passed using a link to an image file (typically QCOW) available for being dowloaded via HTTP. At the moment, passing an image file inside the VNF Package is not supported: there is some work in progress to allow it.
- Metadata file providing additional information to the NFVO for understanding what's the content of the package.
- Scripts: containing all the scripts that can be used for lifecycle management

A comprehensive description of the VNF Package Management is provided at [OpenBaton-VNFPM]

3.5.2.2 VNF Descriptor Management

The VNFD descriptor has to be provided together with the VNF Package Management and has to be included as part of the Network Service Descriptor. A comprehensive description of the VNFD and capabilities is provided at [OpenBaton-VNFD].

3.5.2.3 VNF Lifecycle Management

The VNFD includes a set of events that are supported for the lifecycle: INSTANTIATE, CONFIGURE, START, TERMINATE and SCALE (in, out, up, down). More information is available at [OpenBaton-VNFD].

3.5.2.4 Network Service Descriptor Management

The NSDs are part of the overall slice package and can follow the specific formats of:

- JSON file representation of the information model specified by the ETSI MANO specification
- TOSCA compliant Network Service Template on-boarded via CSAR archive compliant with the TOSCA Simple Profile for NFV.

3.5.2.5 Network Service Lifecycle Management

The network lifecycle is automatic based on the information from the NSD. Additional operations to enable outside control of the network service lifecycle during runtime may be provided through a non-standard interface to the OSS of the service in order to test the opportunity of such operations

3.5.2.6 Network Slice Template Lifecycle Management

The network slice template is directly associated with the network service in case of the OpenBaton based deployments. Above mentioned considerations apply.

3.5.2.7 Hosting of third party VNFs and Images management

3rd party VNFs can be onboarded through the creation of a comprehensive network service package which includes existing VNFs and new VNFs. The onboarding is controlled by the facility management to assure the IPR compliance as well as the proper service integration.

3.5.2.8 VNF, NSD validation

The VNFD and NSD are validated through the testing of the end-to-end slice deployment.

3.5.2.9 VNF upgrade/migration/suspension issues and dependencies with NSDs

VNFs can be upgraded and suspended through the usage of a non-standard management interface which overrides the rules and dependencies of the NSD for an operations and maintenance dedicated time interval. This is possible as the Berlin and Luxemburg facilities are experimental oriented and thus, do not need to assure a reliability of the system beyond the experiments duration.

4 Support and LCM of Network Slice Services

The focus of this section is to describe per facility site their approaches on Network Slice Service management, which models and specifications are supported by their deployed services and tools and how the provisioning of slices will be performed by the 5G-VINNI experimenters. This section outlines the approach for the lifecycle management of network slice services, for both release 0 and release 1.

CSP will expose a number of VINNI-SBs towards CSCs. These VINNI-SBs represents service offerings against which CSCs can issue service orders for the delivery of network slice services. The structure of any VINNI-SB was extensively discussed in D3.1 [D3.1] (Chapter 4). As shown in Figure 4-1, VINNI-SB consists of four main parts:

- *Service type*: provides a high-level description of the slice service to be provided from this VINNI-SB. Any VINNI-SB must specify one of the following options: enhanced Mobile Broadband (eMBB), ultra-Reliable Low Latency Communication (uRLLC), massive IoT (mIoT) and customised.
- *Service topology:* specifies the functional nodes of the slices and their associated topology. Each of this functional node is referred to as a Service Component (SC), each being a unified abstraction of a given network functionality. SCs are technology-agnostic, modular, and can be easily chained to form different topologies, being these topologies flexibly extended (or even modified) with the attachment of 3rd party VNFs. The topology presented in a VINNI-SB and its possibilities of extension (or even modification) are highly dependent on the service type the VINNI-SB refers to.
- *Service requirements*: specifies the requirements of the slice service to be provided from the VINNI-SB. These requirements include *i)* performance requirements, *ii)* functional requirements and *iii)* network optimisation requirements. The requirements presented in a VINNI-SB are highly dependent on the service type the VINNI-SB refers to.
- *Service exposure, monitoring and testing*: specifies the service capability exposure made available to the CSC. This exposure is based on a four-level classification, with each higher level allowing the CSC to gain access to a lower abstraction management entity. Depending on the selected level, the CSC can consume management data (e.g. performance measurements, fault data) and trigger enabled management operations (e.g. LCM) at different abstraction layers, which is relevant for testing and monitoring activities conducted at run-time.

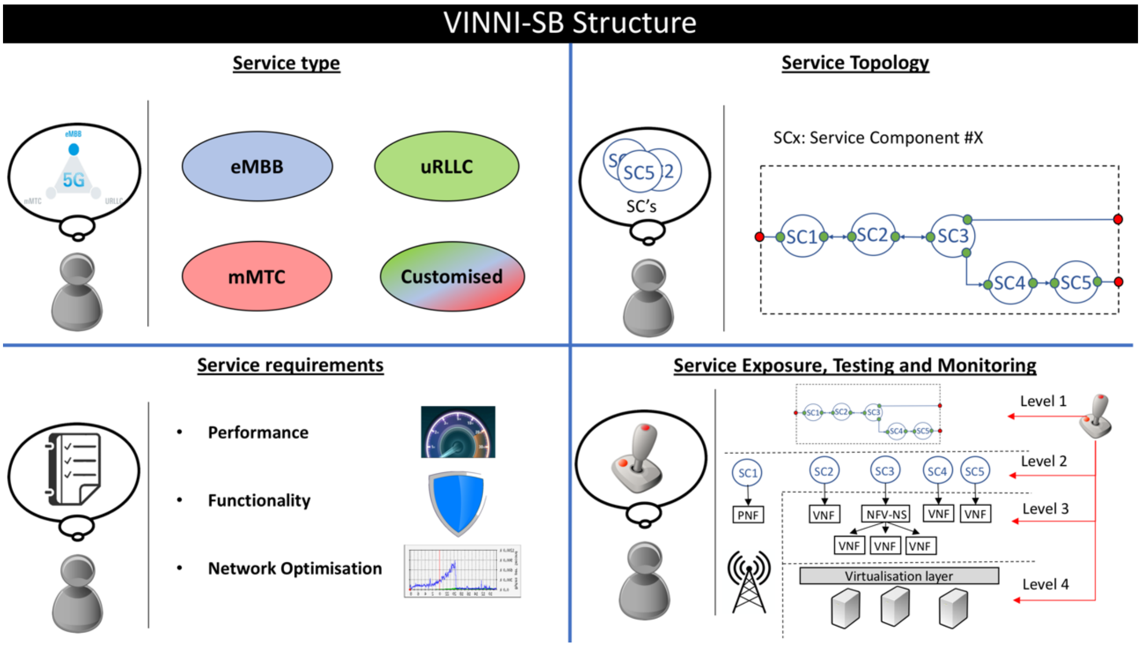


Figure 4-1 VINNI-SB Structure

4.1 Norway and UK

Both facility sites are based on FlowOne, so the approach described in this section is common.

4.1.1 Network Slice LCM

4.1.1.1 Network Slice LCM for Release 0

Release 0 will provide only a basic capability for network slice services, based on eDecor. The NSDs, containing associated VNFs, will be on-boarded into FlowOne. These NSDs will then be used as templates to instantiate each of the 3 network slices. The 3 network slices will share the NS instances for vRAN and HSS, while instantiating 3 separate vCores to support a dedicated core for each network slice. See section 3.3 for more details on network slice composition in terms of NSDs and VNFs.

4.1.1.2 Network Slice LCM for Release 1

Figure 4-2 depicts a high-level view of distribution and mastering of specifications (Catalogue), processes, and instantiated elements (Inventory; that eventually deliver the customer services), across the layers of the orchestration stack.

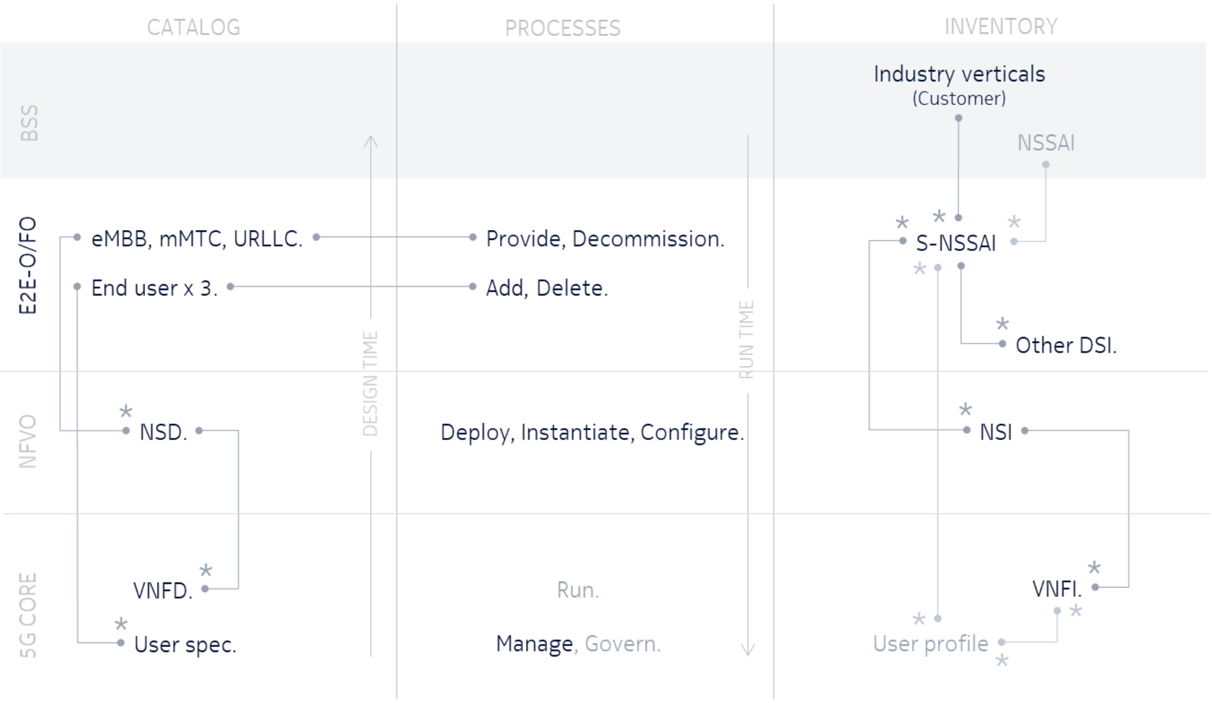


Figure 4-2 A high-level view of distribution and mastering of specifications (Catalogue), processes

In the above figure, all entities in dark blue text depicts the scope of FlowOne, the E2E orchestrator. The horizontal layers in which each of them appears, depict which layer masters or owns or provides them. For the entities in grey text, FlowOne is not an actor (consumer or provider or manager), but, are indirectly related to E2E service fulfilment and hence, are depicted here to set the End-to-End context, right.

Domain Service Instances (DSIs) refers to identifiers of services running in domains other than the 5G Core. Such services, although instantiated by respective domain controllers or via manual work order, must generate service identifiers and this information will be documented in FlowOne's inventory. S-NSSAI is the end-to-end slice identifier within a PLMN. NSSAI is a collection of S-NSSAIs. NSSAIs are UE domain entities, hence are positioned in the BSS domain.

Nokia FlowOne will document *contracts* between Industry verticals (CSC) and the CSP (Telenor/BT). Meaning, industry verticals will be 'Customers' and products sold will be 'Network slices', i.e., S-NSSAIs, under 'Subscriptions'. Services and Resource Inventory (SRI) will not document 'User's or 'End user's or, 'User Equipment' information. It is assumed that End Users and their services will be documented by inventory of the CSC. Nevertheless, within 5G-VINNI ecosystem, this information can be fetched from slice specific HSS/PCRF VNFs. In other words, a query by UE is not. The query should at least have a NSSAI to locate and fetch information of a UE. User (or, UE) profiles in UDMs of each slice (S-NSSAI) will also list all the S-NSSAIs that the UE is subscribed to. To FlowOne, all UE management orders will come in with exactly one S-NSSAI..

Either Serving PLMN or UE can be the master of a NSSAI. Deliverable D1.2 [D1.2] describes the idea behind NSSAI such that, if a UE configures a NSSAI, it will be able to get connected to several slices (S-NSSAIs) that it has subscribed to. As such, a NSSAI can be part of commercial contracts in several ways, e.g., a UE can be a direct customer of it, a CSC owning several slices can bundle for its UEs, etc. Service Orchestration offers and manages S-NSSAIs.

In addition to this, there is gap for the BSS layer requirements, e.g., data model and user experience. This topic is currently a 5G-VINNI research item.

4.1.2 Supported Network Slice Service Modeling approaches

4.1.2.1 Supported Network Slice Service Modeling approaches (Rel 0)

The proposed model is based on TMF SID (ref. GB922 Information Framework R18.0, *Service Domain::Service ABE::Customer Facing Service ABE*; *Diagram Figure SO.06 - Basic Service Model – A Starting Point*). In the B2B context, where a CSP sells a NSaaS to a CSC, S-NSSAI and NSSAIs are CFSs. NSIs are Resource Facing Services (RFSs). VNF instances are resources. Similarly, going bottom-up, VNFDs are resource specs, NSDs are RFS templates and Slice specs are CFS templates.

It is worth noting that in a B2B2C context, wherein, User Equipment CFSs are sold by the CSP to a CSC, network slices are RFSs (Resource Facing Services). But, since here we are interested in documenting only the B2B contracts, in the inventory, network slices (instances) will not be visible as RFSs, in practice. And, since a slice will always be sold and delivered sequentially first, before on-boarding an User Equipment, there is no point in modeling the network slice types as RFS specs in the Service Catalogue.

4.1.2.2 Supported Network Slice Service Specification approaches (Rel 1)

At the moment, a responsibility split is being drafted. Here, responsibility refers to managing service decomposition and applying intelligence for assigning values to relevant attributes, at run time, during a Network Slice order delivery.

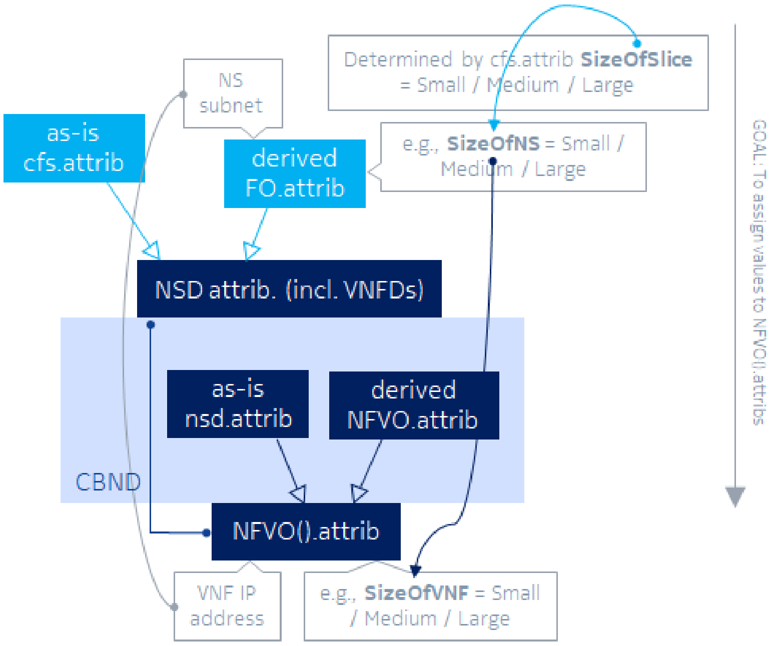

Determined by cfs.attrib **SizeOfSlice** = Small / Medium / Large
NS subnet
as-is cfs.attrib
derived FO.attrib
e.g., **SizeOfNS** = Small / Medium / Large
NSD attrib. (incl. VNFDs)
as-is nsd.attrib
derived NFVO.attrib
CBND
NFVO().attrib
VNF IP address
e.g., **SizeOfVNF** = Small / Medium / Large
GOAL: To assign values to NFVO().attribs

Figure 4-3 A typical run time scenario is depicted

In the above Figure 4-3, the arrows among the blue boxes are referring to categorization using UML notation. For example, 'as-is CFS.attrib' is a category of 'NSD.attrib'.

In the above Figure 4-3, a typical run time scenario is depicted. The end goal is to determine and supply values to attributes of the function calls that NFVO triggers towards the 5G Core in order to deploy network services. FlowOne would receive a set of attributes and values within the CFS containers that it exposes to Order Entry. Next, FlowOne would fill up the NSD templates. To do so, some values can be carried forward as is from the order, whereas some needs to be derived. The NFVO would do similar tasks but at the next level: from NSDs to deploying Network services by instantiating the required VNFs and peripherals, one-by-one. Again, for some attributes, it can just carry forward values from what it got from FlowOne. And, for some attributes it will derive the values.

A typical example could be IP assignment for each VNF. FlowOne would supply an IP subnet for a NS. NFVO would assign IP addresses to each element of that NS. Another, rather indicative (just to hold the general idea; may or may not turn out to be practical, going into the implementation) example is the SizeOfX attribute. The thinking at this point is the Slice order would have an attribute such as 'Size of Slice' with possible values such as Small, Medium, Large, Extra Large, etc. FlowOne would derive the Size of NS for each NS that needs to be instantiated. And, NFVO would, in turn, derive the Size of VNF attribute for each VNF.

4.1.3 Provisioning of network slice instances (Rel 1)

For Network slice CFSs, the following request types will be implemented:

- Provide.

  |> get (also book) IP subnet(s) for NSI(s) from NM.

  |> get NSI reusability info (also book) from SRI. CFS specs in Catalogue to specify reusability.

  |> instantiate NS(s) --- to instantiate the 5G core slice on CBND.

  |> manual Work Order --- setup other networks e.g., transport, etc. & connect the 5G slice.

  |> record in SRI

  |> respond with S-NSSAI.
- Decommission.

  |> get NS ID(s) (5G core) & DSI from SRI --- query by S-NSSAI.

  Also, mark the S-NSSAIs as 'decommissioning-in-progress', in SRI.

  |> decommission NS(s) --- decommission the 5G EPC slice; call CBND with the NS IDs.

  |> manual Work Order --- to cease connectivity services on another network.

  |> mirror SRI --- reflect the changes in SRI.

For User Equipment CFSs, the following request types will be implemented. Post go-live, only new subscriber creation and then deletion of only these newly created ones, will be supported.

- Add --- Profile based subscriber creation on HSS.

  |> get HSS-FE ID, IP from SRI --- query by S-NSSAI.

  |> Execute 'create subscriber'
- Delete --- Delete subscriber from HSS.

  |> get HSS-FE ID, IP from SRI --- query by S-NSSAI.

  |> Execute 'delete subscriber'

4.2 Greece and Spain

Both facilities are based on OSM, so the approach described in this section is common. For 5G-VINNI Rel-0 what Greece and Spain will offer is based on OSM Release FIVE, while for 5G-VINNI Rel-1 a migration to OSM Release SIX is planned. which brings advancements in monitoring. The VINNI-SB structure shown in Figure 4.1 is expected for next releases of 5G-VINNI project, after Rel-1. So, for Rel-1 only a subset of the parameters will be available for vertical specification.

OSM software stack allows the deployment and operation of a wide variety of NFV-NSs, and their offering towards OSM clients following the Network as a Service (NaaS) delivery model. This model allows an OSM clients to request the instantiation of one or more NFV-NSs, by selecting the appropriate NSD(s) - elaborated at design time - and providing (missing) instance-specific parameters. The result is one or more NFV-NS instances that the OSM client can consume for its own purposes. This is the way OSM has traditionally worked up to Release FOUR, e.g. in 5GinFIRE project.

However, 5G-VINNI project is committed to go beyond NFV approach, introducing the network slicing concept and make it available to CSC, allowing them to take the role of OSM clients. For this end, the OSM release FIVE has extended the information model from previous releases. This new model introduces the network slice information element following the conceptual outline suggested in ETSI NFV-EVE 012 (see Figure 4-4) and extensively discussed in D1.3 [D1.3].

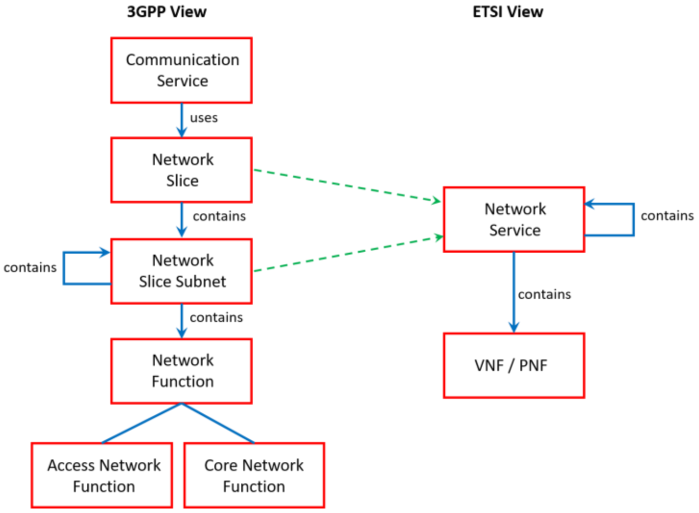


Figure 4-4 Network slicing and network service relationships (adapted from ETSI NFV-EVE 012)

This figure shows that an NFV-NSs can be viewed as the resource-centric view of a network slice subnet, and these network slice subnets can be flexibly combined to build out different (E2E) network slices. According to this approach, in OSM a Network Slice Instance (NSI) will be composed of one or more Network Slice Subnet Instances (NSSIs), each deployed as an instance from a given NFV-NS. For the management of these NSIs, the OSM has extended the NBI component (see Figure 4.1) with the incorporation of a module Slice Manager. The resulting functionality of OSM and corresponding mapping to 3GPP management system and NFVO is shown in Figure 4-5.

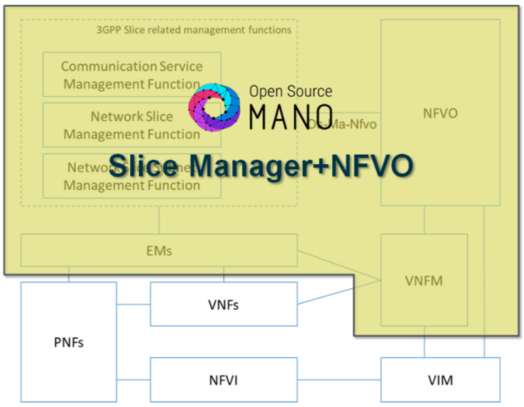

3GPP Slice related management functions
Communication Service Management Function
Network Slice Management Function
Network Slice Subnet Management Function
Open Source MANO
Os-Ma-Nfvo
NFVO
Slice Manager+NFVO
EMs
VNFM
VNFs
PNFs
NFVI
VIM

Figure 4-5 OSM extended to support E2E management of NSIs.

Once deployed and operative, NSIs can be delivered to corresponding OSM clients following the NSaaS model, allowing customers to build on top their own services and carry out experimentation activities on the infrastructures. This delivery implies the definition of an appropriate exposure level, according to the requirements set by each OSM client and its networking expertise. Different exposure levels mean that the OSM client is allowed to consume management data at different abstraction levels, and to reach management blocks placed at different abstraction layers to trigger LCM operations.

In this section the deployment, operation and delivery of NSIs with OSM will be discussed. Section 4.1 presents the information model on which NOP will rely on for defining and building up NSIs on top of an NFV-ready infrastructure. Then, Section 4.2 focuses on how a CSP makes its service offerings to CSCs following NSaaS model, allowing the CSCs to make service orders towards the CSP. The CSP translates these service orders into concrete network slice requirements. Finally, Section 4.3 provides a description of how NOP will handle NSI provisioning, according to the network slice requirements sent from CSP.

4.2.1 **Supported Network Slice Service Modelling approaches**

OSM release FIVE will allow NOP to deploy and operate NSIs and make them available to the CSP. The CSP will be in charge of delivering these NSIs to the CSCs following the NSaaS (see Deliverable D3.1) model. This service delivery model can be described as a particularization of the NSaaS model, but more focused on the enablement of 5G use cases. The evolution from NSaaS (up to OSM release FOUR) to NSaaS (from OSM release FIVE onwards) requires the definition of a single unified framework where the views on network slicing from both 3GPP and ETSI NFV can coexist.

- On one hand, 3GPP deals with the network slicing concept from a functional viewpoint, focusing on *NSI application layer management*. Examples of issues addressed under this vision are the study of what specific RAN/CN functions make up an NSI at both UP and CP, how these functions communicate with each other via normative (service-based) interfaces, and the implications their sharing brings in terms of isolation and customization.
- On the other hand, ETSI NFV studies network slicing from a deployment viewpoint, focusing on *NSI virtualized resource management*. This vision study how the flexible allocation of VNFs and their dynamic composition into NFV-NSs can support the resource requirements of an NSI.

As seen from the above description, these views are complementary and need to interact with each other. Indeed, slicing without 3GPP vision is simply an application-agnostic NFV deployment, without any notions on the semantics that allow NFV-NSs to behave as desired. Conversely, slicing without NFV vision does not enable flexibility and agility in network slicing run-time operation, or dynamic provision of resources where and when required. To align these views, OSM release FIVE has proposed the unified conceptual framework shown in Figure 6-13, highly inspired by the ETSI NFV-EVE 012 [EVE012] proposal (Figure 4-4). The resulting framework allows the NOP to cope with the network slice concept and define a consistent information model that can be used for the design and instantiation of network slices.

Please refer to Annex H.1 on how the OSM framework allows the definition of an information model for network slicing in 5G

4.2.2 **Supported Network Slice Service Specification approaches**

This section will focus on CSP and CSC roles. The CSP role will be taken by both Spain and Patras facility sites. The VINNI-SBs offered by Spanish and Greek CSPs in their service catalogues will be based on the structure shown in Figure 4.1. For network slice service specification, the CSC selects the corresponding VINNI-SB (e.g. eMBB/uRLLC/mIoT/customised VINNI-SB) and fills it according to its particular service order requirements, resulting in a Service Order. Stored by the CSP, this Service Order provides a self-contained specification of the network slice service instance for that CSC.

Once a CSC specifies the service as desired and clicks the “Submit service order”, a new VINNI-Service Order Request is created out of the VINNI-SB. Stored by the CSP, this request provides a self-contained specification of the network slice service ordered by that CSC.

4.2.3 Provisioning of network slice instances

The provisioning of NSIs can be described throughout their lifecycle. The lifecycle management process network slicing is originally presented in 3GPP TS 28.530 [3GPP28.530], and extensively discussed in D1.3 [D1.3].

The lifecycle management of NSIs carried out by Spanish and Greeek NOPs is compliant with this 3GPP view, thanks to the conceptual outline the OSM information model is based on (see Figure 6-13). In the next subsections, each phase will be discussed individually.

4.2.3.1 Preparation phase

This phase begins when CSP receives a service order from the CSC. At that moment, a VINNI-SD is created out of VINNI-SB and stored in OSM repository. Then, CSP checks if that VINNI-SD is feasible (e.g. check if all the mandatory parameters have been specified, if there is enough capacity, etc.), and if so, sends corresponding network slice requirements to NOP. With these requirements, the NOP carries out a set of operations arranged into sub-phases: NST design and NST on-boarding.

For the first sub-phase, the NOP takes the received requirements and translates them into the following tuple: {required NST, instantiation parameters}, so NOP can deploy the NSI when and where required. For the preparation phase, only the first field of this tuple is relevant. The required NST shall follow the information model introduced at the beginning of Section 4 and shown in Annex A. This NST can already exist (pre-defined NST, designed beforehand) or not (NST created on demand) in the service catalogue. To handle these two scenarios, two operations are made available to NOP through NST management interface (Section 3):

- *nst-create*: allows designing an NST from scratch. This operation is triggered by the NOP when there is no pre-defined NST that can be reused for the definition of the required NST.
- *nst-update*: allows modifying some fields from a given NST. This operation is triggered when the required NST can be obtained by updating some of the content of an already existing NST.

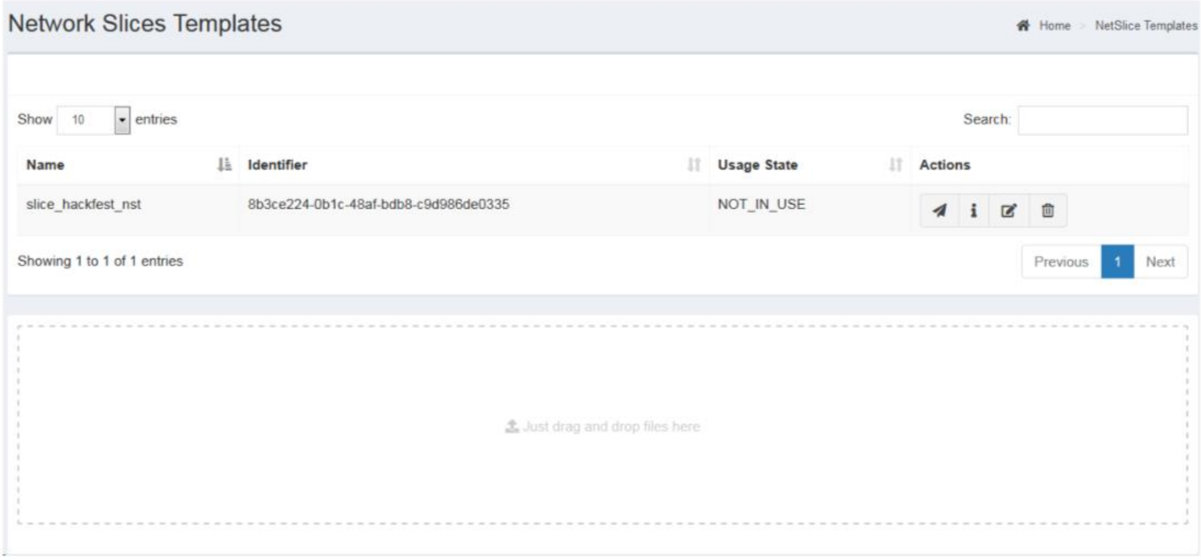
Either of the abovementioned options allows NOP to have the required NST designed, and thus the first sub-phase completed. From that moment onwards, the NST is on-boarding can get started.

In this second sub-phase, the NOP aims at injecting the required NST into the service catalogue. This process not only consists in on-boarding the NST itself, but also the NFV-NS and VNF Packages referred by the NST (see information model in Annex A). OSM’s NBI offers APIs that support CRUD (Create, Read, Updated, Delete) operations to handle these NFV-NS and VNF packages (and their contained NSDs and VNFDs). In these operations, the necessity checks to validate in-model and cross-model consistency are performed. These API calls are implemented over the NFV-NS/VNF Package management interfaces and the NSD/VNFD management interfaces described in Section 3.

Once the on-boarding process finishes, the NST is stored in the service catalogue, so it can be used for NSI creation later on. At this moment, the NOP can invoke two operations from NST management interface (section 3):

- *nst-list*: allows listing NSTs available in the service catalogue.
- *nst-show*: allows showing the content of a given NST.

An example of the invocation of the *nst-list* operation is shown in Figure 4-6. For sake of clarity, the operation has been invoked using the UI.



Network Slices Templates

Name	Identifier	Usage State	Actions
slice_hackfest_nst	8b3ce224-0b1c-48af-bdb8-c9d986de0335	NOT_IN_USE	

Showing 1 to 1 of 1 entries

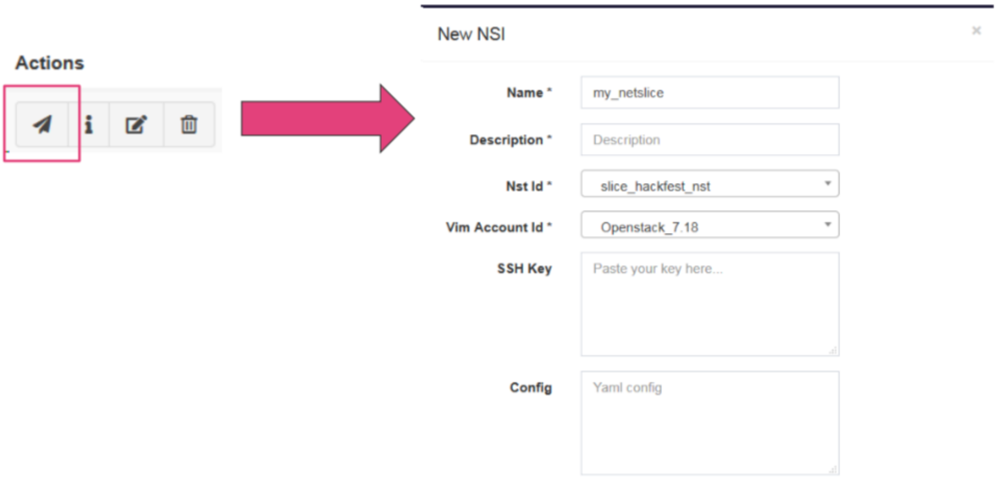
Figure 4-6 List of NSTs on-boarded to the OSM catalogue

When the NOP finishes invoke either of the two operations, the preparation phase gets finished.

4.2.3.2 Commissioning phase

Once the NST and corresponding NS/VNF Packages have successfully been on-boarded in the service catalogue, they can be used as deployment templates for the actual NSI deployment. Alike the preparation phase, the commissioning of the NSI from the NST is a phase that can be split into two sub-phases: network slice instantiation and NSI configuration.

Network slice instantiation consists of the day-0 operations that allows creating an NSI where all the components are instantiated. These operations are defined at all the abstraction levels, ranging from VDU level (e.g. VNF component) to network slice level, and are invoked using the VNF, NFV-NS and network slice lifecycle management interfaces described in Section 3. To trigger the instantiation of a network slice, the NOP will consume the *nsi-create* operation from the network slice lifecycle management interface. An example of how this operation looks like is shown in Figure 4-7.



Name * my_netslice

Description * Description

Nst Id * slice_hackfest_nst

Vim Account Id * Openstack_7.18

SSH Key Paste your key here...

Config Yaml config

Figure 4-7 Instantiation of a network slice

For *nsi-create* operation, the two fields of the tuple {required NST, instantiation parameters} that was derived in the preparation phase is now needed. A brief overview of the steps that OSM takes when this operation is invoked is shown below:

1. First, the NOP analyses the content of the NST and decompose it into its constituents: network slice subnets and virtual links connecting them.
2. Secondly, for each network slice subnet, the corresponding NSSI is created. This NSSI is an NFV-NS instance. For the deployment of this NFV-NS instance, the following information is considered:
    a) *Instantiation parameters*: from the tuple {NST, instantiation parameters} derived in the preparation phase. The NOP extracts from these parameters the requirements that are relevant for the network slice subnet to be instantiated.
    b) *NSD information element*: from the IM of the required NST (see Annex A). This information element points to the NSD that will be used for deploying the NFV-NS instance. This instance can be deployed in various forms (e.g. with different topologies, with different capacity) and with different deployment constraints considering the above instantiation parameters, by using the mechanisms that NSD has for that end (e.g. flavoring, affinity/anti-affinity rules, etc).
    c) *Iss-shared-nss information element*: from the IM of the required NST (see Annex A). This information element specifies if the above NSD can be shared or not. If this information element set to YES, and if an already NFV-NS instance provides similar capabilities to those needed by the NSSI, then that NFV-NS instance can be associated with the NSSI. Otherwise, a new NFV-NS instance needs to be deployed.
3. Thirdly, for each virtual link providing inter-NSSI connectivity, the corresponding virtual link instance is created. For the deployment of this virtual link, the following information is considered:
    a) Instantiation parameters: from the tuple {NST, instantiation parameters} derived in the preparation phase. The NOP extracts from these parameters the requirements that are relevant for the virtual link to be instantiated.
    b) VLD information element: from the IM of the required NST (see Annex A). This information element points to the VLD that will be used for deploying the NFV-NS instance. This instance can be deployed in various forms (e.g. with different QoS parameters) considering the above instantiation parameters, by using the mechanisms that VLD has for that end (e.g. flavoring).

With the abovementioned steps, the NSI is already created, although not configured.

To start the configuration process, day-1 operations are required at the NSI components. Typical day-1 operations include model-driven interaction with (Virtual/Physical/Hybrid)NFs through the use of Juju charms, which allow NEPs to encapsulate their configuration mechanisms (e.g. YANG/NETCONF, Ansible, SSH+scripts). For (V/P/H)NF application layer configuration, two different kinds of Juju charms can be used: native charms and proxy charms. A brief comparison of these two Juju charms is shown in next table.

Table 4-1 Native charm vs proxy charm

Type of Juju charm	Native Charm	Proxy charm
Scope	Juju charm used for those NFs that are able to run charms inside, e.g. cloud-like VNFs.	Juju charm used for those NFs that do not support running charms inside, e.g. PNFs
Interaction	NF interaction happens directly from the orchestrator	Proxy charm uses the appropriate configuration protocol to interact with the NF and run the desired actions from the primitive

4.2.3.3 Operation phase

Once the NSI has been successfully commissioned, the NSI becomes a relevant object for further operation actions. Unlike 3GPP view, operation phase in OSM includes deactivation/activation (e.g. pausing/resuming) as part of the set of modification operations that can be triggered at run-time, depending on the outcomes resulting from the supervision and reporting activities carried out over the NSI throughout its lifetime.

Examples of basic supervising and reporting activities that the NOP can conduct over the NSI are allowed through the following operations, all exposed in the network slice lifecycle management interface:

- *nsi-list*: list of all operative NSIs
- *nsi-op-list*: show the history of operations that has been triggered over the NSI since its creation.
- *nsi-op-list*: shows the information of the operation over a NSI
- *nsi-show:* shows the record of the NSI

Other more sophisticated activities can be performed, assisted by the MON module, or by the Bugzilla(see 6.2).

Depending on the information received from these supervision and reporting activities, the running NSI might need to be modified, in order to keep it in the desired state. For this end, the NOP can trigger a wide variety of day-2 operations over that NSI. These API-driven operations are quite aligned with the specificities indicated by the CSC in the VINNI-SD, and can fall into one of these categories:

- *Operations at the virtualized resource level*: includes operations that has a direct impact on the virtualized resources supporting the NSI. Examples of these operations include scaling operations (e.g. subjected to upper and lower thresholds), creation and deletion of performance measurement jobs, subscribe and notify operations for performance metrics and fault alarms, instantiation and termination of testing components to complete test campaigns, etc. Depending on the service exposure level selected by CSC (see Figure 4-1), these operations can be performed at different abstraction levels, e.g. ranging from NSI level to VNF instance levels. For more details on these operations, see Section 6.2.
- *Operations at the application level:* includes operations that are relevant only for the specific functionality that the NSI offers. Examples of these operations include the addition, modification and deletion of new subscribers, changes in security parameters, changes in routing across the service chain, etc. Those actions need to be enumerated and codified in the constituent NFV-NS Packages the NST refers to and are exposed by the API as primary actions available in that given NSI.

4.2.3.4 Decommissioning phase

As any other (on-demand) instance of a manageable entity, it is possible to decommission an NSI. This decommissioning means removing all the dedicated components and releasing their underlying resources, but not the shared and dependent components. For those components, they need to be re-configured and their resources should be adjusted accordingly. To illustrate this scenario, consider the case where a NSSI from the NSI that needs to be decommissioned was deployed using a running NFV-NS instance (e.g. a NFV-NS instance also serving other NSSI from another NSI). In such a case, where the NSSI from the NSI to be decommissioned is removed, the NFV-NS instance cannot be removed. Otherwise, the other running NSSI will be affected. To avoid this, when removing the required NSSI, the NFV-NS instance will be re-configured (e.g. disassociated from the removed NSSI) and scale-in accordingly.

As seen again, the lifecycle management of NSSI and NFV-instances, although dependent, are separate.

For the decommissioning phase, Spanish and Greek NOPs may invoke the *nsi-delete* operation through the network slice LCM interface.

4.3 Portugal

4.3.1 **Supported Network Slice Service Modeling approaches**

NOPs create Network Slices instances and manage its lifecycle, providing to the CSPs. This section describes how this process can be done

4.3.1.1 SONATA Release 5.0 Overview

The SONATA open source software is the result of the developments performed by the H2020 SONATA project and the further extensions of the 5GTANGO H2020. SONATA implements the ETSI NFV MANO functions, namely VNFM and NFVO. In addition to MANO capabilities, the recent SONATA Release 5.0 supports Network Slicing management capabilities

SONATA Release 5.0 is aligned with different standards. It is aligned with NGMN/3GPP view on the concept that enables the composition of Network Slices by using smaller pieces: i.e. the Network Slice Subnets. Network Slice Subnets can in turn be mapped into ETSI NFV artifacts, namely to Network Services, as suggested in ETSI NFV EVE012 [ETSI-NFV-EVE012]. As a result of that, Network Slices can be created using ETSI NFV artefacts as building blocks, namely Network Services (NSs). Thus, a Network Slice can be defined as a list of NSs being interconnected. This view is also aligned with 5G America architecture.

SONATA Release 5.0 has an internal component, named Slice Manager, which follows a similar approach as the Network Slice Life-Cycle Management component defined in 5G Americas. Similarly, it is responsible to manage the lifecycle of Network Slices by managing the lifecycle of the underlying NSs, as well as the combination of them through network interconnections. Interconnections can be either local to a certain Point of Presence (PoP) or remote (inter PoP), in the latter case requiring the setup of a WAN link.

The NGMN model considers the possibility of sharing Network Slice Subnets among different Network Slices. In fact, the 3GPP architecture suggests that some components of the 5G Core could be shared among different Slices, making this capability important for 5G. SONATA Release 5.0 supports the creation of Network Slices which share a certain Network Service; i.e. a single instance of an NS is used by two or more instances of Network Slices. When a Network Slices is removed, shared NSs are not removed until no Network Slice instances are using this NS.

SONATA Release 5.0 supports the on-boarding of Network Slice Templates (NSTs), which designs the Network Slice, indicating the list of comprising NSDs, the particular characteristics of each NS, as well as the set of networks interconnecting them. Using NSTs multiple Network Slice Instances (NSIs) can be created with the same characteristics. The NSI lifecycle can be managed, creating and removing instances. This model is aligned with 5G Americas parlance, and similar to the NGMN concept of blueprints (templates).

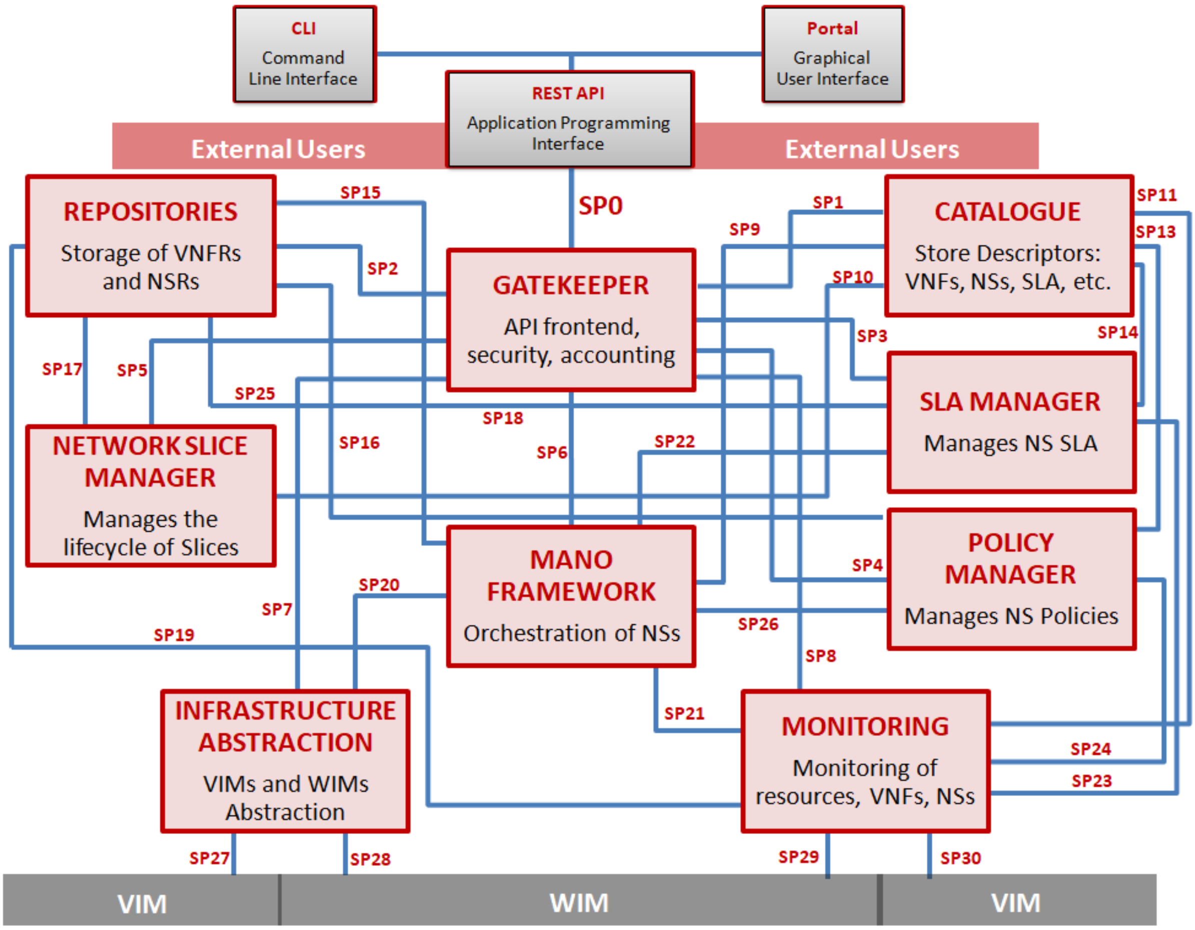


Figure 4-8 SONATA Release 5.0 Architecture.

4.3.1.2 SONATA Release 5.0 Modeling

The automated lifecycle management (LCM) provides the Network Slice with the ability to be agile, enabling the on-boarding of NSTs, as well as the creation, scaling, healing, modification or termination of NSIs. In this SONATA Release 5.0 the only LCM operations available are on-boarding of NSTs, creation of NSIs and termination of NSIs.

The NST design is the first step that a Network Operator (NOP) needs to perform, describing the Network Service in a JSON file. Once the NST is designed, the lifecycle can be managed using this template. See Annex I.1 for an NST description in SONATA

4.3.2 Supported Network Slice Service Specification approaches

The SONATA Release 5.0 makes available a Service Management portal (GUI) for CSCs to self-manage NS and Slice-based Services (deployment and other LCM operations). This Portal displays the Services Catalogue, showing the available Services templates (VINNI-SB). For Service instantiation, the CSC selects the Service Template to be used, associated to the desired Service type. Once selected, the Service can be tailored by tackling on some input parameters related to topology, requirements, and service exposure, testing and monitoring (as mentioned above), enabling a further customization of the Service. When the Service is properly configured, the final order can be submitted. A similar approach is followed for other Service LCM operations for existing Service instances.

4.3.3 Provisioning of network slice instances

The lifecycle management (LCM) of Network Slice Instances (NSIs) is described in 3GPP TS 28.530 [3GPP28.530] and extensively described in D1.3 [D1.3]

This model is support by the SONATA Release 5.0, although some LCM operations are not available at the moment.

4.3.3.1 Preparation

The preparation phase includes the creation and on-boarding of NSTs, which are templates that describe how Network Slice instances will be built. It is the responsibility of the Network Operator (NOP) to design NSTs based on ETSI NFV artifacts and on-board them into the SONATA Release 5.0 for later use.

The on-boarding of NSTs using the Service Platform is performed via APIs (Portal is not available). The NOP has to edit the NST in any editor of his wish and design it according to the modeling schema described in section 4.3.1. An OpenAPI/Swagger specification of this endpoint is available[11]. An example of this data is:

```
{
  "name": "Example_NST",
  "version": 1.1,
  "author": "5gTango",
  "vendor": "5gTango",
  "description": "This is a description of a NS",
  "sliceServices": [
    {
      "servname": "service_name",
      "nsdID": "66a1c857-76d7-48db-98a1-6674b531b010",
      "slaID": "a365dd7e-42c7-4db1-947f-1e2de4e432cc"
    },
    {
      "servname": "service_name",
      "nsdID": "66a1c857-76d7-48db-98a1-6674b531b010",
      "slaID": "a365dd7e-42c7-4db1-947f-1e2de4e432cc"
    }
  ]
}
```

In case the on-boarding works properly, the NOP should be able to see the new template in the Service Platform. For that, it can use the following REST API:

To get all NSTs

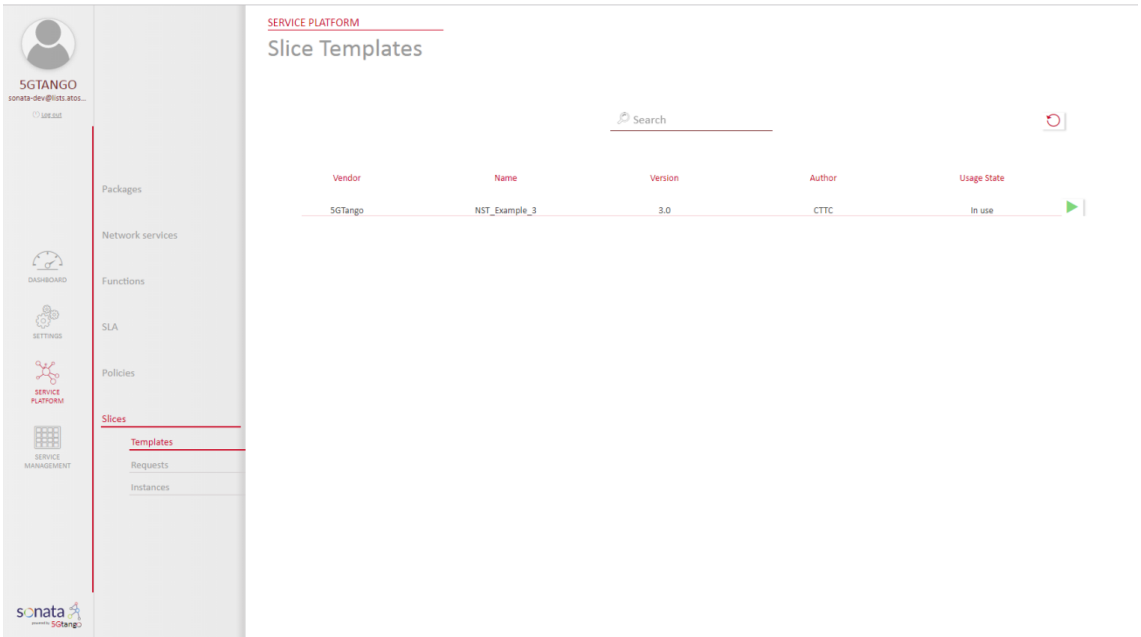
```
curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET
http://{base_url}/api/nst/v1/descriptors
```

To get a specific NST

```
curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET
http://{base_url}/api/nst/v1/descriptors/{nstId}    ; to get one by Id
```

The NOP can also use the Portal to see the list of available NSTs. For that, go to the *Service Platform* menu, and select the menu *Slices* and then *Templates* (see an example in Figure 4-9). Clicking over the template it can be seen the details of the NST, namely the NSs and Virtual Links (VLs) involved.

[11] OpenAPI/swagger specification: https://sonata-nfv.github.io/tng-doc/?urls.primaryName=5GTANGO%20SLICE%20MANAGER%20NetSlice%20Template%20API%20v1.2#/default/post_descriptors



Slice Templates

Vendor	Name	Version	Author	Usage State
5GTango	NST_Example_3	3.0	CTTC	In use

Figure 4-9 SONATA Release 5.0 Portal: List NSTs.

The NST can be removed from the Service Platform using the following REST API (not via Portal).

```
curl -X DELETE http://{base_url}/api/nst/v1/descriptors/{nstId}
```

4.3.3.2 Commissioning

The commissioning phase includes the instantiation of an NSI based on a particular NST. Before the instantiation to be possible, the NST has to be previously on-boarded (Preparation phase). It is the responsibility of the CSC to instantiate a Service which will trigger the instantiation of an NSI.

The commissioning of a Service (NSI) can be performed by the CSC using the following REST API:

```
http {base_url}/api/v3/requests nst_id=<uuid_of_an_existing_NST> name="<nsi-name>"
request_type=CREATE_SLICE description="<nsi-description>"
```

The CSC can also use the Portal to instantiate a Service. Clicking on the highlighted green arrow of the desired NST line, the Service will be commissioned, by immediately instantiating a NSI.

After some time, the Service will be instantiated. To check that instantiation, the following REST API needs to be invoked.

```
curl -i -H "Content-Type: application/json" -X GET
http://{base_url}/api/nsilcm/v1/nsi/{nsiId}
```

On the other hand, to see all Service instances running, the following REST API needs to be invoked.

```
curl -i -H "Content-Type: application/json" -X GET http://{base_url}/api/nsilcm/v1/nsi
```

The CSC can also use the Portal to check if The Service (NSI) was instantiated. Clicking over the instance line it can be seen the details of the Service (NSI), namely the NSs instances involved

4.3.3.3 Operation

The operation phase includes assurance action such as monitor performance, alarms, supervision, etc. It also includes a potentially large list of LCM operations to be performed during runtime operation, such as, scaling, updating, reconfiguration, migration, etc. However, those features are not supported by now in SONATA Release 5.0.

4.3.3.4 De-commissioning

The de-commissioning phase includes the release of all the resources associated to an NSI. By that time, the NSI should be already deactivated, not providing service in a production environment. It is the responsibility of the CSC to trigger de-commissioning of NSIs when Services are not needed anymore.

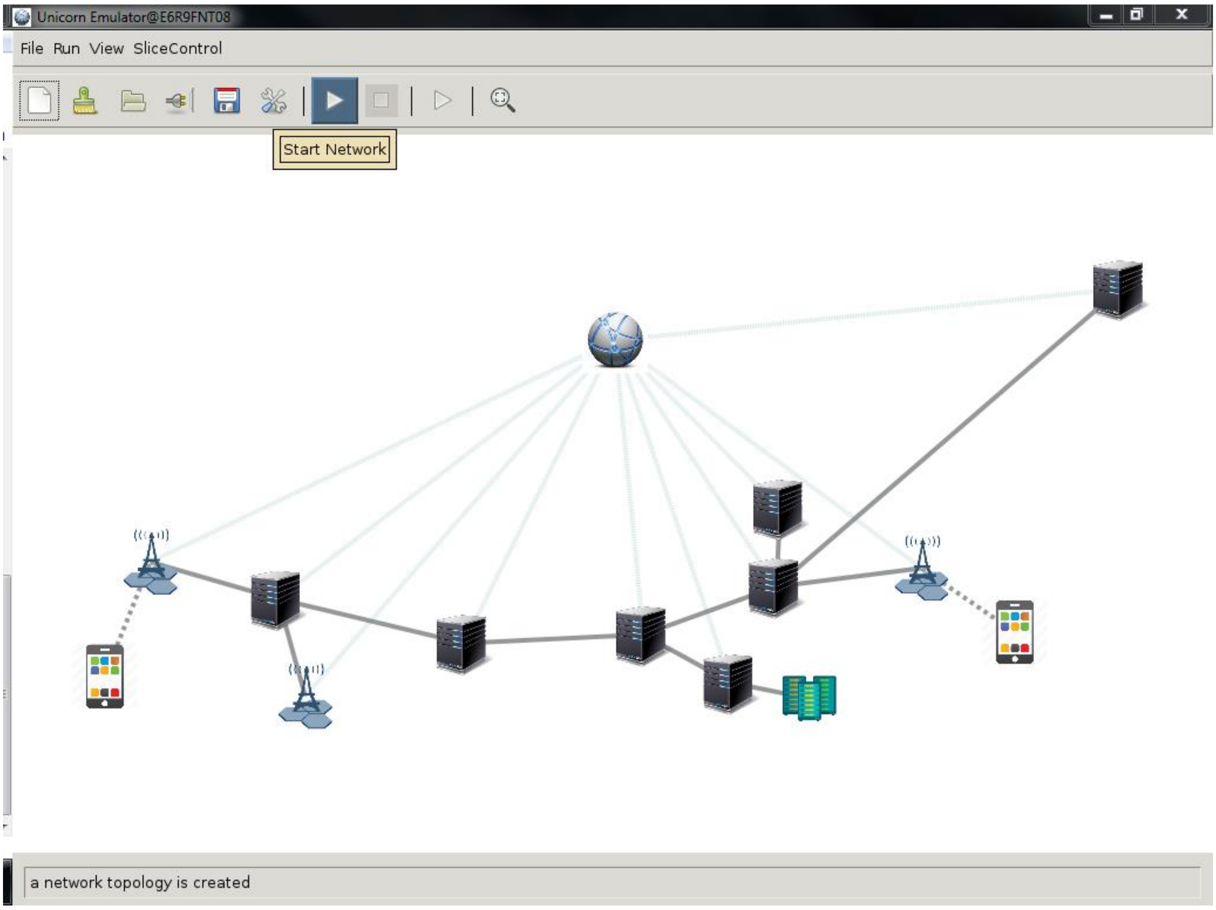
The de-commissioning of an NSI can be performed by the NOP using the following REST API, and indicating the unique NSI id:

```
http {base_url}/api/v3/requests instance_uuid=<nsi_uuid> request_type=TERMINATE_SLICE
```

The NOP can also use the Portal to destroy an NSI. Clicking on the highlighted red square of the desired instance line, the NSI will immediately de-commissioned.

4.4 Germany-Munich

HWDU experimental facility site, does not have a formal MANO system in the sense of ETSI NFV definition. However, we have full control over both the physical platform and the deployed virtual network, including runtime control. A special Network Control Graphical User Interface (GUI) developed in house is used for these purposes



a network topology is created

Figure 4-10 A GUI developed by HWDU Munich to operate NFVI

This GUI allows:

- Virtual network design (topology, number and types of nodes, addressing, assignment to physical hosts)
- Virtual network deployment (start and stop)

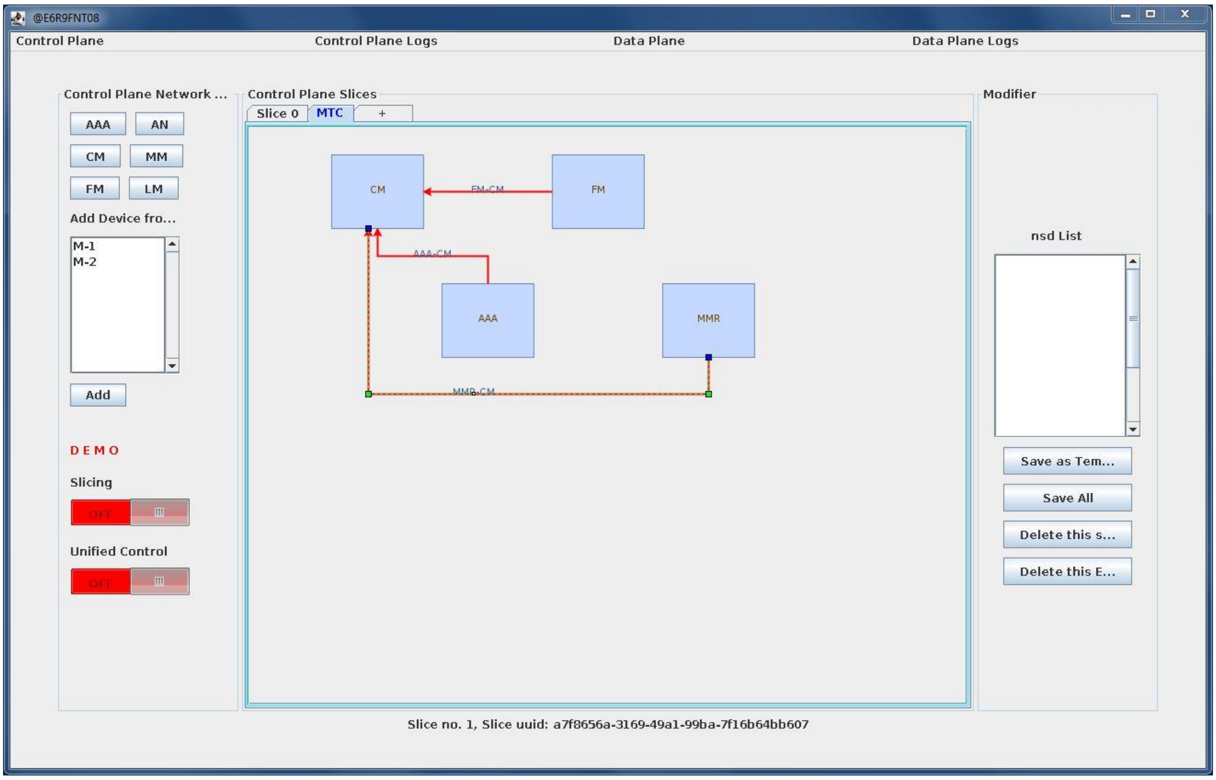
- Virtual network runtime control, including topology control (link up/down, node reattachment, node up/down) and running service control (like remote shell capability into the virtual nodes and the changes to the virtual link).

4.4.1 Supported Network Slice Service Modelling approaches

This is not applicable for the Munich facility site. Network slices are pre-defined for the vertical customers. The focus is on low latency slices for V2X applications.

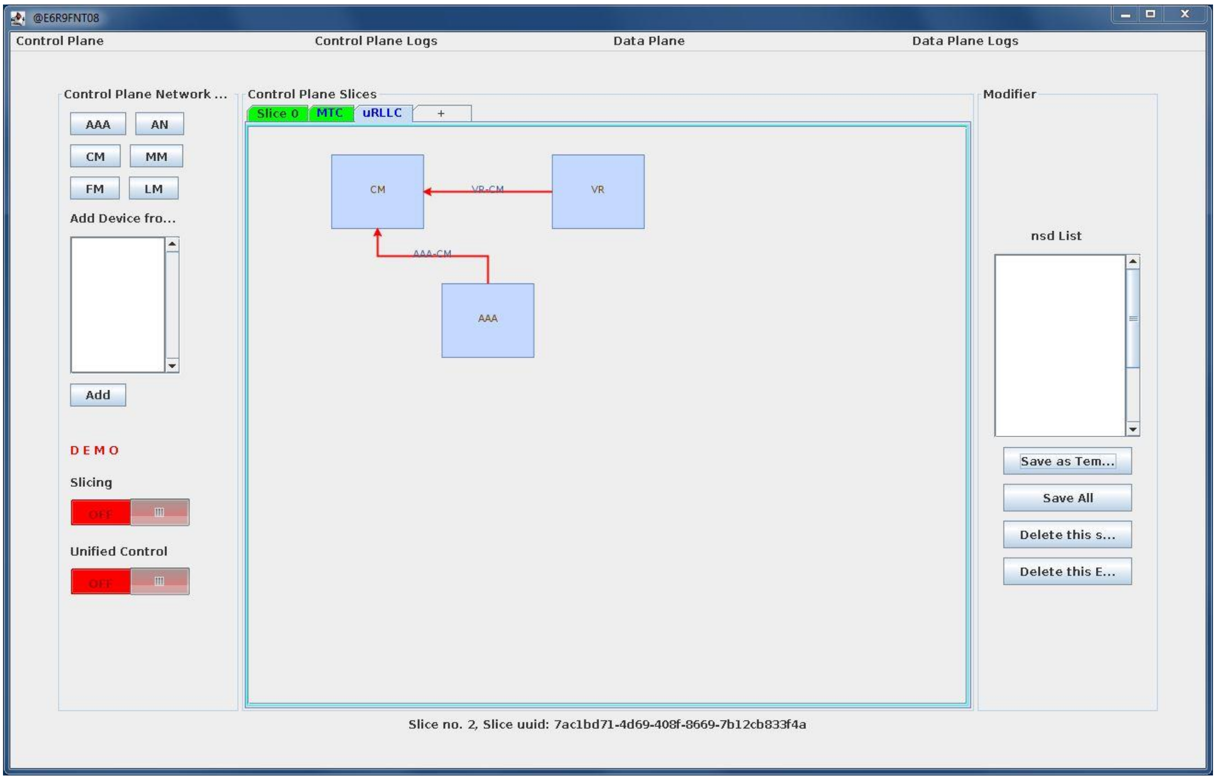
In the Figure 4-11 the GUI that is used to build the model of the slice service, it is possible to specify the different network functions required in the slice such as AAA, MMR, etc.. and the interconnection that is required. It is then possible to deploy this control plan slice to the system.

The acronyms used below are as follows; CM (Connection Manager), FM (Flow Manager), AAA (Authentication Authorization and Accounting), MMR (Mobility Management Reactive).



Slice no. 1, Slice uuid: a7f8656a-3169-49a1-99ba-7f16b64bb607

Figure 4-11 Slice creation functionality, example of an MTC slice.



Slice no. 2, Slice uuid: 7ac1bd71-4d69-408f-8669-7b12cb833f4a

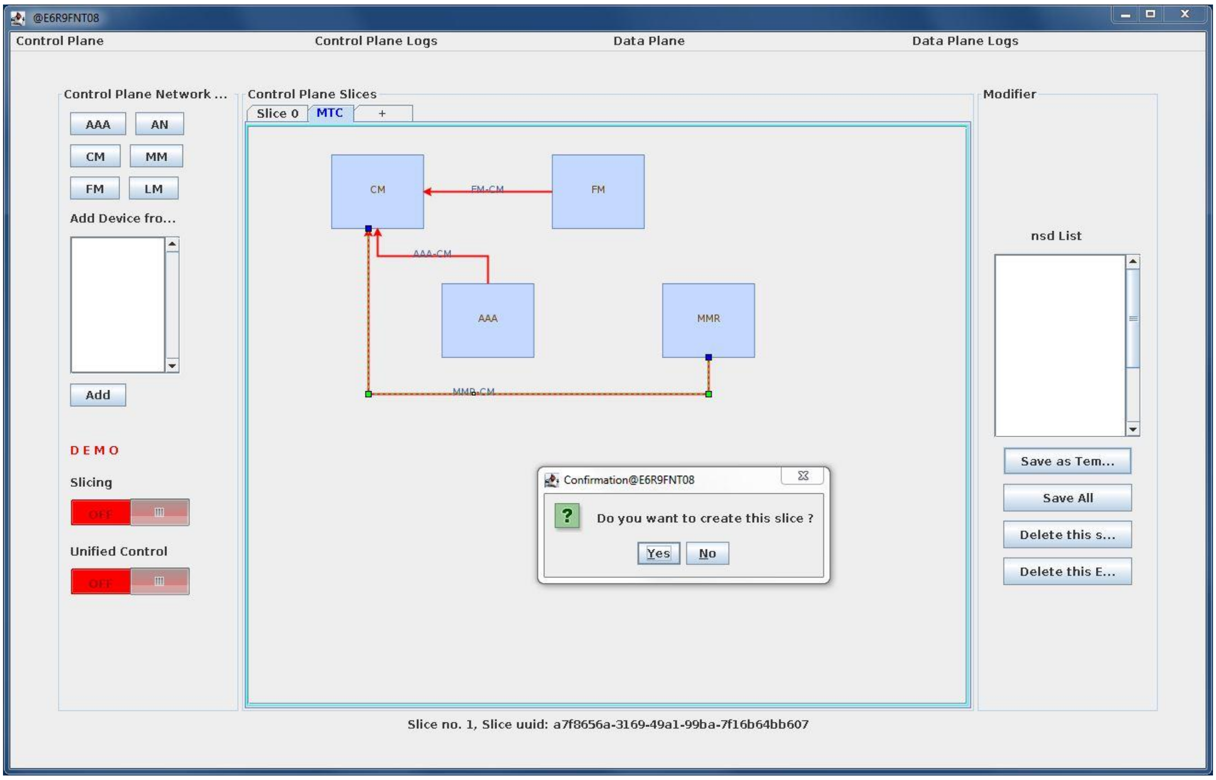
Figure 4-12 Slice creation functionality, example of an URLLC slice.

4.4.2 Supported Network Slice Service Specification approaches

This is not applicable for the Munich facility site. The focus is on slices as a general concept to realize a certain functionality such as low latency communication.

4.4.3 Provisioning of network slice instances

The provisioning of network slices is done using on a GUI that handles the topology build, which then is sent to the specific domain slicing manager. A slice can then be managed following the typical slice life cycles as show in the next figures.



Do you want to create this slice ?

Slice no. 1, Slice uuid: a7f8656a-3169-49a1-99ba-7f16b64bb607

Figure 4-13 Slice provisioning after creating the template

4.5 Germany-Berlin, Luxemburg

OpenBaton software represents an agile ETSI NFV orchestrator which enables standard and innovative support and life-cycle management of network slice services.

At the current moment, OpenBaton does not distinguish between Network Slice Services and Network Services. The main reason for this, is that such a distinction was considered redundant as multiple Network Services may be composed within an end-to-end service under the same tenant.

Within 5G-VINNI project, the orchestration framework proposed will be extended with an end-to-end orchestrator which will act as a TMF Service Orchestrator concentrating only on the split deployment across two domains represented by two NFVOs. This feature is especially interesting for the distributed deployments with edge nodes as proposed for the Luxemburg and Berlin experimental facility sites.

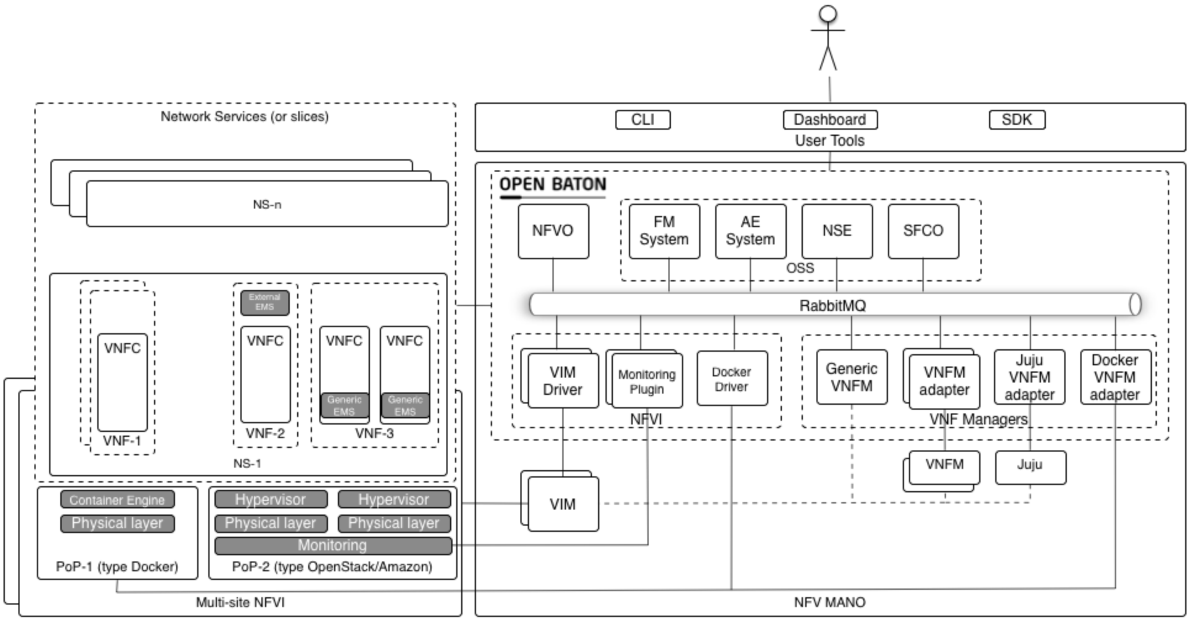


Figure 4-14 OpenBaton as NFVO in the end-to-end architecture

OpenBaton consists mainly of an NFVO, a generic VNFM and VIM drivers, which add the support for different NFVI solutions (e.g. OpenStack and Docker). In this section, the possibilities of deploying and managing Network Slice Services with OpenBaton are described. Section 4.5.1 provides the details of the information model which can be used by NOP for defining and deploying Network Slice Services. Section 4.5.2 focuses on the CSP and the specific slice models offered. Section 4.5.3 describes how NSIs are provisioned.

4.5.1 Supported Network Slice Service Modelling approaches

OpenBaton allows NOP to deploy Network Services in the context of network slicing, supporting the requirements of ETSI NFV for deploying NSI. CSPs can deliver NSIs in the form of deployed Network Service Slices to the CSCs.

A Network Service has to be modeled as an NSD and is deployed as an NSR.

The NSD is a template file, whose parameters are following the ETSI MANO specification, used by the NFVO for deploying network services (as combination of multiple VNFs). There are two different formats supported by the NFVO for NSDs:

- JSON file representation of the information model specified by the ETSI MANO specification
- TOSCA compliant Network Service Template on boarded via CSAR archive compliant with the TOSCA Simple Profile for NFV.

See Annex G.3 for a detailed description.

4.5.2 Supported Network Slice Service Specification approaches

The VINNI-SBs offered by Berlin and Luxemburg experimental facility sites are based on the VINNI-SB defined in D3.1 [D3.1] (Chapter 4). However, being experimental facility sites, the service types will be mainly customized ones according to the needs of the use cases beyond the eMBB, URLLC and mMTC as offered by the main facilities of 5G-VINNI.

The design of these VINNI-SBs will be based on the principle of *use case customization*. The Berlin and Luxemburg facility sites are based on a highly reconfigurable core network, the Fraunhofer FOKUS Open5GCore, which gives the opportunity to develop fast new features and to integrate them into new slice models. With this, the experimental goal of the facility sites is reached, providing the practical path finding for new features beyond the initial 5G standardization.

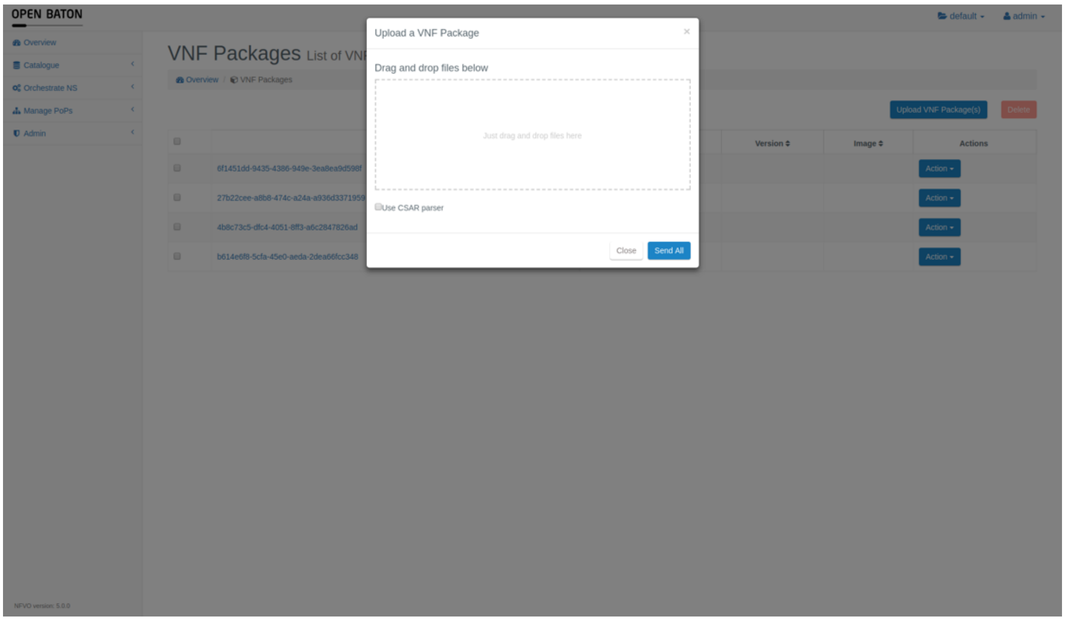
From a service topology, the VINNI-SB is especially concentrating on the split between remote edges, located “on premise” of the use case and the central location of the testbed. The functionality split is considered essential for such deployments as well as the functionality needed to make two slice pieces to work together across a distributed service topology. Albeit the service components will be mostly the same, different edge-central split models will be tested addressing especially the need to reach the service requirements.

Furthermore, it is considered that the service requirements can be achieved not only through the further adaptation of the service and service types towards innovative ones but also by the service topology as presented in the previous paragraph. A set of experiments will be executed across the experimental slice types in order to determine if the requirements will be easier met by service topology change, easier to implement as it relates only to deployment, or through the development of new core network features, harder to make accepted by the large community.

The service topology and the monitoring of the different system metrics will be exposed through a special GUI developed for the core network specifically. Through these means, the tenant can check the quality of the slice proposed and give advice on how it could be further optimzied. This creates a cycle process of optimziation resulting in better slice templates which can be directly used by the use cases. The GUI will be part of the deployed slice and will be exposed to the tenant through an open API, which allows the tenant to access the different network functions deployed in the slice.

4.5.3 Provisioning of network slice instances

A network slice is created when the Network Service is deployed. This process can be started by the NOPs using the OpenBaton GUI (Figure 4-15 The OpenBaton GUI) or the REST API.



6f1451dd-9435-4386-949e-3ea8ea9d598f
27b22cee-a8b8-474c-a24a-a936d3371959
4b8c73c5-dfc4-4051-8ff3-a6c2847826ad
b614e6f8-5cfa-45e0-aeda-2dea66fcc348

Figure 4-15 The OpenBaton GUI

4.5.3.1 Preparation phase

The first step of the Network Service deployment process is the preparation of the descriptors. The required components are VNF packages, VNFDs, and NSDs. The NOP has to write their own VNFDs which describe the structure and functionality of the VNFs of which their Network Service consists. They have to provide the scripts which shall be executed in the individual lifecycle operations (instantiate, configure, start, etc.). The scripts are shell scripts and in these they have the possibility to use parameters which will be made available by the NFVO via environment variables. With this mechanism, the NFVO resolves dependencies and provides configuration parameters defined in the VNFDs or during launching of the Network Service.

4.5.3.2 Onboarding phase

In the onboarding phase the NOPs can upload the created VNF packages to the NFVO. Once the VNF packages of a Network Service have been uploaded to the NFVO, also the VNFDs which are contained in the VNF packages are available. The NSD can then be uploaded to the NFVO or constructed directly in the GUI. The NSD references the VNFDs which have been uploaded together with the VNF packages.

4.5.3.3 Deployment phase

The NSD can be launched using the GUI, which results in the creation of a Network Service Record (NSR). The NSR is a representation of the running Network Service and contains runtime information, for example the IP addresses of the VNF instances.

While launching the Network Service, the NOP has the possibility to change configuration parameters, add SSH keys to the instances of the Network Service and select the point of presence, that is the VIM on which the VNFs shall be deployed. If nothing is specified during launching or in the descriptors, a random VIM will be selected by the NFVO.

The NFVO triggers the creation of the networks based on the virtual link descriptors in the description of the Network Slice Service with the help of the OpenBaton VIM drivers.

4.5.3.4 Operation phase

The user can check the status of the deployed Network Service by fetching the corresponding NSR. The ACTIVE status indicates a successful deployment of a Network Service while ERROR is the status of failed deployments. The VNFs contained in the NSR have statuses themselves so that it is possible to narrow down the origin of the failure. The GUI and REST API provide operations to list and show NSRs.

Furthermore, it is possible to add monitoring support to OpenBaton by adding a monitoring plugin as can be seen in Figure 4-16 or auto scaling support which performs scaling operations based on predefined rules.

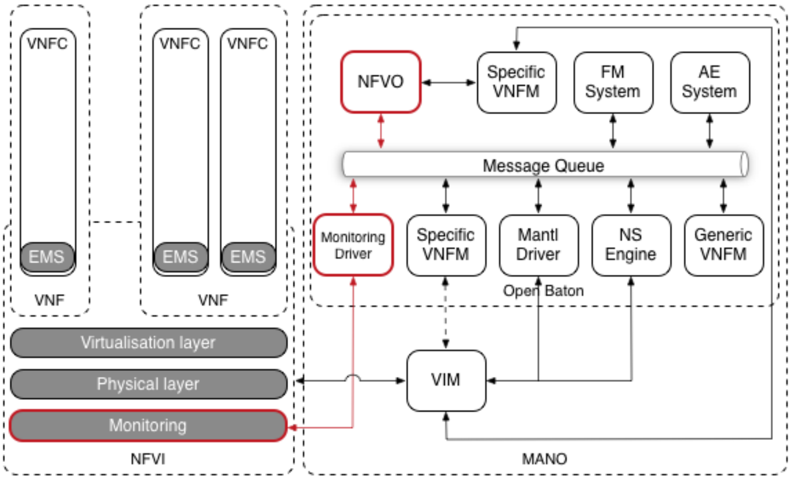


Figure 4-16 Monitoring

Alternatively, scaling operations can be executed manually. The number of instances to which the Network Service can scale is restricted in the VDU definition of the VNFD.

VNF instances can be stopped and restarted, which results in the execution of dedicated lifecycle scripts.

4.5.3.5 Termination phase

After the Network Slice Service has been deployed, it can be removed just as easily using the GUI. The VNFs perform the scripts which are defined in their TERMINATE lifecycle and the Network Service is removed. The resources used for the Network Service are cleaned up.

5 External facing interfaces readiness

This section describes how the facility sites provide access to their customers for accessing the catalogue management and the requests to deploy artifacts

5.1 Norway and UK

Norway and UK solutions are based on Nokia's FlowOne, so they are described here in common.

5.1.1 Catalogue Management interface

Nokia FlowOne's Service Catalogue can expose Service Catalogue Management API based on TMF633. The service catalogue comes with an 'on-boarding manager' and a 'design canvas'. It is suitable for human operation. One can on-board NSDs from NFVOs and then design a slice template and then *publish* it to make it available for ordering.

5.1.2 LCM interface

The Ordering portal is a BSS system and therefore not in the scope of 5G-VINNI, but work is ongoing to find an ordering portal for use. FlowOne will expose Service Ordering API based on TMF641. Anyway, before a sophisticated solution arrives, to start with, CSPs can go manual — meaning, order capture and negotiations with industry verticals can happen over meetings and e-mail communications and then someone, preferably CSP's pre-sales (human) could place the order in FlowOne (E2E-O) using Order Hub. The human operator might also like to have relevant entries in other BSS, e.g., Billing.

5.1.1 Monitoring interface

Monitoring of the network slice delivery order across its phases is available.

5.1.2 Resource Management interface

As an E2E orchestrator, FlowOne inventory will provide views of resources and its eventual relationship with a customer. But, there is not much scope for resource management or resource LCM. This has to come in as LCM requests, via ordering.

5.2 Greece and Spain

Greece and Spain facility sites use the same solution based on OSM and 5GinFIRE portal, thus they are presented in common.

5.2.1 Catalogue Management interface

The Catalogue Management can be done by the facility site portal, which can be accessed by the customers (CSC role) for both the. For the Greece facility site[12] and the Spain facility site[13].

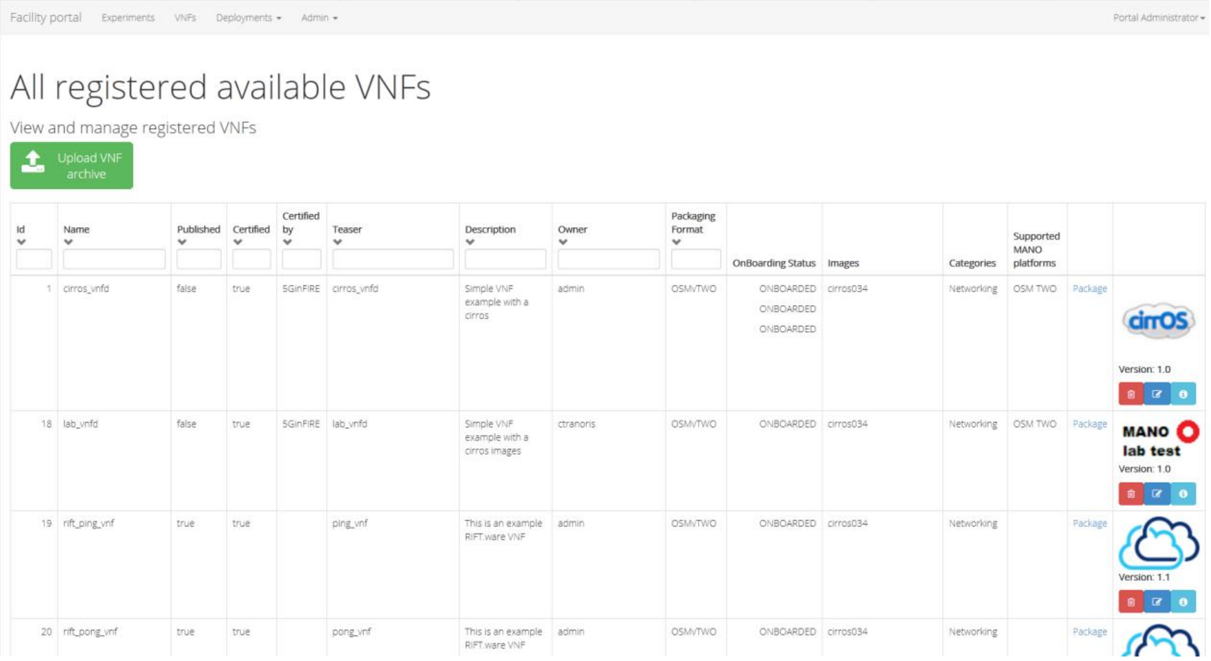
The following artifacts can be managed through the facility portal

- Users
- VNFs/NSDs catalogue
- NFVO endpoints via OSM NBI
- Deployment requests

[12] Patras Facility site portal: https://patras5g.eu

[13] Spain Facility site portal: https://www.5tonic.org/5G-VINNI

The following image provides an example of VNF management on-boarded by users. These VNFs are on-boarded to the facility site's OSM.



All registered available VNFs

View and manage registered VNFs

Id	Name	Published	Certified	Certified by	Teaser	Description	Owner	Packaging Format	OnBoarding Status	Images	Categories	Supported MANO platforms		
1	cirros_vnfd	false	true	5GinFIRE	cirros_vnfd	Simple VNF example with a cirros	admin	OSMvTWO	ONBOARDED ONBOARDED ONBOARDED	cirros034	Networking	OSM TWO	Package	cirrOS Version: 1.0
18	lab_vnfd	false	true	5GinFIRE	lab_vnfd	Simple VNF example with a cirros images	ctranoris	OSMvTWO	ONBOARDED	cirros034	Networking	OSM TWO	Package	MANO lab test Version: 1.0
19	rift_ping_vnf	true	true		ping_vnf	This is an example RIFT.ware VNF	admin	OSMvTWO	ONBOARDED	cirros034	Networking		Package	Version: 1.1
20	rift_pong_vnf	true	true		pong_vnf	This is an example RIFT.ware VNF	admin	OSMvTWO	ONBOARDED	cirros034	Networking		Package	

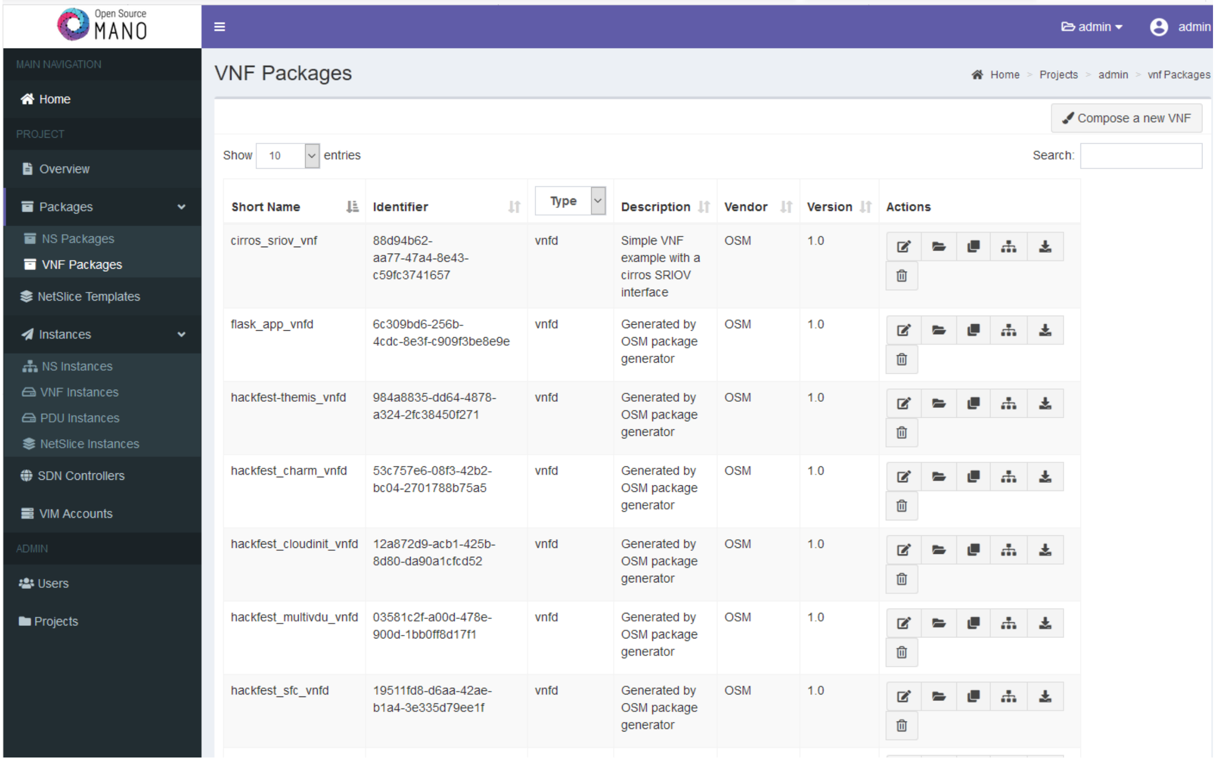
Figure 5-1 VNF Management

Using the same facility site portal, CSC roles can request services, that is to make service orders based on specific service templates

5.2.2 Accessing the facility's OSM

CSP and NOP can access the MANO of the facility site, namely the OSM. For the Patras facility site, OSM is accessible only via a VPN connection. An example of the VNF catalogue is given in Figure 5-2.

For the Spain facility site, OSM is accessible only via a VPN connection.



Short Name	Identifier	Type	Description	Vendor	Version	Actions
cirros_sriov_vnf	88d94b62-aa77-47a4-8e43-c59fc3741657	vnfd	Simple VNF example with a cirros SRIOV interface	OSM	1.0	
flask_app_vnfd	6c309bd6-256b-4cdc-8e3f-c909f3be8e9e	vnfd	Generated by OSM package generator	OSM	1.0	
hackfest-themis_vnfd	984a8835-dd64-4878-a324-2fc38450f271	vnfd	Generated by OSM package generator	OSM	1.0	
hackfest_charm_vnfd	53c757e6-08f3-42b2-bc04-2701788b75a5	vnfd	Generated by OSM package generator	OSM	1.0	
hackfest_cloudinit_vnfd	12a872d9-acb1-425b-8d80-da90a1cfcd52	vnfd	Generated by OSM package generator	OSM	1.0	
hackfest_multivdu_vnfd	03581c2f-a00d-478e-900d-1bb0ff8d17f1	vnfd	Generated by OSM package generator	OSM	1.0	
hackfest_sfc_vnfd	19511fd8-d6aa-42ae-b1a4-3e335d79ee1f	vnfd	Generated by OSM package generator	OSM	1.0	

Figure 5-2 The VNF Catalogue accessed via the deployed OSM in Patras facility site

5.2.3 LCM interface

In the facility site portal some functionalities that facilitate the LCM of various artefacts have been defined, as presented in next sections.

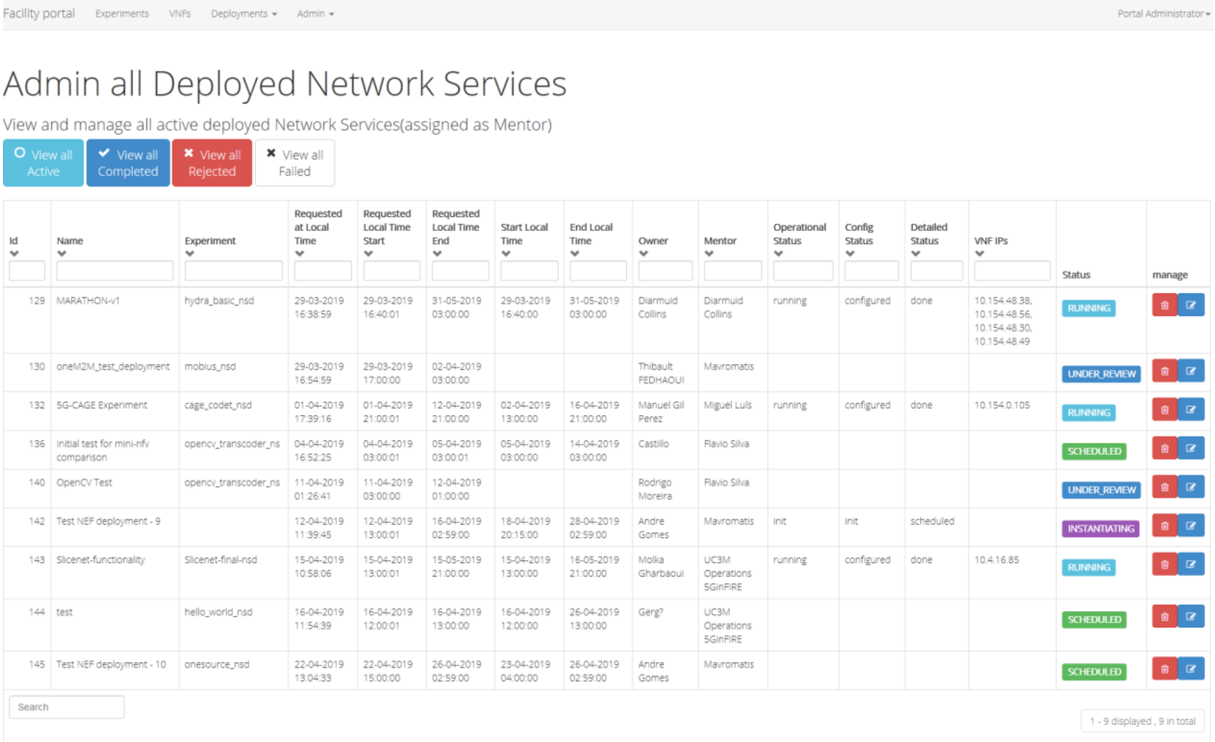
5.2.3.1 Onboarding/Offboarding VNFs/NSDs to MANO

CSC and CSP can Onboard and Offboard VNF and NS descriptors to the facility site in order to be used later for slices requested via Service Ordering. During onboarding a validation against the Information Model is performed.

5.2.3.2 LCM of Network Services

Services deployment requests and scheduling as well VNF placement to multiple VIMs for distributed Network Services is performed from the facility portal towards the OSM via the OSM SOL005 NBI API

Figure 5-3 displays the management and the Lifecycle Management of the deployed Network Services.



Admin all Deployed Network Services

View and manage all active deployed Network Services(assigned as Mentor)

Id	Name	Experiment	Requested at Local Time	Requested Local Time Start	Requested Local Time End	Start Local Time	End Local Time	Owner	Mentor	Operational Status	Config Status	Detailed Status	VNF IPs	Status	manage
129	MARATHON-v1	hydra_basic_nsd	29-03-2019 16:38:59	29-03-2019 16:40:01	31-05-2019 03:00:00	29-03-2019 16:40:00	31-05-2019 03:00:00	Diarmuid Collins	Diarmuid Collins	running	configured	done	10.154.48.38, 10.154.48.56, 10.154.48.30, 10.154.48.49	RUNNING	
130	oneM2M_test_deployment	mobius_nsd	29-03-2019 16:54:59	29-03-2019 17:00:00	02-04-2019 03:00:00			Thibault FEDHAOUI	Mavromatis					UNDER_REVIEW	
132	5G-CAGE Experiment	cage_codet_nsd	01-04-2019 17:39:16	01-04-2019 21:00:01	12-04-2019 21:00:00	02-04-2019 13:00:00	16-04-2019 21:00:00	Manuel Gil Perez	Miguel Luís	running	configured	done	10.154.0.105	RUNNING	
136	Initial test for mini-nfv comparison	opencv_transcoder_ns	04-04-2019 16:52:25	04-04-2019 03:00:01	05-04-2019 03:00:01	05-04-2019 03:00:00	14-04-2019 03:00:00	Castillo	Flavio Silva					SCHEDULED	
140	OpenCV Test	opencv_transcoder_ns	11-04-2019 01:26:41	11-04-2019 03:00:00	12-04-2019 01:00:00			Rodrigo Moreira	Flavio Silva					UNDER_REVIEW	
142	Test NEF deployment - 9		12-04-2019 11:39:45	12-04-2019 13:00:01	16-04-2019 02:59:00	18-04-2019 20:15:00	28-04-2019 02:59:00	Andre Gomes	Mavromatis	init	init	scheduled		INSTANTIATING	
143	Slicenet-functionality	Slicenet-final-nsd	15-04-2019 10:58:06	15-04-2019 13:00:01	15-05-2019 21:00:00	15-04-2019 13:00:00	16-05-2019 21:00:00	Molka Gharbaoui	UC3M Operations 5GinFIRE	running	configured	done	10.4.16.85	RUNNING	
144	test	hello_world_nsd	16-04-2019 11:54:39	16-04-2019 12:00:01	16-04-2019 13:00:00	16-04-2019 12:00:00	26-04-2019 13:00:00	Gerg?	UC3M Operations 5GinFIRE					SCHEDULED	
145	Test NEF deployment - 10	onesource_nsd	22-04-2019 13:04:33	22-04-2019 15:00:00	26-04-2019 02:59:00	23-04-2019 04:00:00	26-04-2019 02:59:00	Andre Gomes	Mavromatis					SCHEDULED	

Figure 5-3 Management and Lifecycle of the deployed Network Services.

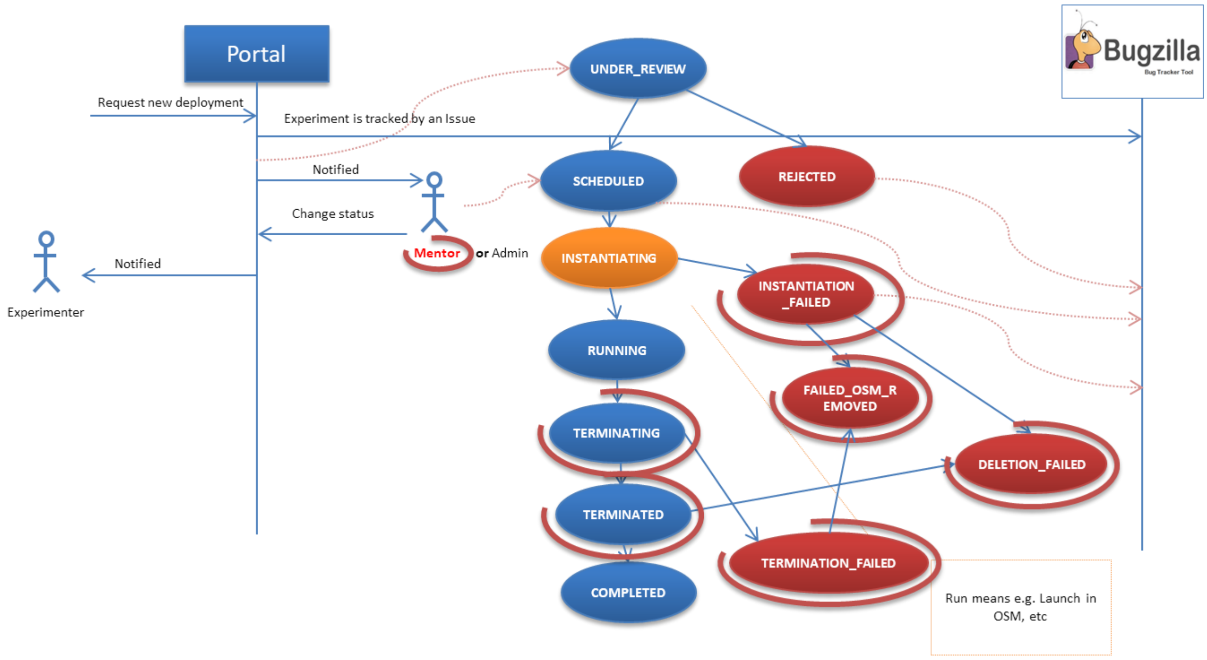


Figure 5-4 NS Deployment Lifecycle

Figure 5-4 displays the NS Deployment Lifecycle. The NS deployment request upon creation get the status “UNDER REVIEW”.

The CSP or NOP (called also mentor of the NS Deployment) needs to examine the deployment and either approve or reject the request.

If the Mentor approves the deployment of the NS the status of the deployment request is changed to SCHEDULED for the NS to be deployed at the specified date/time.

If the mentor does not approve the NS deployment request, then the status of the request is set to REJECTED and this is the final status for this specific request.

If the NS deployment request has status SCHEDULED, at the date/time that the NS must start, the status of the deployment request changes to INSTANTIATING and the instantiation request is started by the portal.

If the instantiation succeeds the NS deployment status changes to RUNNING.

If the instantiation fails, the NS deployment status changes to FAILED. This status means that the instantiation of the deployment has not been successful.

If the deployment request has status RUNNING, at the date/time that the NS must be terminated, the status of the deployment request changes to TERMINATING.

If the termination (tear down) succeeds the status changes to TERMINATED. This state means that the NS deployment was successful, and it was terminated successfully. If the termination fails, the status changes to TERMINATION FAILED.

If the NS deployment request has any of the failed states (FAILED, TERMINATION_FAILED) or the TERMINATED state and the time of the deployment end time has passed, the deletion of the deployment from the OSM is triggered from the portal. If the removal is not successful, the status is changed to DELETION_FAILED and this is the final status of the deployment request.

If the deployment request state is TERMINATED and the removal is successful, the status is changed to COMPLETED and this is the final status of the deployment request.

5.2.4 **Monitoring interface**

For the Patras and Spain facility sites currently, the following mechanisms are used:

5.2.4.1 OSM MON: The monitoring framework of OSM

Current OSM release (FIVE) permits monitoring and visualization of measurements for performance management activities. This is made possible in OSM with the introduction of MON component (see Figure 6.3), which is composed of three main modules:

- **mon-central**: Handles VIM and alarms-related messages in the Kafka bus.
- **mon-collector**: Iterates continuously over the records of VNF Instance and collects metrics using VIM and VCA plugins, then exposes them to be consumed by the Prometheus Time Series Data Base (TSDB).
- **mon-evaluator**: Iterates continuously over alarms and evaluates them. In case of triggering, it sends a notification over the Kafka bus

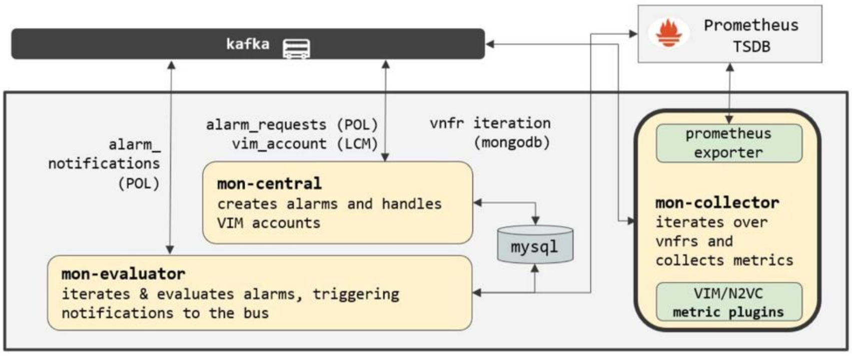


Figure 5-5 OSM's MON component

As seen from Figure 5-5, the MON module featuring monitoring activities is “mon-collector”, which is in charge of collecting metrics specified in the descriptors, including NSTs and NFV-NS/VNF Packages (and contained NSDs/VNFDs). Up to two types of metrics can be grabbed by MON, depending on the source entity which produces them:

- *NFVI metrics*: made available by VIM's Telemetry System. For Spain and Patras facility sites, OpenStack Ceilometer will be used as telemetry system.
- *VNF metrics*: made available by OSM's N2VC. For Spain and Patras facility sites, N2VC will be based on Juju.

Through the OSM Policy Manager (POL) and mon-evaluator components, OSM is able to create, manage and trigger alarms based on infrastructure or VNF events and metrics. Alarms are created as part of auto-scaling rules based on metric thresholds. In future releases, alarms will be associated to any kind of event that is relevant for the proper operation of Network Services.

5.2.4.2 Openstack ceilometer

The Ceilometer Openstack project is a framework for monitoring and metering the OpenStack cloud and is also expandable to suit other needs. A central agent polls utilization statistic for other resources not tied to instances or compute nodes. A compute agent polls metering data and instances statistics from the compute node (primarily the hypervisor).

5.2.5 **Resource Management interface**

For the Patras and Spain facility sites, access to the Openstack based NFVI is allowed only through a specific VPN connection to the datacenter. Through the Openstack Horizon interface, CSP and NOP can manage the tenant accounts and Openstack projects.

Usually, for a specific Vertical use case deployment, a new tenant and project are created at the Openstack level, so resources for this tenant are completely isolated from others. This tenant is then configured in the OSM as a VIM to be used to deploy slices for the specific Vertical.

5.3 Portugal

The SONATA Release 5.0 exposes a set of interfaces to enable external parties to manage the Service Platform. A set of RESTful APIs [SONATA-API] are provided to permit the operation of the Service Platform. In addition, an intuitive Portal [SONATA-PORTAL], running on top of the APIs, is provided to manage key capabilities in a comprehensive and simple manner. This section describes the basics of those interfaces and provides references for further details.

Note: The APIs and Portal full URLs will depend on the IPs (and FQDNs) to be assigned to the SONATA Service Platform.

5.3.1 **Catalogue Management Interface**

The Catalogue Management allows the interactions with the database of the Service Platform to store Descriptors for VNFs (VNFDs) and NSs (NSDs), as well as for Network Slice Templates (NSTs) and other data related to Policies and SLAs. SONATA also stores Packages in the Catalogue, which are basically an envelope that contains multiple VNFDs and NSD to be on-boarded. Table 5-1 shows the REST APIs related to Catalogues. Note that as the on-boarding of ETSI artefacts (VNFs, NSs) is performed using Packages, only the packages APIs allow to store new information in the Catalogue (POST). The other APIs related to VNFs and NSs only allow getting information from the Catalogue (GET). Exceptions are NSTs, which are not on-boarded within packages, but directly using the template (NST).

Table 5-1 SONATA Release 5.0 APIs: Catalogues Management

Feature	Base API	Methods	Description
Packages Catalogue Management	/api/v3/packages	GET,DELETE,OPTIONS,POST	Query, add and delete Packages on the Catalogue
Functions Catalogue Management	/api/v3/functions	GET,OPTIONS	Query information about VNFs in the Catalogues
Services Catalogue Management	/api/v3/services	GET,OPTIONS	Query information about NSs in the Catalogues
Slices Catalogue Management	/api/v3/slices	GET,DELETE,OPTIONS,POST	Query information about NST in the Catalogues

The information of the Catalogue can be visualized in the Portal, listing the available NSDs and NSTs. This information can be seen in the Portal by both the Slice designer and the customer (see Annex I.2).

5.3.2 LCM Interface

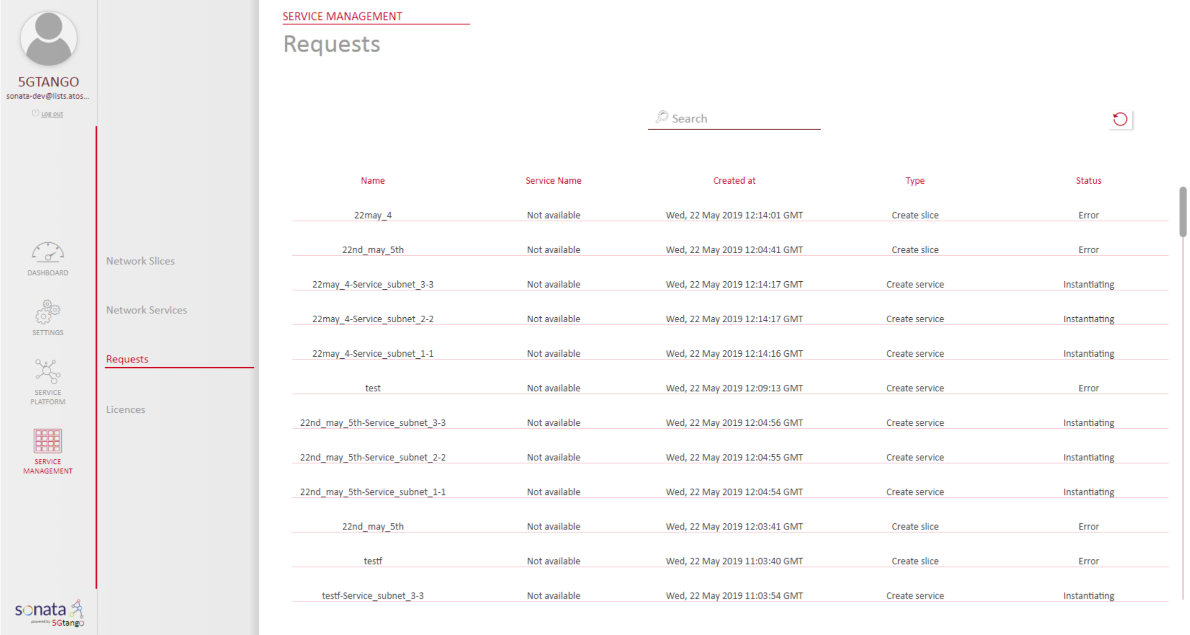
The Lifecycle Management (LCM) capabilities allow the instantiation of instances of NSs and NSIs from NSDs and NSTs, respectively. Other operations may also be included, like the termination, updating, upgrading, scaling, migration, etc. The REST APIs support the instantiation, scaling (only for NSs) and termination. Other operations can be supported in runtime for NSs, but this needs to be triggered by Policies or developed as SSMs (Service Specific Managers).

The request of LCM operations is performed using the base APIs described in Table 5-2. The path and body include the details of the LCM operation to be requested (instantiation, termination, etc.) and the target artefacts (NSs, Network Slices).

Table 5-2 SONATA Release 5.0 APIs: LCM Requests Management.

Feature	Base API	Methods	Description
LCM Requests Management	/api/v3/requests	GET,OPTIONS,POST	Issue and Query Requests for LCM operations on NSs and Slices

The Request of LCM operations can be performed in the Portal as shown in Figure 6-21 and Table 5-3for the Instantiation of NSs and NSIs, respectively, as well as for the termination of NSIs (also available for NSs. In addition to that, the LCM requests issues can be listed, as depicted in Figure 5-6, showing the status of the request (new, instantiating, error, ready, terminated, etc.).



SERVICE MANAGEMENT

Requests

Name	Service Name	Created at	Type	Status
22may_4	Not available	Wed, 22 May 2019 12:14:01 GMT	Create slice	Error
22nd_may_5th	Not available	Wed, 22 May 2019 12:04:41 GMT	Create slice	Error
22may_4-Service_subnet_3-3	Not available	Wed, 22 May 2019 12:14:17 GMT	Create service	Instantiating
22may_4-Service_subnet_2-2	Not available	Wed, 22 May 2019 12:14:17 GMT	Create service	Instantiating
22may_4-Service_subnet_1-1	Not available	Wed, 22 May 2019 12:14:16 GMT	Create service	Instantiating
test	Not available	Wed, 22 May 2019 12:09:13 GMT	Create service	Error
22nd_may_5th-Service_subnet_3-3	Not available	Wed, 22 May 2019 12:04:56 GMT	Create service	Instantiating
22nd_may_5th-Service_subnet_2-2	Not available	Wed, 22 May 2019 12:04:55 GMT	Create service	Instantiating
22nd_may_5th-Service_subnet_1-1	Not available	Wed, 22 May 2019 12:04:54 GMT	Create service	Instantiating
22nd_may_5th	Not available	Wed, 22 May 2019 12:03:41 GMT	Create slice	Error
testf	Not available	Wed, 22 May 2019 11:03:40 GMT	Create slice	Error
testf-Service_subnet_3-3	Not available	Wed, 22 May 2019 11:03:54 GMT	Create service	Instantiating

Figure 5-6 SONATA Release 5.0 Portal: LCM Requests Listing.

5.3.3 Monitoring Interface

The Monitoring capabilities allow the management of monitoring, setting and getting data related to VNFs, NSs and Network Slices. This monitoring data is collected by the Monitoring component and can be delivered by the Service Platform to external parties. In addition to data, the Service Platform can also deliver some charts in a graphical view (charts). Those features can be accessed using the REST APIs described in Table 5-4.

Table 5-3 SONATA Release 5.0 APIs: Monitoring Management.

Feature	Base API	Methods	Description
Data Management	/api/v3/monitoring/data	GET,OPTIONS,POST	Query data from Monitoring
Graphics Management	/api/v3/monitoring/graphics	GET,OPTIONS,POST	Query graphics (charts) from Monitoring

The key Monitoring data can also be visualized in a chart format on the Portal, through the *Dashboard* menu. Figure 5-7 shows an example of the Portal showing some monitoring data.

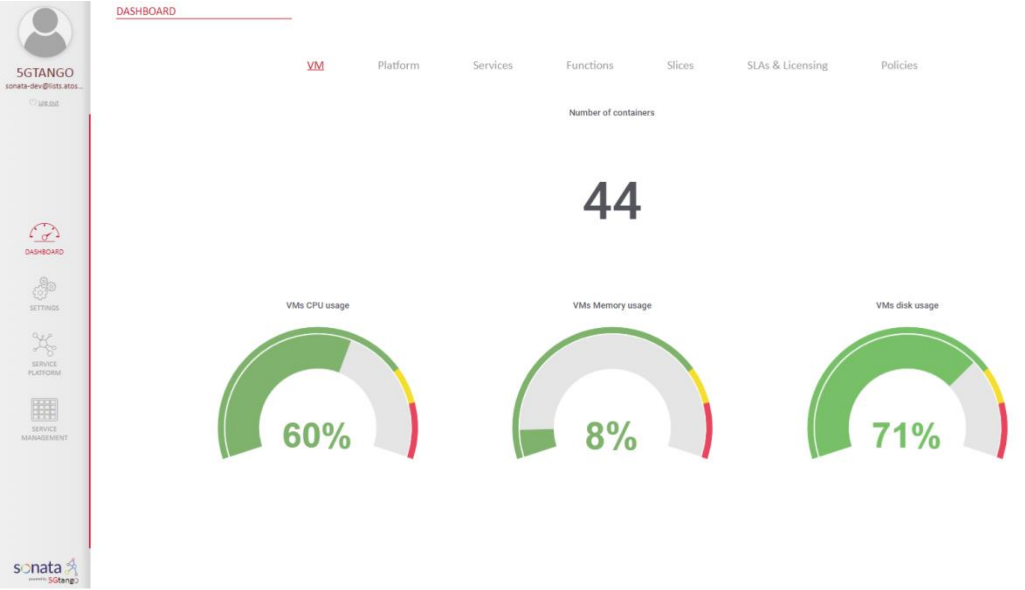
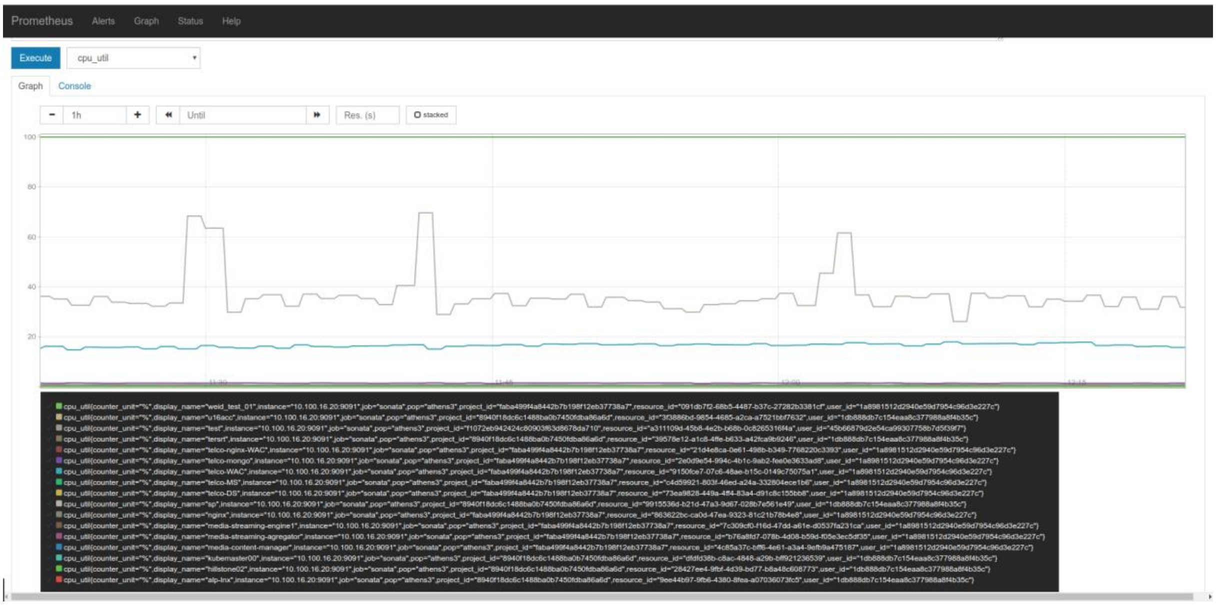


Figure 5-7 SONATA Release 5.0 Portal: Dashboard.

The full Monitoring details can be visualized by accessing directly to the Monitoring Management component of the Service Platform, which is based on the *Prometheus* software (see Figure 5-8 for an example



```
cpu_util{counter_unit="%",display_name="weid_test_01",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="091db7f2-68b5-4487-b37c-27282b3381cf",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="u16acc",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="8940f18dc6c1488ba0b7450fdba86a6d",resource_id="3f3886bd-9854-4685-a2ca-a7521bbf7632",user_id="1db888db7c154eaa8c377988a8f4b35c"}
cpu_util{counter_unit="%",display_name="test",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="f1072eb942424c80903f63d8678da710",resource_id="a311109d-45b8-4e2b-b68b-0c8265316f4a",user_id="45b66879d2e54ca99307758b7d5f39f7"}
cpu_util{counter_unit="%",display_name="tersrt",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="8940f18dc6c1488ba0b7450fdba86a6d",resource_id="39578e12-a1c8-4ffe-b633-a42fca9b9246",user_id="1db888db7c154eaa8c377988a8f4b35c"}
cpu_util{counter_unit="%",display_name="telco-nginx-WAC",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="21d4e8ca-0e61-498b-b349-7768220c3393",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="telco-mongo",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="2e0d9e54-994c-4b1c-9ab2-fee0e3633ad8",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="telco-WAC",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="9150fce7-07c6-48ae-b15c-0149c75075a1",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="telco-MS",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="c4d59921-803f-46ed-a24a-332804ece1b6",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="telco-DS",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="73ea9828-449a-4ff4-83a4-d91c8c155bb8",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="sp",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="8940f18dc6c1488ba0b7450fdba86a6d",resource_id="9915536d-b21d-47a3-9d67-028b7e561e49",user_id="1db888db7c154eaa8c377988a8f4b35c"}
cpu_util{counter_unit="%",display_name="nginx",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="863622bc-ca0d-47ea-9323-81c21b78b4e8",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="media-streaming-engine1",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="7c309cf0-f16d-47dd-a61e-d0537fa231ca",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="media-streaming-agregator",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="b76a8fd7-078b-4d08-b59d-f05e3ec5df35",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="media-content-manager",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="faba499f4a8442b7b198f12eb37738a7",resource_id="4c85a37c-bff6-4e61-a3a4-9efb9a475187",user_id="1a8981512d2940e59d7954c96d3e227c"}
cpu_util{counter_unit="%",display_name="kubemaster00",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="8940f18dc6c1488ba0b7450fdba86a6d",resource_id="dfdfd38b-c8ac-4848-a29b-bff921236539",user_id="1db888db7c154eaa8c377988a8f4b35c"}
cpu_util{counter_unit="%",display_name="hillstone02",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="8940f18dc6c1488ba0b7450fdba86a6d",resource_id="28427ee4-9fbf-4d39-bd77-b8a48c608773",user_id="1db888db7c154eaa8c377988a8f4b35c"}
cpu_util{counter_unit="%",display_name="alp-lnx",instance="10.100.16.20:9091",job="sonata",pop="athens3",project_id="8940f18dc6c1488ba0b7450fdba86a6d",resource_id="9ee44b97-9fb6-4380-8fea-a07036073fc5",user_id="1db888db7c154eaa8c377988a8f4b35c"}
```

Figure 5-8 SONATA Release 5.0 Portal: Monitoring Management (Prometheus).

5.3.4 Resource Management Interface

The Resource Management capabilities allow the NOP to manage the VIMs and WIMs that can be used to deploy VNFs, NSs and NSIs. Those features are accessed using the REST APIs described in Table 5-4.

Table 5-4 SONATA Release 5.0 APIs: Resource Management.

Feature	Base API	Methods	Description
VIM Management	/api/v3/settings/vims	GET,DELETE,PATCH,OPTIONS,POST	Query, add, update and delete VIMs
WIM Management	/api/v3/settings/wims	GET,DELETE,PATCH,OPTIONS,POST	Query, add, update and delete WIMs

5.3.5 Repositories Interface

The Repositories capability allows the access to the Repositories database where the records of the instances are stored. This database stores records with all the details of VNFs, NSs and NSIs (VNFRs, NSRs, NSIRs) instances. Those databases have information about running instances as well as historical data. Table 5-5 shows the REST APIs that enable access to the records from Repositories. Records are stored on the Repositories as part of the LCM operations (e.g. instantiation, termination, etc.).

Table 5-5 SONATA Release 5.0 APIs: Repositories.

Feature	Base API	Methods	Description
Functions Repositories	/api/v3/records/functions	GET,OPTIONS	Query VNF records (VNFR) on the Repositories
Services Repositories	/api/v3/records/services	GET,OPTIONS	Query NS records (NSR) on the Repositories
Slices Repositories	/api/v3/records/slices	GET,OPTIONS	Query NSI records (NSIR) on the Repositories

5.3.6 SLA Management Interface

The SLA Management capability allows the management of the SLAs associated to a NS. The SLA feature enables the creation of SLA Templates associated to particular NSs, describing Service Level Objectives (SLOs), which define particular requirements in terms of bandwidth, latency, availability, etc. Once the instantiation of a particular NS takes place, the customer selects an SLA (among the available for that NS) and an SLA agreement is automatically created. From this moment on, the SLA Manager will check whether SLA violations occur (via monitoring). The SLA Manager also takes care about NS licensing. Table 5-6 shows the REST APIs that allow the management (CRUD) SLA Templates, the listing of automatically created SLA agreements and the enumeration of SLA violations, if they exist.

Table 5-6 SONATA Release 5.0 APIs: SLA Management

Feature	Base API	Methods	Description
SLA Template Management	/api/v3/slas/templates	GET,DELETE,OPTIONS,POST	Query, add and delete SLA Templates
SLA Agreements Management	/api/v3/slas/agreements	GET,DELETE,OPTIONS,POST	Query, add, update and delete SLA Agreements
SLA Violations Query	/api/v3/slas/violations	GET,OPTIONS	Query Violations occurred a given NS and SLA Agreement
Licensing Management	/api/v3/slas/licenses	GET,DELETE,OPTIONS,POST	Query, add, update and delete SLA-based Licenses
SLA Configurations Query	/api/v3/slas/configurations	GET,OPTIONS	Query SLA Miscellaneous configurations (flavours, SLOs, etc.)

The management of SLA templates and the visualization of SLA agreements and violations can also be done via Portal. Licenses can also be created via Portal (by the customer role) in the *Service Management* menu and then *Licenses*.

5.3.7 Policy Management Interface

The Policy Management capability allows the management of the Policies that rule (1) placement, the location where Services (NSs) are placed, and (2) runtime, how NSs behave when they are running. In runtime, based on the provisioned policies and rules, and with the access to monitoring data, some actions like scaling, healing, or updating can be automatically triggered. Placement policies are generic for all services (e.g. what VIM has priority), while runtime policy are for a particular NS. Table 5-7 shows the REST APIs which allow the management (CRUD) of placement rules as well as runtime rules.

Table 5-7 SONATA Release 5.0 APIs: Policy Management.

Feature	Base API	Methods	Description
Runtime Policy Management	/api/v3/policies	GET,DELETE,OPTIONS,PATCH,POST	Query, add, update and delete Runtime Policies
Placement Policy Management	/api/v3/policies/placement	GET,DELETE,OPTIONS,PATCH,POST	Query, add, update and delete Placement Policies

The management of policies can also be done via Portal, both for placement and runtime policies. In addition to that, the list of "generated actions" can be consulted, listing all the actions suggested/enforced by the Policy Manager (e.g. scale, heal, etc.).

5.3.8 Authentication and Authorization Management Interface

The Authentication and Authorization Management capability allows the management of users and roles that can have access both to the REST APIs and the Portal. This allows an administrator to define what users belong to particular roles and, this way, assign permissions to get access to particular features and actions. Table 5-8 shows the REST APIs related to management users and roles and respective authentication and authorization.

Table 5-8 SONATA Release 5.0 APIs: Authentication and Authorization Management.

Feature	Base API	Methods	Description
Manage Users	/api/v3/users	GET,DELETE,OPTIONS,PATCH,POST	Query, add, update and delete users
Manage Roles	/api/v3/users/roles	POST,GET,PATCH	Query, add, update and delete roles
Authenticate and Authorize Users	/api/v3/users/sessions	POST	Authenticate and authorize users

The authentication and authorization of users and roles is not supported by now in the Portal.

5.4 Germany-Munich

As an experimental facility site a specific external facing interface is not provided. The virtualization infrastructure is pre-configured in order to realize the scenarios required by the vertical customers.

5.5 Germany-Berlin, Luxemburg

In general, the 5G-VINNI Berlin and Luxemburg facility sites are addressing experimental level slice deployments and will not expose towards the outside a comprehensive set of interfaces. This is because the two facilities are not meant to offer active services. Instead being on a use case by use case deployed, the main interfaces are related to the assessment of the service.

5.5.1 Catalogue Management interface

A marketplace for open source components is offered through the the OpenBaton Market Place[14]. Please note that the essential components related to the 5G radio, 5G core network as well as the components required to connect edge nodes to central locations as well as the specific application enablers designed to address the use case needs (beyond the goal of 5G-VINNI) are not present into the open catalogue. Instead they are to be configured according to a requirements discussion made with the use case owners.

5.5.2 Active slices interfaces

For understanding the status of a specific slice instance from an infrastructure perspective, the OpenBaton orchestrator is offering a set of GUIs towards the tenants

The OpenBaton dashboard, as illustrated in Figure 5-9 provides an overview of the deployed network tenant from an infrastructure perspective. This includes the allocation of compute, memory and public IP addresses for the tenant component as well as the number of instances of the different components. The dashboard is filled with dynamic monitoring information including the momentary degree of occupation of the specific resources.

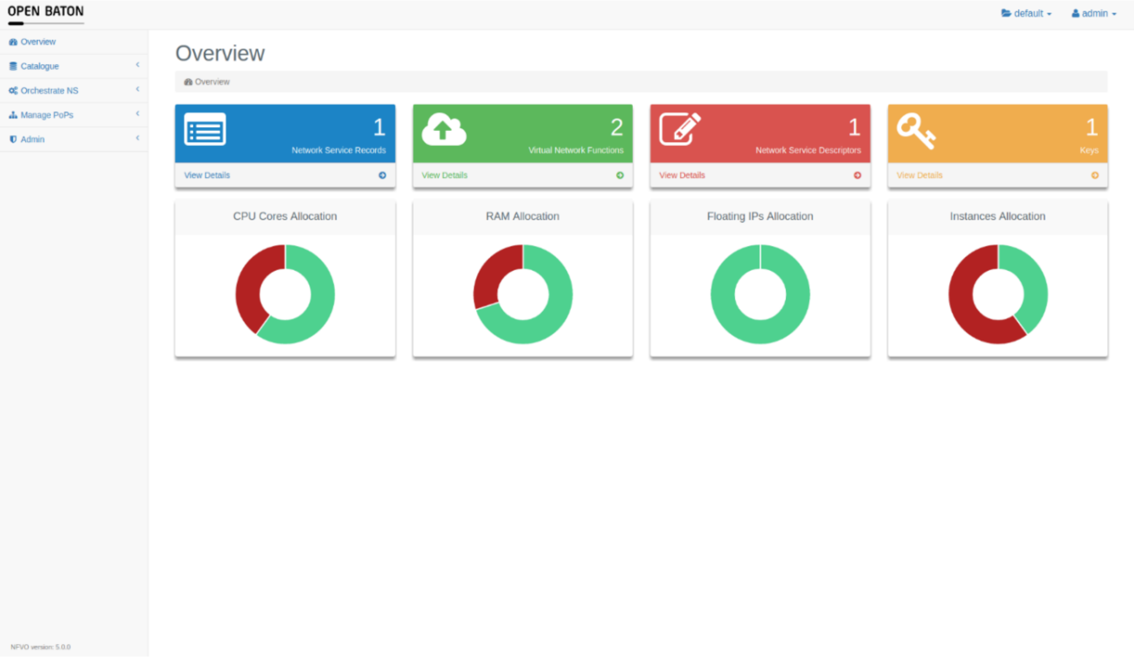


Figure 5-9 OpenBaton DashBoard

5.5.3 Monitoring interface

The Berlin and Luxemburg facility have two monitoring mechanisms, one at the infrastructure level, exposed through the NFV infrastructure and one at the service level, exposed directly through one of the network functions within the slice.

[14] Open Baton Marketplace: https://openbaton.github.io/documentation/marketplace/

5.5.3.1 Infrastructure level monitoring

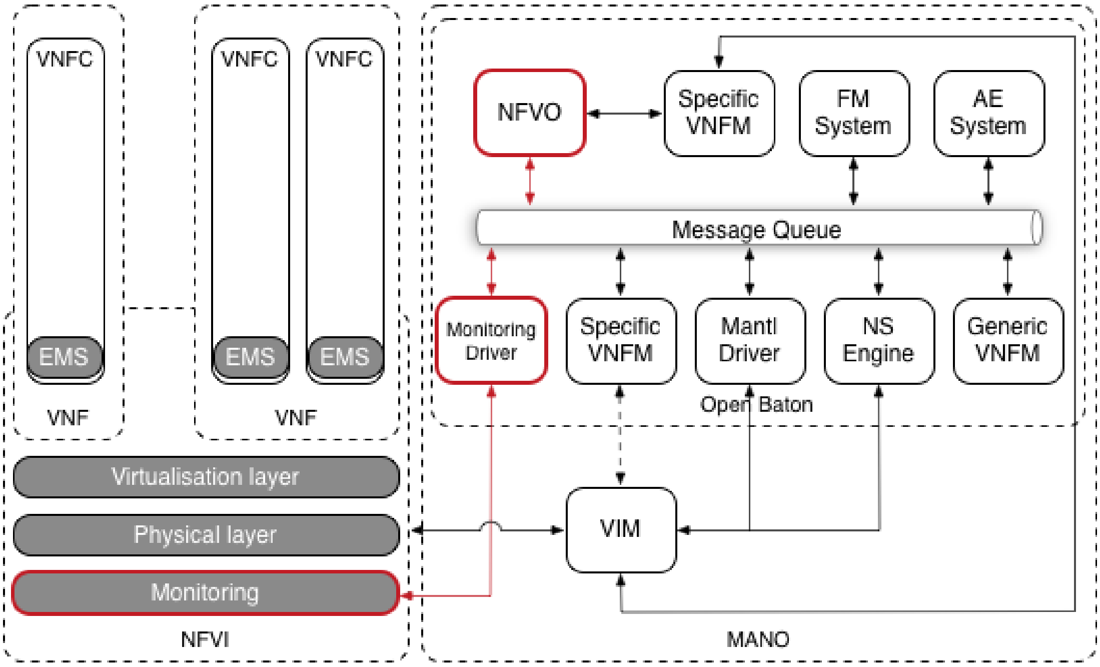


Figure 5-10 OpenBaton Monitoring

OpenBaton interfaces with monitoring systems using a plugin mechanism. The plugin mechanism allows OpenBaton to easily use multiple monitoring systems. The figure 5.7 shows how the monitoring plugin communicate with the NFVO. As you can see from the picture above the Monitoring Plugin (or Driver) is an intermediate between OpenBaton and the specific monitoring system. The Monitoring Plugin implements a Standard interface in order to be used by OpenBaton. An example of monitoring plugin is Zabbix plugin which allows OpenBaton to use Zabbix Server.

The Zabbix plugin is an open source project providing a reference implementation of two interfaces of the VIM, based on the ETSI NFV MANO specification.

The two interfaces are:

- VirtualisedResourceFaultManagement
- VirtualisedResourcePerformanceManagement

In particular with the Zabbix plugin you can create/delete items, trigger and action on-demand.

Some of the benefits introduced by the usage of such plugin:

1) Make the consumers (NFVO, VNF managers, Fault Management System, AutoScaling Engine) independent of the monitoring system.
2) The communication between the consumers and Zabbix Plugin is JSON based, so the consumers can be written in any languages.

The values of the items are cached and updated periodically in order to avoid to contact the Zabbix Server each time a specific metric is required.

5.5.3.2 Slice-level monitoring

The slice level monitoring provides the tenant with a live overview of the status of the network functions within the slice as well as with historical data across a very large number of time series, which include the usage of the resources as well as the specific 5G network subscriber and data session status.

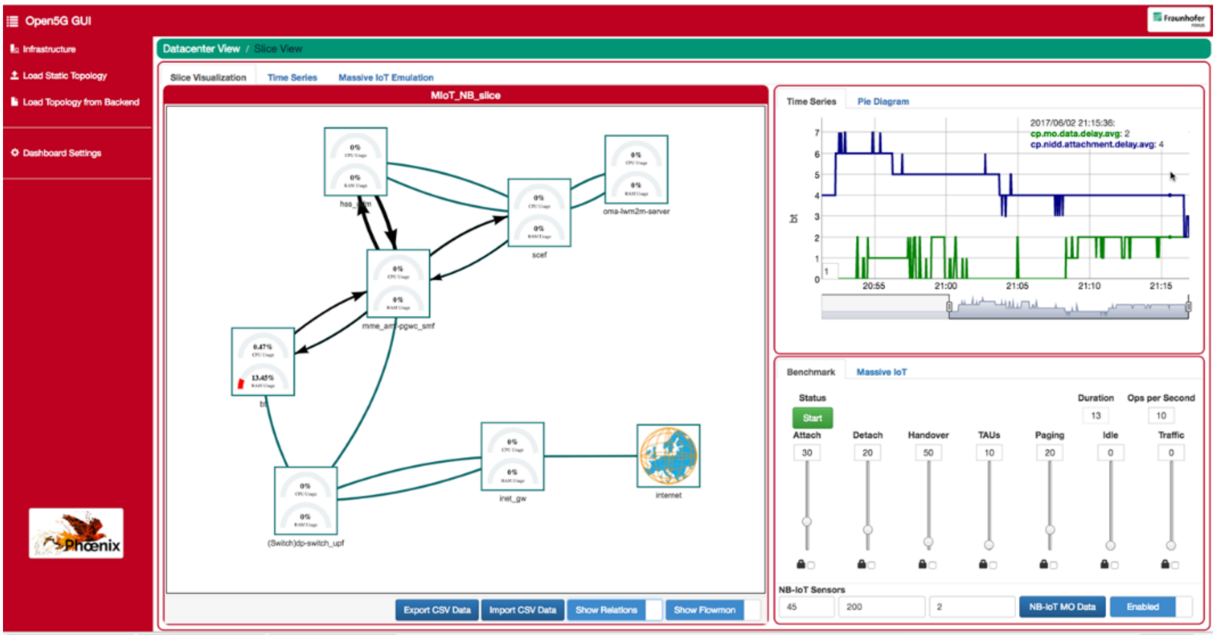


Figure 5-11 Sample of the Slice level monitoring GUI

Each of the monitored network functions include a Prometheus agent to be able to transfer the monitored data to the monitoring server. For network functions which were not developed by the 5G-VINNI partners and not open source, the monitoring agent was installed as part of the specific monitoring tools. Same approach will be taken for the 5G base stations which will be acquired at a later date.

It was considered that for scalability reasons as well as for the privacy of the monitored information, each slice has its own monitoring system. For this, the monitoring information is accumulated into a Prometheus server which is deployed together with the GUI as part of the slice and could be accessed through a public IP by the CSC and the CSP.

The slice-level monitoring is used only for the accumulation of data and for understanding the status of the system and not for immediate actuation.

6 Day to Day operations

Any 5G-VINNI role from those identified in Section 2 should be able to request assistance in case of problem or new request. As the goal of the project is to provide CSC with network slices services to be used for experimentation environments, it is highly important to report issues and track them until their resolution. Thus, both CSCs and other roles within the 5G-VINNI facility site (i.e. CSP, NOP, etc.) can be made aware of what is happening, and get feedback accordingly. Issues in 5G-VINNI include not only faults/bugs, but also requests. For example, for every new network slice service ordering, it will be tracked by the system via an issue. All related stakeholder will be notified via emails. Moreover, the facility services will be enabled to trigger issues automatically and notify interested parties.

6.1 Norway and UK

6.1.1 Issue management entry point

In release 0, no support for issue management is included. Release 1 can be re-scoped with Order Hub. It will provide a lean UI for ordering. At least, the corrective measure can be entered, after issue is captured and a possible solution is determined such that the E2E orchestrator can fix it, in its scope of operation.

6.1.2 Monitoring entry point and Data assurance

Monitoring of the network slice delivery order, across its delivery phases can be monitored.

At the moment, Nokia FlowOne can integrate with 3rd party Assurance solutions and support orchestrating the corrective measures. Anyway, this is not in scope for VINNI.

6.2 Greece and Spain

6.2.1 Issue management entry point

For issue management support, Greece and Spain facility site will rely on Bugzilla[15]. Bugzilla is a ticketing tool that allows issue reporting and tracking via tickets. Figure 6-1 shows the overall Bugzilla architecture and how this ticketing tool shall interact with the Portal and other monitoring tools.

[15] Bugzilla at Patras facility site: https://patras5g.eu/bugzilla

Bugzilla at Spain facility site: https://5tonic.org/5g-vinni/bugzilla

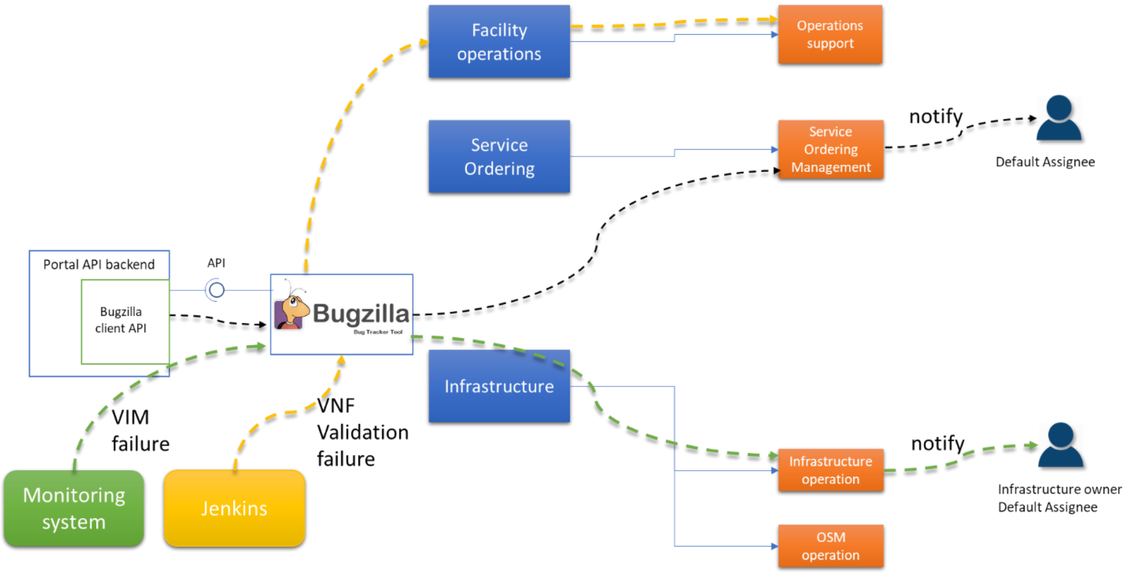


Figure 6-1 Bugzilla working environment

The workflow that has been selected for 5G-VINNI is shown in Figure 6-2. This workflow is the one that Bugzilla uses by default, i.e. to handle defects. As said earlier, Bugzilla tickets in 5G-VINNI will be not only used for defaults/bugs, but also for general requests.

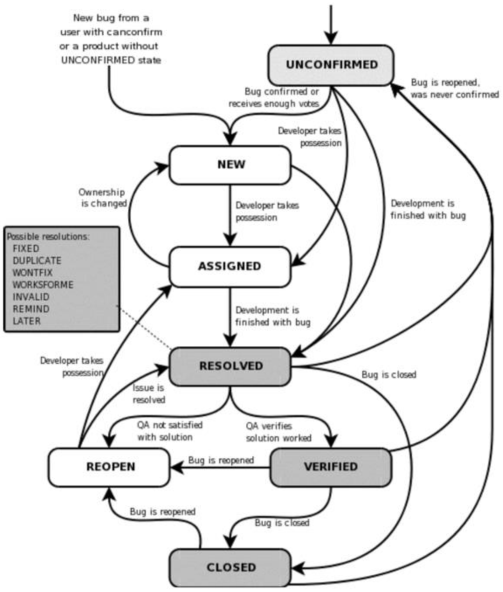

New bug from a user with canconfirm or a product without UNCONFIRMED state
UNCONFIRMED
Bug confirmed or receives enough votes
Bug is reopened, was never confirmed
Developer takes possession
NEW
Ownership is changed
Developer takes possession
Development is finished with bug
Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID
REMIND
LATER
ASSIGNED
Development is finished with bug
Developer takes possession
RESOLVED
Bug is closed
Issue is resolved
QA not satisfied with solution
QA verifies solution worked
REOPEN
Bug is reopened
VERIFIED
Bug is reopened
Bug is closed
CLOSED

Figure 6-2 Bugzilla workflow

Bugzilla defines a severity field in the reported ticket to inform interested parties of any request or detected faults, so priority can be introduced for their handling. By default, this field can have the following values: blocker, critical, major, normal, minor, trivial, and enhancement. For 5G-VINNI release 0, Spain and Patras facility sites adopt a severity level model to be used for notification in case of assistance request. This model is a simplified version of default Bugzilla model, with the severity field taking three possible values:

- *Critical*: a request is “critical” when it leads to the inoperability of the service and no fallback or workaround solution is available.
- *Major*: a request is “major” when it leads to a limitation of the functionalities or the performances of the service, or to the necessity to use fall-back mechanisms or workarounds.

- *Minor*: a request is "minor" when it has no operational impact but leads to difficulties to operate the service.

Fault supervision for VNFs and NSs is performed by OSM, thus alarms can be escalated to CSCs through Bugzilla.

Alarm management is automated based on monitoring and collected data (events and metrics). Alarms can be created based on metric thresholds or associated with events relevant for the proper operations of NSs. Alarm management plays a central role in auto-scaling behaviour too. Monitoring data (metrics) is stored and correlated in a local, highly scalable and performant Time Series Database to enhance lifecycle automation based on metrics aggregation and correlation, independent of their source. Figure 6-3 demonstrates how Fault Management is implemented.

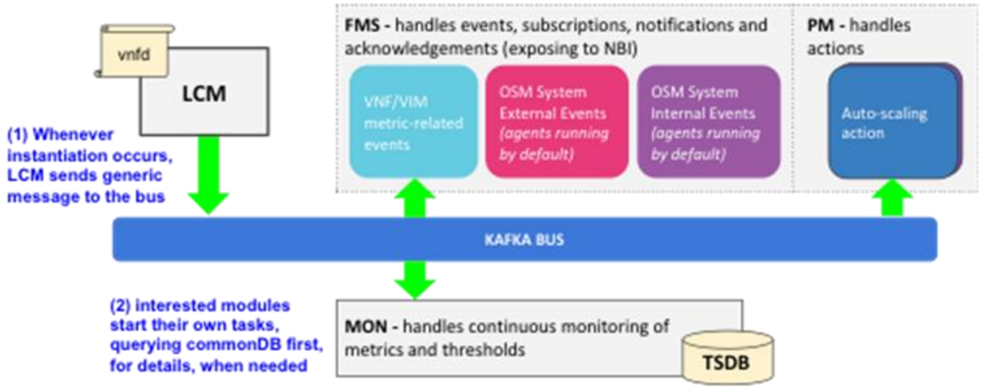

vnfd
LCM
(1) Whenever instantiation occurs, LCM sends generic message to the bus
FMS - handles events, subscriptions, notifications and acknowledgements (exposing to NBI)
VNF/VIM metric-related events
OSM System External Events (agents running by default)
OSM System Internal Events (agents running by default)
PM - handles actions
Auto-scaling action
KAFKA BUS
(2) interested modules start their own tasks, querying commonDB first, for details, when needed
MON - handles continuous monitoring of metrics and thresholds
TSDB

Figure 6-3 OSM Fault Management Architecture

6.2.2 Monitoring entry point and Data assurance

For Greece and Spain facility sites, the "mon-collector" described in 5.2.4 will take the lead in performance monitoring activities, collecting NFVI metrics from OpenStack Ceilometer and VNF metrics from OSM's N2VC module (Juju). Once collected, the "mon-collector" performs two activities: *i)* puts these metrics in the Kafka bus, from which they can be read by any authorized entity who requires it, and *ii)* exports them to Prometheus TSDB, so they can be stored in its data base, and -if needed - presented in a user-friendly manner using Grafana visual tool. The complete description of this workflow is shown in Figure 6-4.

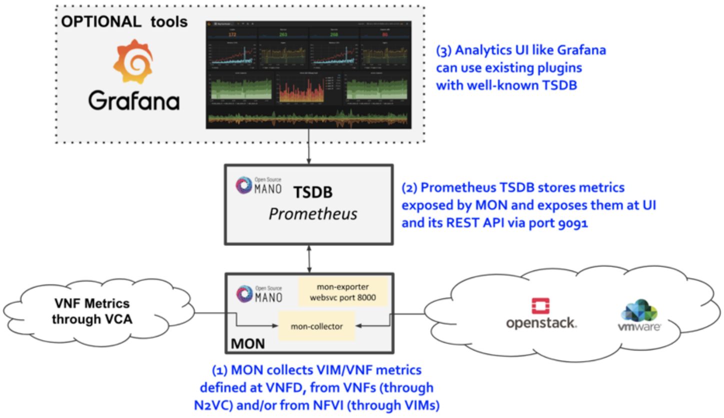

OPTIONAL tools
Grafana
(3) Analytics UI like Grafana can use existing plugins with well-known TSDB
Open Source MANO
TSDB
Prometheus
(2) Prometheus TSDB stores metrics exposed by MON and exposes them at UI and its REST API via port 9091
VNF Metrics through VCA
mon-exporter websvc port 8000
mon-collector
MON
openstack.
vmware
(1) MON collects VIM/VNF metrics defined at VNFD, from VNFs (through N2VC) and/or from NFVI (through VIMs)

Figure 6-4 MON collector working environment

For the Greece and Spain facility sites, monitoring can be performed by the CSP/NOP and in some cases by CSC, for those cases where selected exposure level is 2 or higher.

Additionally, for the Spain facility site the monitoring system may be extended with the implementation of a model-driven telemetry solution as explained in Annex D.2

6.3 Portugal

Currently there is no issue management and monitoring subsystem.

6.4 Germany-Munich

Day to day issues and operation is done based on manual troubleshooting. Due to the nature of our experimental facility, it is not feasible or required to have a big issue management and monitoring subsystem.

6.5 Germany-Berlin, Luxemburg

6.5.1 Issue management entry point

For the 5G-VINNI Berlin experimental facility it is highly important to have a very short feedback loop with the use case owner of the service. For this, two mechanisms are set in place to assure that the specific slice deployed for the use case is further advancing through experimentation towards the specific use case requirements.

1) Analysis of the monitored information – a very large amount of information is monitored within the slices. As this is not a commercial service, or even open to the public, the information will be provided to the CSP for further analysis. The information will be displayed using the GUIs presented in the previous section and will be analysed in common by the CSC and the CSP in order to determine whether the use case requirements were met and/or optimizations are needed.
2) Bug tracing – each deployment of the Open5GCore and associated radio, backhaul and use case enablers is directly bound to a dedicated ticketing system. The ticketing system enables the reporting of misconfigurations and bugs. The ticketing system is directly forwarding the specific issues to the developers of the network functions resulting into an immediate feedback. Through this, optimizations and fixes can be realized in a matter of work days.

Both these solutions are meant to provide the means to analyse and to optimize the specific slice deployment between experimentation phases, in order to further customize towards the use case needs the deployments.

Furthermore, the two mechanism are proposed to be used also within the experimentation phases to check the viability of the accumulated experimentation data and to further shortcut the loop of experiments failures in order to be able to have a continuous service for the specific selected duration. This is considered essential for most of the use cases deployed, as the access to the specific use case site may be limited to a number of days and thus, a rebooting of experiments without issues is required.

6.5.1 Monitoring entry point and Data assurance

For the Berlin and Luxemburg experimental facility sites, the infrastructure will be continuously monitored using the infrastructure monitoring tools described in the previous section.

As it is foreseen that experimentation will take place only in specific time intervals, the infrastructure management will not transmit specific alarms. Instead this will be checked by the infrastructure administrator before the experimentation as well as continuously during the experimentation, to assure that the infrastructure level is suitable for the specific experiments.

References

[D3.1] "5G-VINNI deliverable D3.1 – Specification of services delivered by each of the 5G-VINNI facilities".

[D1.3] "5G-VINNI deliverable D1.3 – Design for systems and interfaces for slice operation v1".

[EVE012] ETSI GR NFV-EVE 012, "Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architectural Framework", v3.1.1, December 2017.

[OpenAPI] https://projects.tmforum.org/wiki/display/API/Open+API+Table?_ga=2.176185127.387049242.1554322365-540138824.1552864878

[NFV003] ETSI GS NFV 003, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", v1.4.1, August 2018.

[OSM] http://osm-download.etsi.org/ftp/osm-doc/nst.html#

[3GPP23.501] 3GPP TS 23.501: System Architecture for the 5G System.

[3GPP28.530] 3GPP TS 28.530: Management and orchestration; Concepts, use cases and requirements

[OpenBaton] OpenBaton¸ https://openbaton.github.io

[OpenBaton-MP] OpenBaton Market Place, http://marketplace.openbaton.org/#/

[OpenBaton-VNFPM] OpenBaton VNF Package Management, https://openbaton.github.io/documentation/vnf-package/

[OpenBaton-VNFD] OpenBaton VNF Descriptor, https://openbaton.github.io/documentation/vnf-descriptor/

[SONATA-API] https://github.com/sonata-nfv/tng-api-gtw

[SONATA-PORTAL] https://github.com/sonata-nfv/tng-portal

Annex A Norway Facility

A.1 Supported Roles of 5G-VINNI Norway facility

A.1.1 Communication Service Customer (CSC)

The CSC role is supported by the 5G-VINNI Norway facility. Vertical customers are considered as CSC and consume the communications services. These vertical customers (e.g. ICT-19 funded projects) need access to browse, order, and receive communications services from the Norway facility. To support CSCs for requesting and receiving the services, the Norway facility develops

- GUI for browsing the service catalogue, ordering the service, and specifying the requirements/attributes and adds-on services such as 3rd party VNF support. In the short-term, Nokia FlowOne has a lean UI for service order submission. In Release 1, orders can be placed manually by CSP via the lean UI from which a Catalogue item (Customer Facing Service) can be selected and an order can be placed. This process is separate from order capture or C-P-Q (Configure, Price, Quote) sales cycle. In the long-term, it is expected (tentative) for 5G-VINNI to develop a full-fledged CRM/COM application that should be matured to handle the B2B ordering of network slice services
- GUI/APIs to expose capabilities of Level 1, i.e., the customers have access to general slice features and network slice service management.

  Order capture or, C-P-Q cycles between CSP and CSC will be manual, possibly by means of meetings, calls, e-mail communications, etc. Order submission to Nokia FlowOne will be manual, essentially, submitting an XML order payload (Nokia will provide the templates) via some SOAP client application, e.g., SOAP UI, etc. Nokia FlowOne has a lean UI to carry out this order submission process. But, this is not in scope for VINNI. Hence, possibilities of having that in will be assessed. Earliest in release 1 scope.

A.1.2 Communication Service Provider (CSP)

The CSP role is supported by the 5G-VINNI Norway facility. The role is taken by Telenor, with support of Nokia. Telenor and Nokia jointly design, build, and operate the network slices services and provide them to CSCs. The CSP maintains the customer portal and handles the provisioning issues through the daily operations.

A.1.3 Network Operator (NOP)

The NOP role is supported by the 5G-VINNI Norway facility. The role is taken by Telenor that is responsible for providing the network services required to support the communications services. Telenor, as a NOP, will be in charge of the design, deployment and operation of network services at the E2E infrastructure. Telenor has direct access to NFVI and MANO.

A.1.4 Network Equipment Provider (NEP)

The NEP role is supported by the 5G-VINNI Norway facility. Multiple NEPs are considered in the Norway facility to supply the equipment and VNFs to networks, including Ericsson for 5G-RAN, 5G-EPC and later 5G Core (SA); Huawei for 5G-RAN; Palo Alto Networks for Next Generation Firewall; and Keysight for testing and validation tools. In addition, Cisco provides Distributed Data Fabric service as part of the slice. Descriptors of 3rd party VNFs can be on-boarded into the NFVO catalogue. There it can be packaged and published as NSDs (Network Service Descriptors) for the FlowOne's catalogue, which can then incorporate that into a Slice template, ready for ordering as a new slice type or a sub-type of an existing slice type. This is a design time process and needs to be carried out, manually. For Release 0 specific scope, please refer to *§5 Support and LCM of Network Slice Services*.

A.1.5 Virtualization Infrastructure Service Provider (VISP)

The VISP role is supported by the 5G-VINNI Norway facility. Nokia designs, builds and operates a virtualization infrastructure via MANO (OpenStack) to provides virtualized infrastructure services. The services are directly consumed by Telenor to build the network services and by the Facility to build the network slice services. In addition, the service may also be consumed by CSCs if the VIS can be exposed upon agreement with the CSCs (ref to D3.1 Section 4.2.4. For Release 0 specific scope, please refer to *§5 Support and LCM of Network Slice Services*.

A.1.6 NFVI (Network Functions Virtualization Infrastructure) Supplier

The NFVI supplier role is supported by the 5G-VINNI Norway facility. Nokia supplies the NFVI (Open Compute Project) that is used to deploy VNFs and network services.

A.1.7 Data Centre Service Provider (DCSP)

The DCSP role is supported by the 5G-VINNI Norway facility. Telenor is responsible for designing, building and operating the data center running in the Norway facility, with support of Nokia providing the NFVI and MANO and from PaloAlto providing the the Firewalls.

A.1.8 Hardware Supplier

The Hardware Supplier role is supported by the 5G-VINNI Norway facility. Nokia is the hardware supplier providing the servers and switches to build the data center.

A.2 Technical pre-testing of the orchestration flows

N/A

Annex B UK Facility

B.1 Supported Roles of 5G-VINNI UK facility

B.1.1 Communication Service Customer (CSC)

The CSC role is supported by the 5G-VINNI UK facility. Vertical customers are considered as CSC and consume the communications services. These vertical customers (e.g. ICT-19 funded projects) need access to browse, order, and receive communications services from the UK facility. To support CSCs for requesting and receiving the services, the UK facility develops

- GUI for browsing the service catalog, ordering the service, and specifying the requirements/attributes and adds-on services such as 3rd party VNF support. In the short-term, Nokia FlowOne has a lean UI for service order submission. In Release 1, orders can be placed manually by CSP via the lean UI from which a Catalog item (Customer Facing Service) can be selected and an order can be placed. This process is separate from order capture or C-P-Q (Configure, Price, Quote) sales cycle. In the long-term, it is expected (tentative) for 5G-VINNI to develop a full-fledged CRM/COM application that should be matured to handle the B2B ordering of network slices.
- GUI/APIs to expose capabilities of Level 1, i.e., the customers have access to general slice features and slice service management.

  Order capture or, C-P-Q cycles between CSP and CSC will be manual, possibly by means of meetings, calls, e-mail communications, etc. Order submission to Nokia FlowOne will be manual, essentially, submitting an XML order payload (Nokia will provide the templates) via some SOAP client application, e.g., SOAP UI, etc. Nokia FlowOne has a lean UI to carry out this order submission process. But, this is not in scope for VINNI. Hence, possibilities of having that in will be assessed. Earliest in release 1 scope.

B.1.2 Communication Service Provider (CSP)

The CSP role is supported by the 5G-VINNI UK facility. The role is taken by BT, with support of Nokia. BT and Nokia jointly design, build, and operate the network slices services and provide them to CSCs. The CSP maintains the customer portal and handles the provisioning issues through the daily operations.

B.1.3 Network Operator (NOP)

The NOP role is supported by the 5G-VINNI UK facility. The role is taken by BT that is responsible for providing the network services required to support the communications services. BT, as a NOP, will be in charge of the design, deployment and operation of network services at the E2E infrastructure. BT has direct access to NFVI and MANO.

B.1.4 Network Equipment Provider (NEP)

The NEP role is supported by the 5G-VINNI UK facility. Multiple NEPs are considered in the UK facility to supply the equipment and VNFs to networks, including Samsung for 5G-RAN (both NSA 5GNR & 5GTF), 4G-EPC and later 5G Core (SA); Palo Alto Networks for Next Generation Firewall; and Keysight for testing and validation tools. Descriptors of 3rd party VNFs can be on-boarded into the NFVO catalog. There it can be packaged and published as components of NSDs (Network Service Descriptors) for the FlowOne catalog, which can then incorporate that into a Slice template, ready for ordering as a new slice type or a sub-type of an existing slice type. This is a design time process and needs to be carried out, manually. For Release 0 specific scope, please refer to *§5 Support and LCM of Network Slice Services*.

B.1.5 Virtualization Infrastructure Service Provider (VISP)

The VISP role is supported by the 5G-VINNI UK facility. Samsung designs, builds and operates a virtualization infrastructure via MANO (OpenStack) to provide virtualized infrastructure services. The services are directly consumed by BT to build the network services and by the Facility to build the communication services. In addition, the service may also be consumed by CSCs if the VIS is allowed to be exposed upon agreement with the CSCs (ref to D3.1 Section 4.2.4. For Release 0 specific scope, please refer to *§5 Support and LCM of Network Slice Services*.

B.1.6 NFVI (Network Functions Virtualization Infrastructure) Supplier

The NFVI supplier role is supported by the 5G-VINNI UK facility. Samsung supplies the NFVI (Open Compute Project) that is used to deploy VNFs and network services.

B.1.7 Data Centre Service Provider (DCSP)

The DCSP role is supported by the 5G-VINNI UK facility. BT is responsible for designing, building and operating the data center running in the UK facility, with support of Samsung providing the NFVI and MANO and from PaloAlto providing the Firewalls.

B.1.8 Hardware Supplier

The Hardware Supplier role is supported by the 5G-VINNI UK facility. Samsung is the hardware supplier providing the servers and switches to build the data center.

B.2 Technical pre-testing of the orchestration flows

N/A

Annex C Greece Facility

C.1 Supported Roles of 5G-VINNI Patras/Greece facility

This section describes the roles supported by the facility as defined in D3.1 for the the 3GPP roles in 5G-VINNI

C.1.1 Communication Service Customer (CSC)

The CSC roles for our facility are the users that want to gain access to our facility and deploy services. (e.g. ICT-19 funded projects). To facilitate the CSC for requesting a network slice service and gain access to the facility's Service Catalogue in a self-service manner, we adopt the technology developed by the 5GinFIRE project (https://5ginfire.eu/), the 5GinFIRE portal. Under special agreements and for specific experiments the CSC may get access to the orchestrators as well as to the NFVI (i.e. deploy/manage Virtual machines, host new hardware, etc.)

C.1.2 Communication Service Provider (CSP)

The CSP role in the Patras facility site is the 5G-VINNI project supported by the Network Architectures and Management group at the Department of Electrical and Computer Engineering, University of Patras. The CSP will maintain the portal access and any provisioning issues.

C.1.3 Network Operator (NOP)

The NOP in the Patras facility site is the Network Architectures and Management group, at the Department of Electrical and Computer Engineering, University of Patras. The NOP will have direct access to Patras NFVI and orchestrator (OSM) via VPN accounts.

C.1.4 Network Equipment Provider (NEP)

For the Patras facility site, as NEP are various providers (partners in 5G-VINNI: Limemicro and Intracom) as well as external providers. Third party VNFs can be supported either by onboarding via the facility portal or some PNFs could be hosted by the NFVI.

C.1.5 Virtualization Infrastructure Service Provider (VISP)

The VISP in the Patras facility is the Network Architectures and Management group , at the Department of Electrical and Computer Engineering, University of Patras. The VISP will have direct access to Patras NFVI

C.1.6 NFVI (Network Functions Virtualization Infrastructure) Supplier

The NFVI suppliers in the Patras facility are selected through processes defined by the University of Patras.

C.1.7 Data Centre Service Provider (DCSP)

The DCSP in the Patras facility is the Network Architectures and Management group, at the Department of Electrical and Computer Engineering, University of Patras.

C.1.8 Hardware Supplier

The Hardware suppliers in the Patras facility are selected through processes defined by the University of Patras.

C.2 Technical pre-testing of the orchestration flows

C.2.1 Pre-testing of lifecycle management of orchestration flows

The Patras facility performed some pre-testing of orchestration flows, prior to the tests scheduled by WP4. We mainly used as source of tests VNFs, NSDs and NetSlice templates from the 5th OSM Hackfest[16]

C.2.2 Definition of pre-testing aspects and conditions

Only a single tenant is used during the tests.

Parallel orchestrations can be performed

C.2.3 Definition of VNFs/NSDs/NSTs used for pre-testing

The following VNFs are used for pre-testing:

Table 6-1 VNFs used for pre-testing

VNF	Description
cirros_sriov_vnf	Generic VNF for testing EPA, SR-IOV capabilities using the cirros image
ubuntu_sriov_vnfd	Generic VNF for testing EPA, SR-IOV capabilities using the Ubuntu image
hackfest_charm_vnfd	VNF for testing charm for Day1, Day2 configurations
hackfest_cloudinit_vnfd	VNF for testing charm for Day 0 configuration
hackfest_multivdu_vnfd	VNF for testing multiple VDUs
hackfest_sfc_vnfd	VNF for testing Service Function Chaining
slice_hackfest_vnfd	VNF for testing NetSlice capability in OSM
flask_app_vnfd	VNF for deploying a generic service

The following NSDs are used for pre-testing:

Table 6-2 NSDs used for pre-testing

NSD	Description
cirros_sriov_ns	A NS descriptor for testing EPA, SRIOV capabilities with the cirros image
ubuntu_sriov_ns	A NS descriptor for testing EPA, SRIOV capabilities with the ubuntu image
hackfest_charm_nsd	A NS descriptor for testing Day1/2 configurations via the VCA

[16] https://osm.etsi.org/wikipub/index.php/5th_OSM_Hackfest

hackfest_cloudinit_nsd	A NS descriptor for testing Day 0 configurations
hackfest_multivdu_nsd	A NS descriptor for testing a Multi-VDU deployment of a VNF
hackfest_sfc_nsd	A NS descriptor for testing Service Function Chaining
slice_hackfest_ns	A NetServiceDescriptor with 2 vnfs and 2 vld (mgmt and data networks) for testing a Network Slice Template instantiation.

The following NSTs are used for pre-testing:

Table 6-3 NST used for pre-testing

NST	**Description**
slice_hackfest_nst	A Network Slice Template used to test Net Slice instantiations

Annex D Spain Facility

D.1 Supported Roles of 5G-VINNI 5TONIC Spain facility

This section describes the roles supported by the facility as defined in D3.1 for the the 3GPP roles in 5G-VINNI

D.1.1 Communication Service Customer (CSC)

The CSC roles for our facility are the users that want to gain access to our facility and deploy services. (e.g. ICT-19 funded projects). To facilitate the CSC for requesting a communication service and gain access to the facility's Service Catalogue in a self-service manner, we adopt the technology developed by the 5GinFIRE project (https://5ginfire.eu/), the 5GinFIRE portal. Under special agreements and for specific experiments the CSC may get access to the orchestrators as well as to the NFVI (i.e. deploy/manage Virtual machines, host new hardware, etc.)

D.1.2 Communication Service Provider (CSP)

The CSP role in the Spain facility site is the 5G-VINNI project supported by 5TONIC, an open research and innovation 5G laboratory founded by Telefónica and the Institute IMDEA Networks. UC3M, in particular, the NETCOM (Network and Communication Services) research group at the Department of Telematics Engineering, also is member of 5TONIC.

D.1.3 Network Operator (NOP)

TID (supported by UC3M) takes the NOP role for the 5TONIC Spain facility site. The NOP will access to 5TONIC NFVI and orchestrator (OSM) under policy defined by 5TONIC members. The NOP will access to 5TONIC NFVI and orchestrator (OSM) under policy defined by 5TONIC members.

D.1.4 Network Equipment Provider (NEP)

For the 5TONIC Spain facility site, as NEP are Ericsson Spain (5TONIC member) for 5G-RAN and 5GCore, and Keysight (external provider) for testing and validation tools. Additional companies might also take the role of NEP as long as they become 5TONIC member, and under consent of 5G-VINNI consortium. Spain facility site will allow 3rd party VNF on-boarding via the facility portal.

D.1.5 Virtualization Infrastructure Service Provider (VISP)

In the Spain facility site, the VISP is the set of Openstack releases that are deployed, operated and maintained across the 5TONIC's NFVI.

D.1.6 NFVI (Network Functions Virtualization Infrastructure) Supplier

The selection of NFVI suppliers in the Spain facility site are subjected to policy defined by 5TONIC members.

D.1.7 Data Centre Service Provider (DCSP)

The DSCP is the Spain facility site is 5TONIC. 5TONIC has staff that is responsible for designing, building and operating the servers running in the lab.

D.1.8 Hardware Supplier

The selection of Hardware suppliers in the Spain facility site are subjected to policy defined by 5TONIC members.

D.2 Monitoring entry point and Data assurance extensions

For Spain facility site the monitoring can be performed by the CSP/NOP and in some cases by CSC, for those cases where selected exposure level is 2 or higher. As specified in Section 6.2.2, the OSM monitoring system to be used in 5TONIC facility may be extended to support model-driven telemetry. This extension will allow retrieving VNF metrics not only through Juju (OSM V2NC), but also through data models. The use of data models for metering is called as model-driven telemetry. Figure 6.4 shows how OSM MON working environment is extended with the application of model-based telemetry

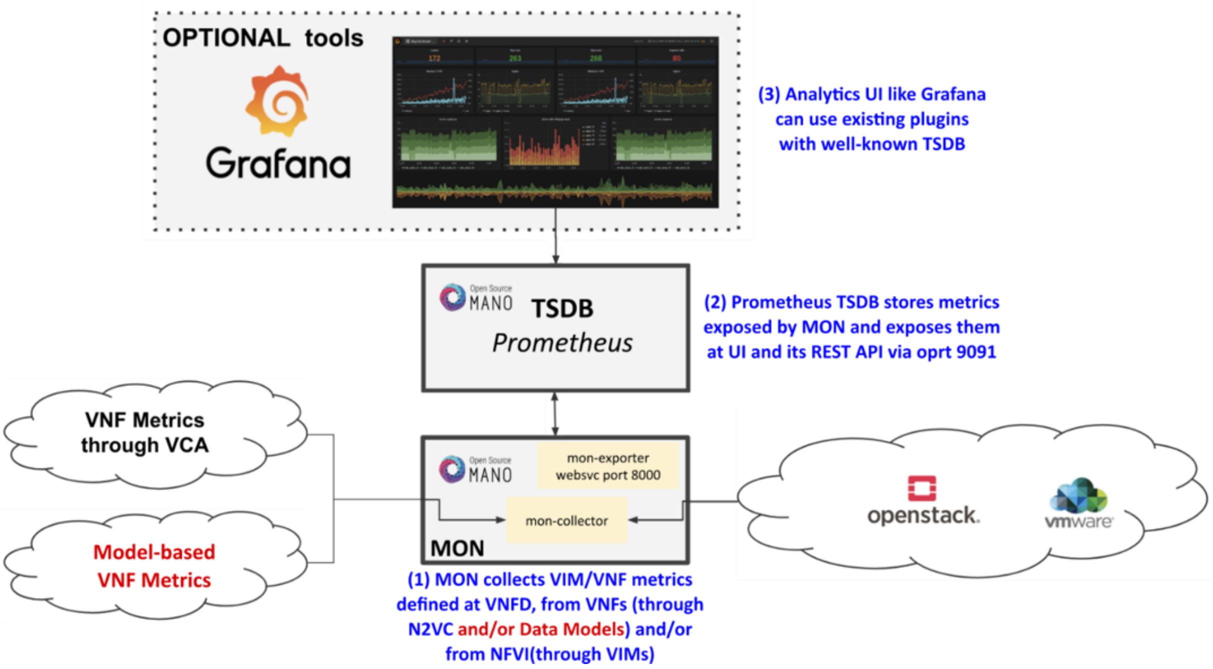

OPTIONAL tools
Grafana
(3) Analytics UI like Grafana can use existing plugins with well-known TSDB
Open Source MANO
TSDB
Prometheus
(2) Prometheus TSDB stores metrics exposed by MON and exposes them at UI and its REST API via oprt 9091
VNF Metrics through VCA
Model-based VNF Metrics
Open Source MANO
mon-exporter websvc port 8000
mon-collector
MON
openstack.
vmware
(1) MON collects VIM/VNF metrics defined at VNFD, from VNFs (through N2VC and/or Data Models) and/or from NFVI(through VIMs)

Figure 6-5 Model-driven telemetry and its applicability to the working environment of OSM monitoring system

Additionally, the monitoring system would be extended with the implementation of a data model-driven telemetry. Specifically, this extension will add flexibility to the second option supported by OSM (R5). In a similar way, the VNFD will add some features to its information model in which the metric is defined.

Model-driven telemetry is a new concept of monitoring the network that follows the push mechanisms instead of a pull mechanism. The push mechanism differs from the pull-based in which the information is sent by the devices each time an update happens, and it is not a manager who has to request each time it is need. It follows a publisher-subscriber architecture, in which the managers can subscribe to a specific data on the devices. This allows to get only the information need and filter out the rest, decreasing the traffic through the network. Furthermore, model-driven telemetry also follows a structure on the data, which means telemetry data must be model-based. In this way, the monitoring tools can ingest the data easily.

Regarding this schema, the network industry has converged on YANG as a data modelling language for networking data. With the success of YANG as a modelling language, many YANG models are arising to ease automation. These YANG modules come from three different sources:

- Standards development Organizations (SDOs) – IETF
- Consortium, fora and open source projects – OpenConfig, Metro Ethernet Forum (MEF), OpenDayLight, etc.
- Native/proprietary YANG models

The YANG modules produced by the SDOs have great relevance as they are reviewed by many people, but it takes a long time to finalize the specifications. Moreover, the data defined should contain a kind of common denominator, as opposed to the full coverage of the different experimental or proprietary features proposed by the different networking vendors.

In contrast, YANG models produced by consortiums, forums, and open source projects are most of the time targeted toward specific use cases, thus proposing complete solutions. Among all the open source projects, the OpenConfig solution is the one that takes more relevance.

The proprietary YANG models are defined by the networking vendors based on their proprietary implementations. This implies that automation across vendors proved difficult.

Figure 6-6 depicts the general architecture of a model-driven telemetry in which YANG models are placed in the lowest layer. This allows to create a model-driven telemetry service with the combination of the transport protocols (NETCONF, RESTCONF, gRPC) and encoding formats (XML, JSON, GPB) that fits better to a specific environment.

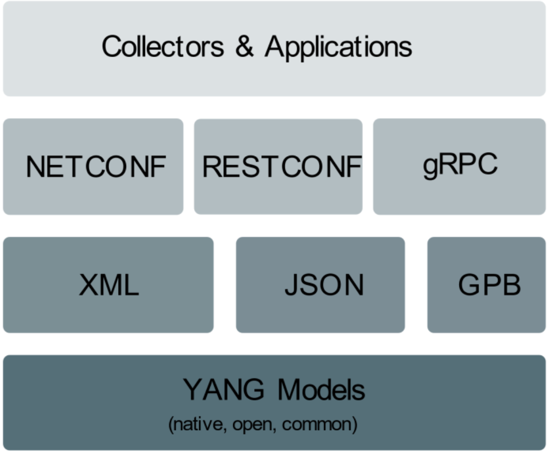


Figure 6-6 Model-driven telemetry solution

In Figure 6-7 it can be seen the model-driven telemetry stack that would be followed in the monitoring system in the Spain facility. As it shows, the monitoring system will integrate the gRPC protocol as well as the protocol buffers codification with an open collector/application. The collector, included in this system is available in the OpenConfig GitHub account (https://github.com/openconfig/gnmi) is called gNMI (gRPC Network Management Interface).

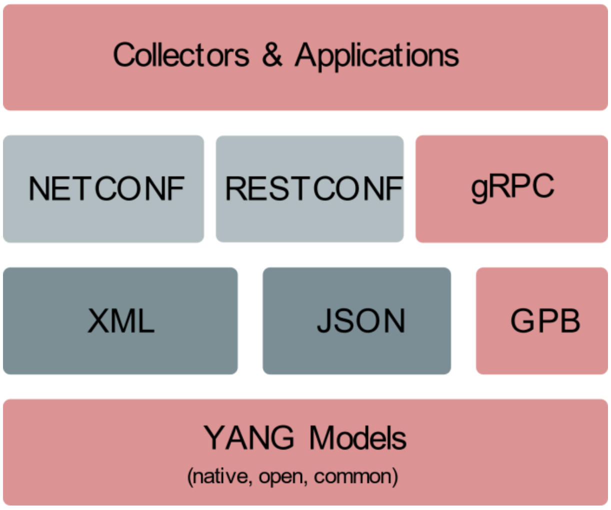


Figure 6-7 Model-driven telemetry stack in Spain facility site

Figure 6-8 provides a detailed view on the telemetry service that may be introduced in OSM to allow retrieving model-based VNF metrics. To facilitate a seamless integration of this extension to the OSM monitoring system, the telemetry service will be deployed in different Docker containers, each carrying out a different task. As it can be seen, it integrates the gNMI client (publicly available in GitHub) which will be in charge of subscribing to a specific data on the VNF and will be listening for the VNF's updates

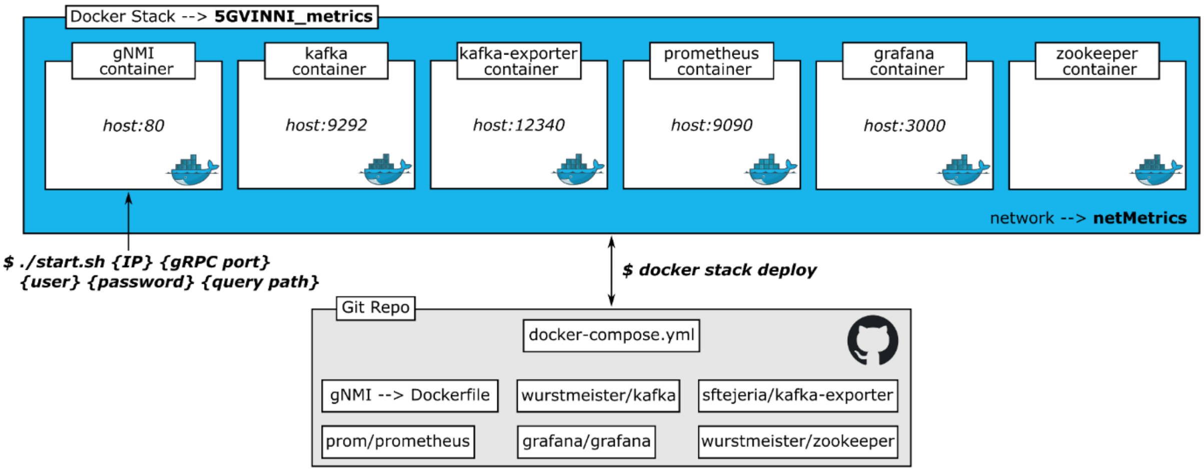


Figure 6-8 Telemetry Service Deployment

Figure 6-9 shows how collected telemetry data flows with the proposed service. As it can be seen, once collected data arrives to the telemetry service, they will be translated into the correct format before saving corresponding information on the Prometheus database.

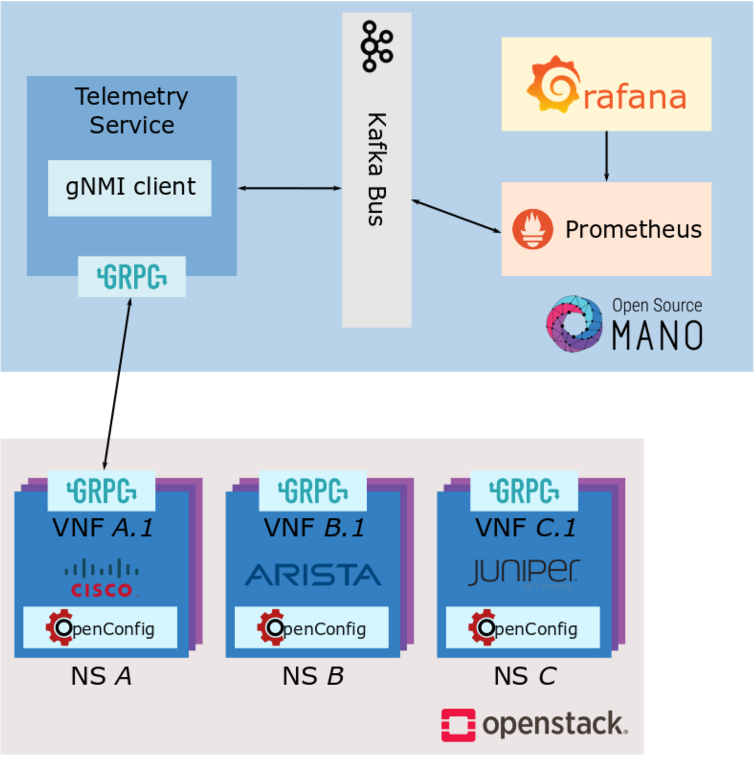


Figure 6-9 Telemetry data flow

D.3 Technical pre-testing of the orchestration flows

D.3.1 Pre-testing of lifecycle management of orchestration flows

The Spain facility performed some pre-testing of orchestration flows, prior to the tests scheduled by WP4. We mainly used as source of tests VNFs, NSDs and NetSlice templates from the 5th OSM Hackfest[17]

D.3.2 Definition of pre-testing aspects and conditions

Only a single tenant is used during the tests.

Parallel orchestrations can be performed

D.3.3 Definition of VNFs/NSDs/NSTs used for pre-testing

The following VNFs are used for pre-testing:

Table 6-4 VNFs used for pre-testing

VNF	Description
cirros_sriov_vnf	Generic VNF for testing EPA, SR-IOV capabilities using the cirros image
ubuntu_sriov_vnfd	Generic VNF for testing EPA, SR-IOV capabilities using the Ubuntu image
hackfest_charm_vnfd	VNF for testing charm for Day1, Day2 configurations
hackfest_cloudinit_vnfd	VNF for testing charm for Day 0 configuration
hackfest_multivdu_vnfd	VNF for testing multiple VDUs
hackfest_sfc_vnfd	VNF for testing Service Function Chaining
slice_hackfest_vnfd	VNF for testing NetSlice capability in OSM
flask_app_vnfd	VNF for deploying a generic service

The following NSDs are used for pre-testing:

Table 6-5 NSDs used for pre-testing

NSD	Description
cirros_sriov_ns	A NS descriptor for testing EPA, SRIOV capabilities with the cirros image
ubuntu_sriov_ns	A NS descriptor for testing EPA, SRIOV capabilities with the ubuntu image
hackfest_charm_nsd	A NS descriptor for testing Day1/2 configurations via the VCA

[17] https://osm.etsi.org/wikipub/index.php/5th_OSM_Hackfest

hackfest_cloudinit_nsd	A NS descriptor for testing Day 0 configurations
hackfest_multivdu_nsd	A NS descriptor for testing a Multi-VDU deployment of a VNF
hackfest_sfc_nsd	A NS descriptor for testing Service Function Chaining
slice_hackfest_ns	A NetServiceDescriptor with 2 vnfs and 2 vld (mgmt and data networks) for testing a Network Slice Template instantiation.

The following NSTs are used for pre-testing:

Table 6-6 NST used for pre-testing

NST	Description
slice_hackfest_nst	A Network Slice Template used to test Net Slice instantiations

Annex E Portugal Facility

E.1 Supported Roles of 5G-VINNI Aveiro facility

E.1.1 Communication Service Customer (CSC)

The CSC role in the Aveiro facility site will be played primarily by EFACEC, in the framework of the ICT-19 5GROWTH project. In addition, the facility will be available for 5G-related experimentation activities outside and beyond ICT-19, either internally (in which case the CSC will be played by Altice Labs) or in collaboration with external partners (not defined at this stage).

E.1.2 Communication Service Provider (CSP)

The CSP role in the Aveiro facility site will be played by the 5G-VINNI project, supported by Altice Labs.

E.1.3 Network Operator (NOP)

The NOP role in the Aveiro facility site will be played by Altice Labs, which will have direct and exclusive access to the infrastructure, as well as to management & orchestration platforms.

E.1.4 Network Equipment Provider (NEP)

For the Aveiro facility site, the most important NEPs will be Fraunhofer Fokus (will provide the mobile core components - Open5GCore); OpenAirInterface Software Alliance (will provide the radio access component - OpenAirInterface) and Altice Labs (will provide the optical backhaul/midhaul/fronthaul component).

E.1.5 Virtualization Infrastructure Service Provider (VISP)

The VISP in the Aveiro facility site will be played by Altice Labs.

E.1.6 NFVI (Network Functions Virtualization Infrastructure) Supplier

The NFVI suppliers in the Aveiro facility site will be selected by Altice Labs through internally defined processes.

E.1.7 Data Centre Service Provider (DCSP)

The DCSP role in the Aveiro facility site will be played by Altice Labs.

E.1.8 Hardware Supplier

The hardware suppliers in the Aveiro facility site will be selected by Altice Labs through internally defined processes.

E.2 Technical pre-testing of the orchestration flows

N/A

Annex F Munich Facility

F.1 Supported Roles of 5G-VINNI Munich facility

F.1.1 Communication Service Customer (CSC)

The CSC role for in our Munich facility is the collection of customers and ESB member that will use the platform. The customers and ESB member will receive access to the platform based some specific scenario. Partially the interaction from the CSC could be based on pre-defined configurations and settings.

F.1.2 Communication Service Provider (CSP)

The role of CSP is played by Huawei as we are the entity running and operating and providing the services. We have a minimal implementation of the role as the roles of CSP and NOP are collocated.

F.1.3 Network Operator (NOP)

The NOP role is also taken by Huawei. The operator role is responsible for making sure the network is operational and providing the services. It offers some specific methods for reporting issues and making sure, using both automated and manual processes, that the network is functioning and provide the correct KPIs and service level agreements.

F.1.4 Network Equipment Provider (NEP)

Huawei is the NEP as most equipment (RAN, Core, Server Rack, etc..) are provided by Huawei.

F.1.5 Virtualization Infrastructure Service Provider (VISP)

For the actual virtualisation we rely on the Docker container technology. They are light weight and provide the best balance between configurability and flexibility. For the actual infrastructure we have deployed our own infrastructure for virtualisation as highlighted in D2.1

F.1.6 NFVI (Network Functions Virtualization Infrastructure) Supplier

Docker[18] container and Mininet[19] are utilized together to realize our NFVI multi-host platform on top of our hardware installation. Docker containers provide a virtualized environment, where different NFs can share the same hardware resource from the server machine isolated from each other.

Mininet provides software-defined networking capability to customize the interconnectivity of the virtualized NFs running in Docker containers through Open Virtual Switch (OVS)[20]. Mininet also provides capability to connect to one or multiple software-defined network controllers. Network control application logic can operate the network via the interfaces provided by the controller.

At HWDU experimental facility site, we use a multiple-host capable, mobility-supporting extended Mininet testbed. In this testbed, nodes are Docker containers with an OVS instance inside (as opposed to OVS, as in standard Mininet). Besides, such nodes can be deployed across different physical hosts; our system guarantees that regardless of the virtual node to physical machine assignment, the desired virtual topology constraints are always respected. Finally, nodes can be reattached within the network, therefore emulating mobility events.

[18] https://www.docker.com/

[19] http://mininet.org/

[20] https://www.openvswitch.org/

F.1.7 Data Centre Service Provider (DCSP)

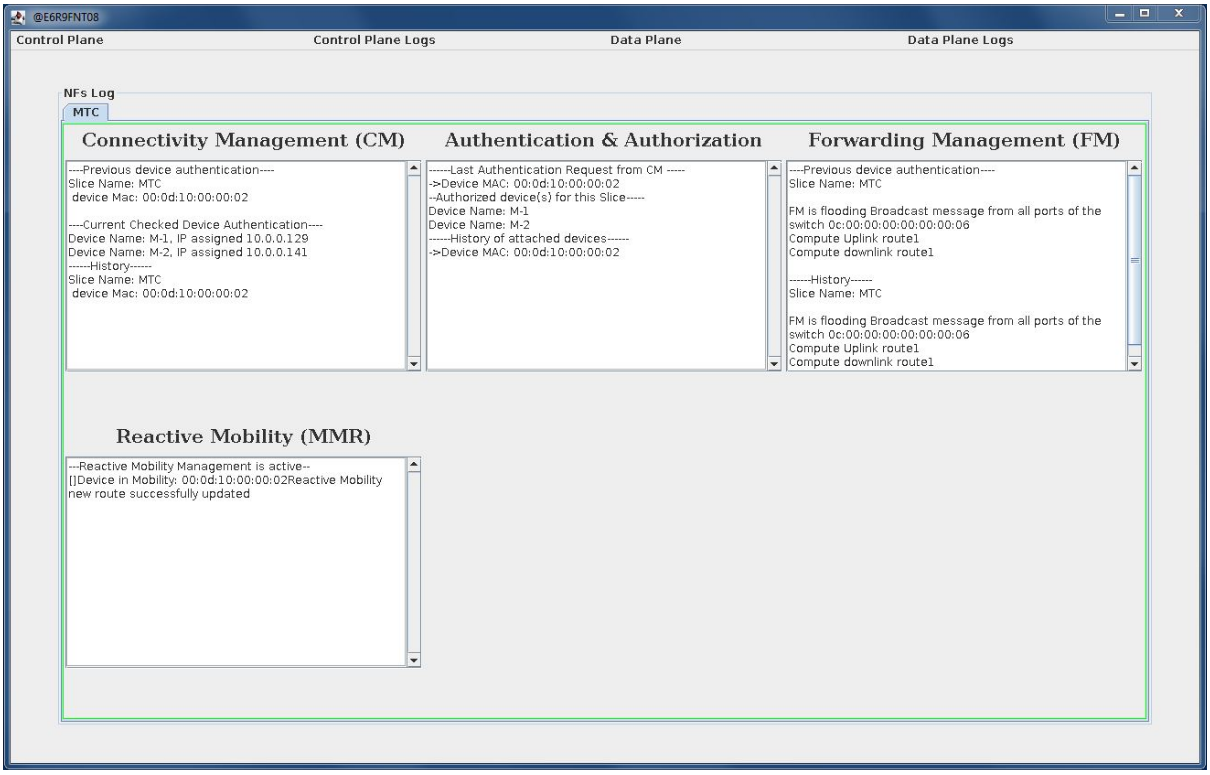
We do not have a specific DSCP, we rely on in-house small-scale data centre.

F.1.8 Hardware Supplier

The hardware is provided in house from Huawei Technologies.

F.2 Technical pre-testing of the orchestration flows

Pre-testing is done with the GUI to assess the created slice along with the individual components. In the below figure we show how the slice created before can be pre-tested and verified.



Connectivity Management (CM)

```
----Previous device authentication----
Slice Name: MTC
 device Mac: 00:0d:10:00:00:02

----Current Checked Device Authentication----
Device Name: M-1, IP assigned 10.0.0.129
Device Name: M-2, IP assigned 10.0.0.141
------History------
Slice Name: MTC
 device Mac: 00:0d:10:00:00:02
```

Authentication & Authorization

```
------Last Authentication Request from CM -----
->Device MAC: 00:0d:10:00:00:02
--Authorized device(s) for this Slice-----
Device Name: M-1
Device Name: M-2
------History of attached devices------
->Device MAC: 00:0d:10:00:00:02
```

Forwarding Management (FM)

```
----Previous device authentication----
Slice Name: MTC

FM is flooding Broadcast message from all ports of the
switch 0c:00:00:00:00:00:00:06
Compute Uplink route1
Compute downlink route1

------History------
Slice Name: MTC

FM is flooding Broadcast message from all ports of the
switch 0c:00:00:00:00:00:00:06
Compute Uplink route1
Compute downlink route1
```

Reactive Mobility (MMR)

```
---Reactive Mobility Management is active--
[]Device in Mobility: 00:0d:10:00:00:02Reactive Mobility
new route successfully updated
```

Figure 6-10 overview of the pre-testing and verification of the created slice/service

Generally, first the basic of each slice/service will be checked. This include checking the assigned IP address and basic connectivity to the different entities. It also verifies the connectivity between the different entities.

Annex G Berlin Facility

G.1 Supported Roles of 5G-VINNI Munich facility

G.1.1 Communication Service Customer (CSC)

As Berlin and Luxemburg are experimentation facilities, the communication with the customers is executed on a discussion base through the presentation of the existing slice catalogue. Based on the catalogue, a new slice model will be customized for the specific use case. It is considered opportune to have such requirements discussion with each of the verticals using the facility to better understand and adapt the slice models, thus providing experimental innovative slice models instead of a standard pre-defined set and complete automation on pre-defined models.

G.1.2 Communication Service Provider (CSP)

The CSP role in the Berlin facility site is the 5G-VINNI project supported by 5G-Playground Berlin (www.5g-playground.org), a research and innovation testbed deployment founded and supported by Fraunhofer FOKUS. The CSP is extended with the Luxemburg facility aiming at extending with satellite specific technologies the 5G environment.

G.1.3 Network Operator (NOP)

The role of the NOP is taken by the SES in Luxemburg and Fraunhofer FOKUS in Berlin. The Berlin and Luxemburg facilities are seen as small private network operators aiming at fostering advancements of technologies and not as commercial deployments.

G.1.4 Network Equipment Provider (NEP)

For the Luxemburg and Berlin facilities, the 5G-RAN equipment is provided by external third equipment providers, the satellite equipment and capacity is provided by SES while the 5GCore is provided by Fraunhofer FOKUS Open5GCore, customized for the specific use case needs. Other 3rd party VNFs, especially the ones customized for the use cases, will be integrated on demand into the platform in order to demonstrate the feasibility of the specific verticals.

G.1.5 Virtualization Infrastructure Service Provider (VISP)

The Luxemburg and Berlin facilities are based on OpenStack which acts as virtual infrastructure manager. The OpenStack is operated by SES and Fraunhofer FOKUS.

G.1.6 NFVI (Network Functions Virtualization Infrastructure) Supplier

The NFVI used is based on existing standard common off-the-shelf servers and switches as available and continuously upgraded at the specific locations. The usage of common hardware makes the research results in the facility easy to port to different other locations. For the remote nodes, low resources hardware is considered including bare-metal parallel slice deployments.

G.1.7 Data Centre Service Provider (DCSP)

Both SES and Fraunhofer are running their own data centers to which the experimental 5G-VINNI facility will be appended to.

G.1.8 Hardware Supplier

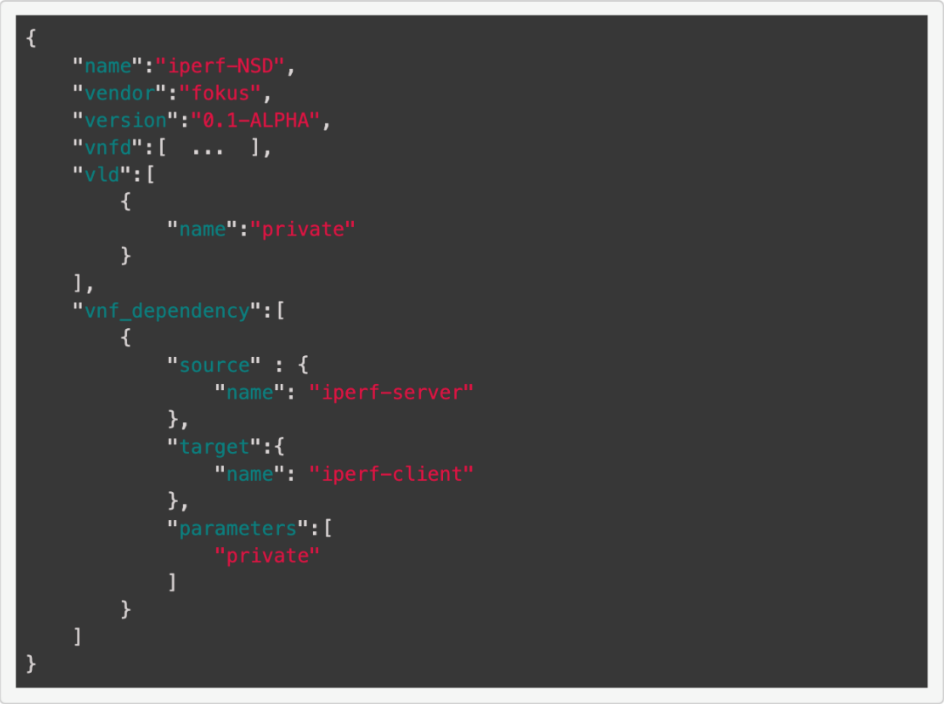
The hardware for the Berlin and Luxemburg facilities is done based on respecting the specific acquisition rules.

G.2 Technical pre-testing of the orchestration flows

N/A

G.3 Supported Network Slice Service Modelling approaches

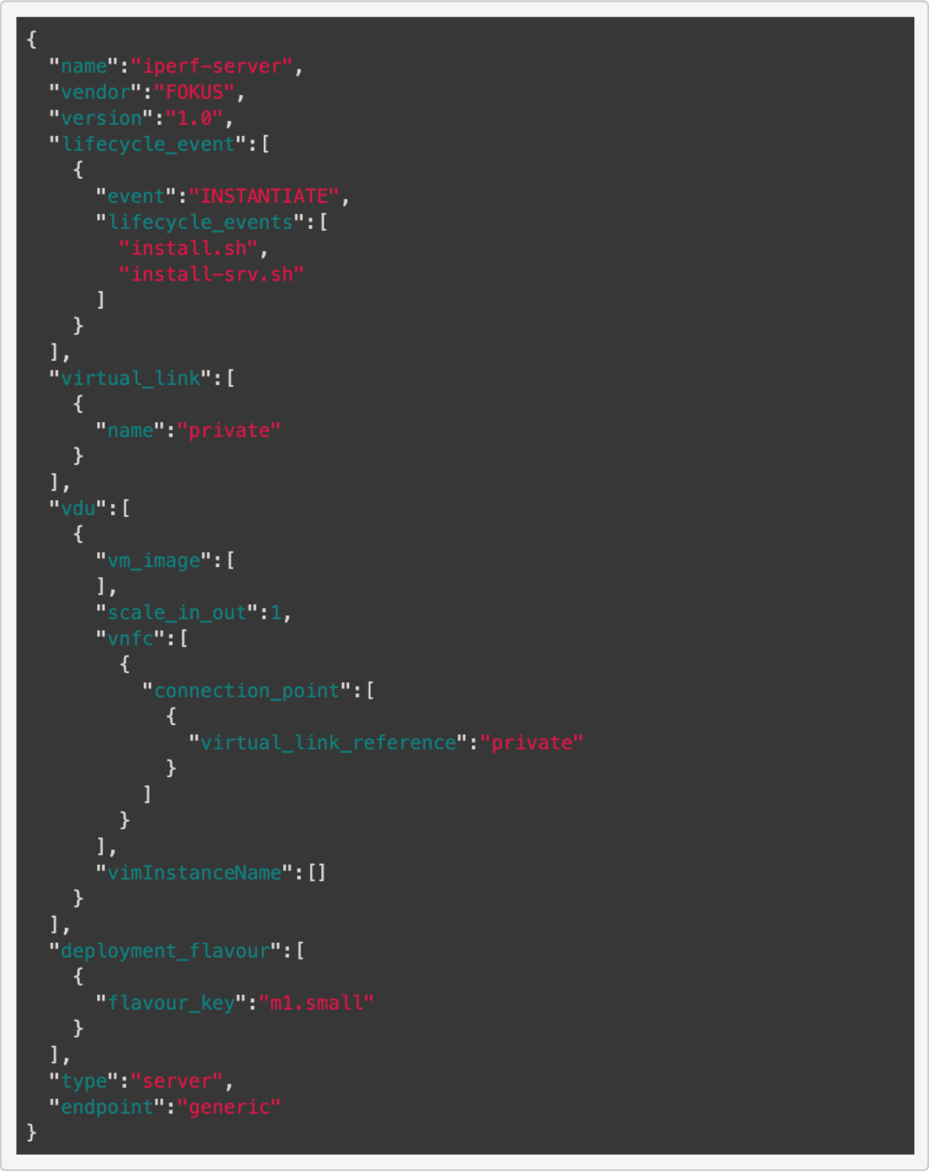
In this section will only go further into the details of option one: JSON file representation of the information model specified by the ETSI MANO specification



```
{
    "name":"iperf-NSD",
    "vendor":"fokus",
    "version":"0.1-ALPHA",
    "vnfd":[  ...  ],
    "vld":[
        {
            "name":"private"
        }
    ],
    "vnf_dependency":[
        {
            "source" : {
                "name": "iperf-server"
            },
            "target":{
                "name": "iperf-client"
            },
            "parameters":[
                "private"
            ]
        }
    ]
}
```

Figure 6-11 Structure of an OpenBaton NSD

Figure 6-11 shows the structure of an NSD. It has a name, vendor and version field. The vnfd section includes individual VNFDs or references to VNFDs that have been uploaded to the NFVO in the form of VNF packages. The vld field contains the virtual link descriptors to which the Network Service deployed from the NSD will be connected. The names of the virtual link descriptors correspond to the networks which are created by OpenBaton on the NFVI. Network Slices can be formed by using separate networks and different network services can be connected to the same network. The vnf_dependency field defines the dependencies which exist between the VNFs contained in the NSD. The source and the target of the dependencies can be specified as well as the name of the parameter. The dependencies are resolved by the NFVO during deployment time so that parameters can be set dynamically.



```
{
  "name":"iperf-server",
  "vendor":"FOKUS",
  "version":"1.0",
  "lifecycle_event":[
    {
      "event":"INSTANTIATE",
      "lifecycle_events":[
        "install.sh",
        "install-srv.sh"
      ]
    }
  ],
  "virtual_link":[
    {
      "name":"private"
    }
  ],
  "vdu":[
    {
      "vm_image":[
      ],
      "scale_in_out":1,
      "vnfc":[
        {
          "connection_point":[
            {
              "virtual_link_reference":"private"
            }
          ]
        }
      ],
      "vimInstanceName":[]
    }
  ],
  "deployment_flavour":[
    {
      "flavour_key":"m1.small"
    }
  ],
  "type":"server",
  "endpoint":"generic"
}
```

Figure 6-12 Structure of an OpenBaton VNFD

The structure of a VNFD [OpenBaton-VNFD] as it is used in OpenBaton can be seen in Figure 6-12 Structure of an OpenBaton VNFD. The VNFD represents a single VNF of which a Network Service is composed of. The lifecycle operations of VNFs are defined on this descriptor level. There are several types of lifecycles available, each responsible for different situations.

- INSTANTIATE: The instantiate lifecycle is executed first after the virtual machine or container running the VNF has been launched. It is intended to run scripts and commands which install the software required by the VNF.
- CONFIGURE: This lifecycle can be used to provide VNFs with the information needed to configure it correctly. Dependencies between the VNFs are resolved in this stage. This means that a VNF has access to the dependency parameters which have been defined in the NSD.
- START: The start lifecycle is used to start the software of the VNF after the configuration stage has finished.
- TERMINATE: Before removing a running VNF it might be necessary to run clean-up operations. For this the terminate lifecycle can be used. It runs shortly before the VNF is removed.

- SCALE_IN: Network Services deployed by OpenBaton can scale if needed. For a certain VNF it might be possible to launch additional instances if needed or to reduce the number. The SCALE_IN lifecycle runs on a VNF instance if a VNF which is the source of a dependency is scaled in (i.e. the number of instances is reduced). This makes sure that the Network Service can adapt to the scaling mechanism and continues to work correctly.

While the VNFD describes the general appearance and functionality of a VNF, it does not represent the instances of the VNF. Multiple instances can be launched from the same VNFD. A VNFD comprises one or more Virtual Deployment Units (VDU), which themselves consist of VNF-components (VNFC). In the deployment of a VNF, each VNFC is mapped to one instantiation of the VNF. The VNFCs specify information about the connectivity of the VNF instances which the VDUs can limit the number of instances and define the type of image from which the VNF instances are created.

Besides the NSD and VNFD, there is a third building block required for describing VNFs and their collaboration in a Network Service. *VNF packages* are used to pack VNFDs and metadata information together so that it can be onboarded to the NFVO.

A VNF Package is a tar-archive containing all the information required for managing the lifecycle of a VNF. First step is to build the archive which then can be onboarded to the NFVO. A typical VNF Package includes

- VNF Descriptor: containing all the information required by the NFVO for deploying the VNF (more information available at the VNF Descriptor page).
- Image: the image to use for starting instances of the VNF
- Metadata file providing additional information to the NFVO for understanding what's the content of the package.
- scripts: containing all the scripts which could be used for lifecycle management.

Annex H OSM

H.1 OSM information model for network slicing in 5G

This section will study with the OSM framework allows the definition of an information model for network slicing in 5G. The key role in this section is the NOP, being this is a role adopted in both Spain and Patras facility sites.

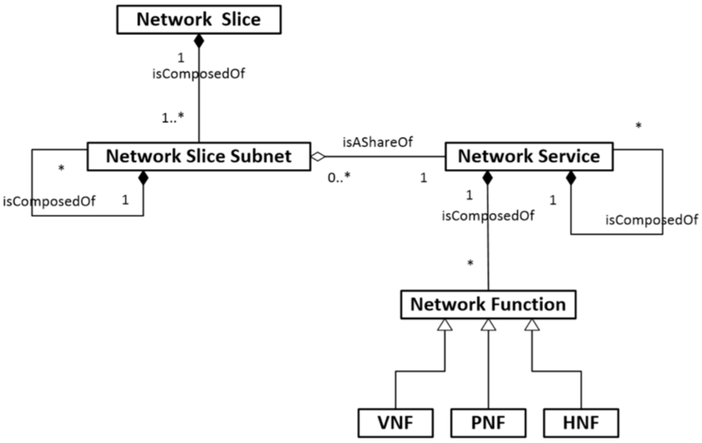


Figure 6-13 OSM framework to align 3GPP and ETSI NFV views on network slicing

The OSM framework shown in Figure 6-13 defines the relation between network slices (3GPP scope, left-side of Figure 6-13) and NFV-NSs (ETSI NFV scope, right-side of Figure 6-13) view in a very specific manner, where the network slice is an end-to-end logical network composed of one or more network slice subnets (e.g. RAN and CN slice subnets), each deployed and operated as a single NFV-NS. According to this vision, an NSI can be composed of one or more NFV-NS instances, being these instances from the same NSD or from different NSDs.

The network slice subnets taking part of the definition of a network slice can be either exclusive to that slice (*dedicated slice subnets*) or shared between several slices (*dedicated slice subnets*). At CN side, an example of a dedicated subnet is an NFV-NS consisting of one or more UPFs and required slice-specific CP NFs (e.g. at least SMF), and an example of a shared CN slice subnet could be another NFV-NS with all the cross-slice CP NFs (e.g. NSSF, UDM, AMF, etc.). At the RAN side, similar examples can also be applied.

The NFV-NS implementing network slice subnet can be defined as a composition of NFs, where at least one of these NFs is deployed as a VNF. Other possible implementations of NFs are Physical Network Functions (PNFs) and Hybrid Network Functions (HNFs).

With the above discussions, the following statement can be derived from OSM reference model: a 3GPP NSSI is a share of a single NFV-NS instance. This statement relies on the following principles:

1. *Composability in the definition of both network slice subnet and NFV-NSs*: both network slice subnets and NFV-NSs are information elements that can be recursively composed. In case of 3GPP, a (coarse-grained) network slice subnet can be flexibly built out of one or more (fine-grained) network slice subnets. For example, the dedicated and shared network slice subnets exemplified earlier can be composed to define a complete CN slice subnet. In case of ETSI NFV, this is enabled with the concepts of composite/nested NFV-NSs [NFV-003], whereby a composite NFV-NS is an NFV-NS that at least includes one nested NFV-NS.
2. *Sharing relationship between NFV-NS and network slice subnet*. The abovementioned sharing relationship is the key new feature which slicing introduces and has an important consequence for orchestration. This consequence is represented in the UML figure above by

the fact that the '*isAShareOf*' relationship is an aggregation (open diamond) and not a composition (solid diamond). This sharing relationship is defined to give support to two possible scenarios:

   a) VNF/PNF/HNFs part of NFV-NS can implement not only 3GPP NFs from a network slice subnet (e.g. 5GC NFs from CN slice subnet, or gNB functions from RAN slice subnet), but also other value-added L4-L7 NFs that the operator may define in the context of security, traffic optimization, etc. In the former case, network slice subnet can be modeled as a nested NFV-NS that is a part of a composite NFV-NS (nested NFV-NS + L4-L7 non-3GPP NFs), where a sharing relationship shall be defined.
   b) The NFV-NS provides shares to more than one network slice subnet. This could happens when the same NFV-NS (e.g. RAN NFV-NS) is used to implement the network slice subnets (e.g. RAN network slice subnets) of different network slices. In such case, when an NSI which is sharing an instance of this NFV-NS is no longer needed and deleted, the constituent NSSIs may be deleted, but the NFV-NS instance itself must not be deleted as it may well be still providing shares to the NSSIs taking part of the rest of NSIs.

3. *Separate LCM of network slice subnets and NFV-NSs*. It is a direct consequence of the sharing relationship. This means that the lifecycle of network slices with their constituent network slice subnets can be managed as a composition hierarchy in same way the lifecycle of NFV-NSs and constituent (V/P/H)NFs can be managed as a composition hierarchy, the two composition hierarchies must not mixed together.

OSM is committed to rely on those principles to natively support network slicing, allowing NOP not only to deploy traditional NFV-NSs, but also NSIs. For this end, the original OSM information model (VNFD, NSD) has been extended in the release FIVE with the inclusion of the Network Slice Template (NST). The NST is a deployment template that assists OSM in the deployment and operation of NSIs.

NSTs are equivalent to NSDs, but applied at a higher abstraction level, i.e. network slice level rather than network service level. In the same way an NSD allows creating one or more instances of a given NFV-NS type (and operating them at run-time), an NST allows creating one or more NSIs of a given network slice type (and operating them at run-time).

The relationship between NST and NSD in the new OSM information model is fully aligned with the conceptual framework shown in Figure 6-13. Proof of this is alignment is shown in Figure 4.4, where three different NSTs (NST#1, NST#2, NST#3) and NSDs (NSD#1, NSD#2, NSD#3) have been defined as a way of example. Some of the relationships that shall be remarked from NST-NSD relationships are the following:

- The concept of network slice subnet is mapped to the concept NFV-NS. This means that each NSSI can be realized as a single NFV-NS instance, and thus deployed from a single NSD.
- Some network slice subnets are *dedicated* (green and blue), while others are *shared* (purple).
- Purple network slice (NST#2) is composed of a single network slice subnet. This means that any NSI deployed from NST#2 is a purple NSSI, realized as an NFV-NS instance from NSD#2.
- Green network slice (NST#1) and blue network slice (NST#3) are composed of two network slice subnets each, where one of them is a purple network slice subnet. This means that both green and purple NSI will contain purple NSSIs.
- A purple NSSI can be shared across a green NSI and a blue NSI, or not. In the latter case, two different NSSIs are deployed, one for each NSI.

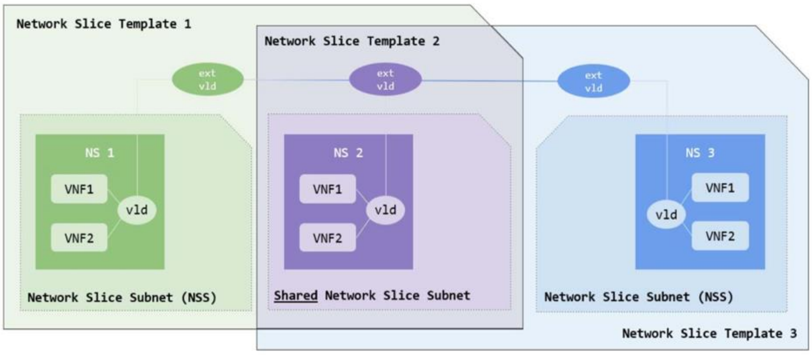


Figure 6-14 NST and NSD relationships

Once the motivation behind the extension of OSM information model with NST has been explained, the next step is to provide an overview of the NST information model on which NOP will rely for network slicing deployment. This model uses YANG as data modelling language, and consists of five differentiated parts that deserve great attention:

- *Part 1 – Network slice ID, name and main parameters*: defines the main characteristics of the network slice. For this end, it specifies an UUID to uniquely identify the NST, provides a brief description of what the network slice is about, and includes a 3GPP-oriented characterisation of the network slice. This characterization is based on the 3GPP concepts of Single-Network Slice Selection Assistance Information (S-NSSAI), i.e. used for 5GC to NSI establishment, and 5G Quality Information (5QI), i.e. QoS parameters [3GPP23.501].
- *Part 2 – Constituent Network slice subnets:* lists all the network slice subnets the network slice is composed of. For each network slice subnet, it is specified the following information:
  - If shared or dedicated
  - Deployment description, i.e. information on the NFV-NS implementing the network slice subnet. This information includes the reference to the corresponding NSD along with required instantiation information (e.g. selected flavor, VIMs where VNFs will be deployed, etc).
- *Part 3 – Network slice connection points:* lists all the connection points the network slice consists of, including the ones exposed by network slice subnets as well as those externally exposed by the network slice (e.g. enabling attachment of 3rd party VNFs).
- *Part 4 – Network slice Virtual Link Descriptors (VLDs):* provides a complete description (e.g. root bandwidth, leaf bandwidth, physical network, network segment) of all the virtual links that take part of the slice, but not on the NFV-NS. This include:
  - Virtual links providing data plane connectivity between network slice subnets (inter-NFV-NS connectivity)
  - Virtual links used for management purposes.
- *Part 5 – Network slice Forwarding Graph Descriptor (FGDs):* describes the topology of the network slice, and optionally includes forwarding rules to describe how traffic shall flow between the network slice subnet defined in this topology. As seen, this field is similar to VNF Forwarding Graphs (VNFFG) in NSD, but applied at network slice level.

A complete view on NST information model is shown in Annex H.2 The navigable version of this model can be found in [OSM].

H.2 OSM NST information model

In this section, we will focus on the above-referred NST parts, studying their main fields and their applicability to a concrete example that is shown in Figure 6-15, and that aims at helping the potential readers of this doc to better understand the possibilities that OSM's NST brings. This example, retrieved from the 5th OSM hackfest held in Barcelona on February 2019 (https://osm.etsi.org/wikipub/index.php/5th_OSM_Hackfest), shows a eMBB network slice consisting of two network slice subnets, each deployed as a separate NFV-NS. The corresponding NST information model is presented in Figure 6-15 NST diagram

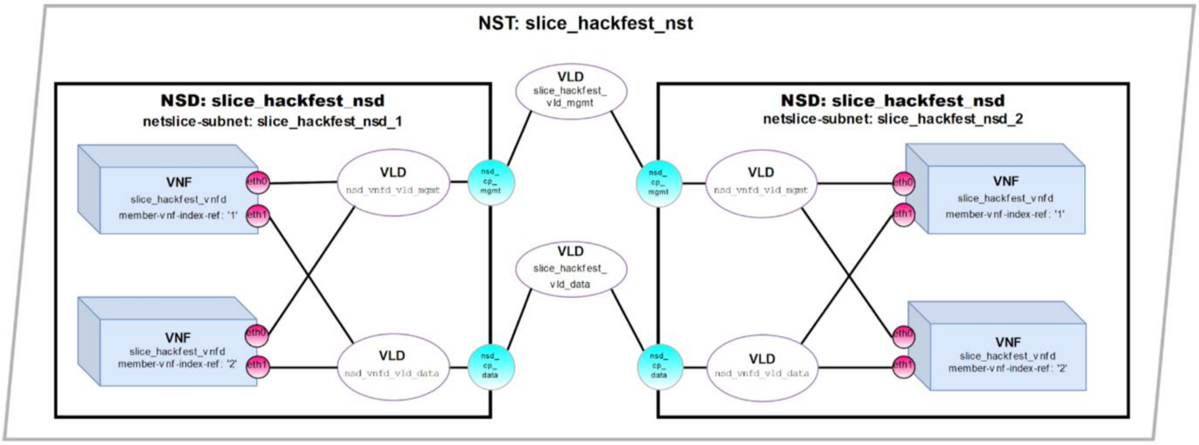
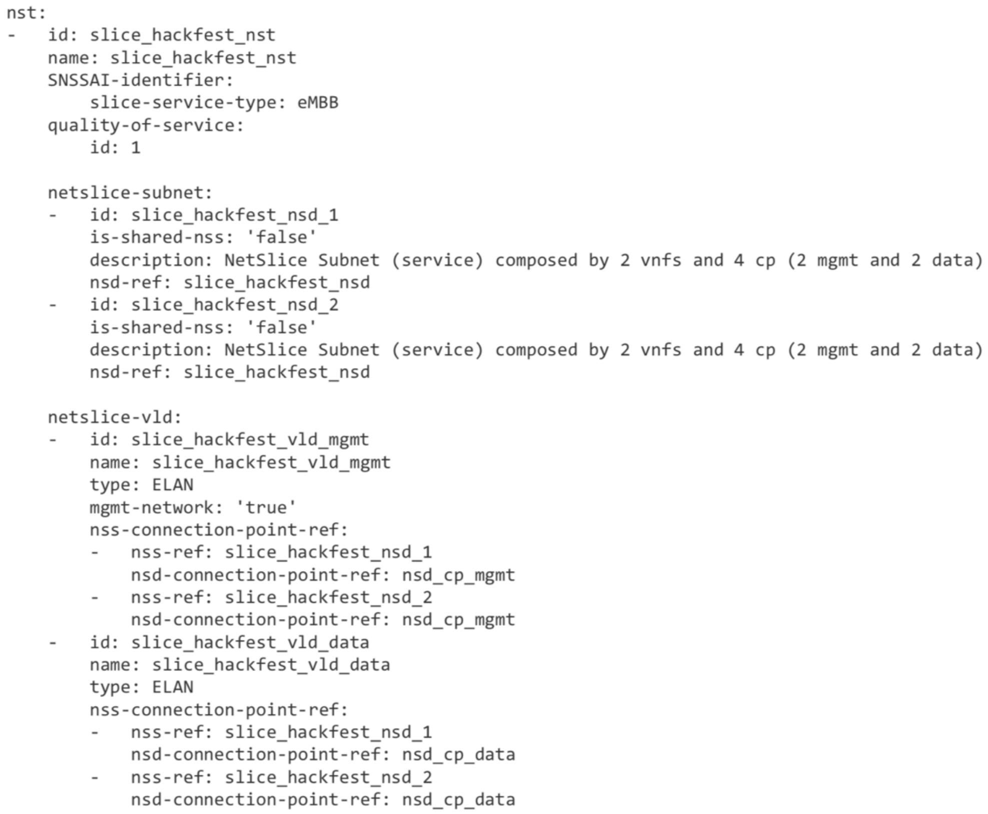


Figure 6-15 NST diagram



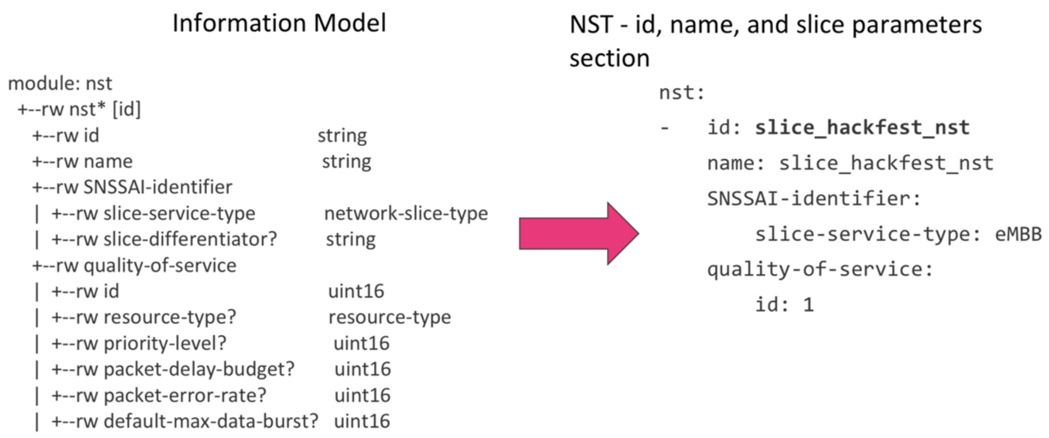
```
nst:
-   id: slice_hackfest_nst
    name: slice_hackfest_nst
    SNSSAI-identifier:
        slice-service-type: eMBB
    quality-of-service:
        id: 1

    netslice-subnet:
    -   id: slice_hackfest_nsd_1
        is-shared-nss: 'false'
        description: NetSlice Subnet (service) composed by 2 vnfs and 4 cp (2 mgmt and 2 data)
        nsd-ref: slice_hackfest_nsd
    -   id: slice_hackfest_nsd_2
        is-shared-nss: 'false'
        description: NetSlice Subnet (service) composed by 2 vnfs and 4 cp (2 mgmt and 2 data)
        nsd-ref: slice_hackfest_nsd

    netslice-vld:
    -   id: slice_hackfest_vld_mgmt
        name: slice_hackfest_vld_mgmt
        type: ELAN
        mgmt-network: 'true'
        nss-connection-point-ref:
        -   nss-ref: slice_hackfest_nsd_1
            nsd-connection-point-ref: nsd_cp_mgmt
        -   nss-ref: slice_hackfest_nsd_2
            nsd-connection-point-ref: nsd_cp_mgmt
    -   id: slice_hackfest_vld_data
        name: slice_hackfest_vld_data
        type: ELAN
        nss-connection-point-ref:
        -   nss-ref: slice_hackfest_nsd_1
            nsd-connection-point-ref: nsd_cp_data
        -   nss-ref: slice_hackfest_nsd_2
            nsd-connection-point-ref: nsd_cp_data
```

Figure 6-16 NST information model for the example shown in Figure 6-15

For simplicity, in this example the Parts 3 (i.e. Network slice connection points) and 5 (i.e. network slice FGD) from NST are not shown, although their specification is straightforward. In the following figures, a more detailed view of the Part 1 (Figure 6-17), Part 2 (Figure 6-18 and Figure 6-19) and Part 4 (Figure 6-20) of NST will be presented.



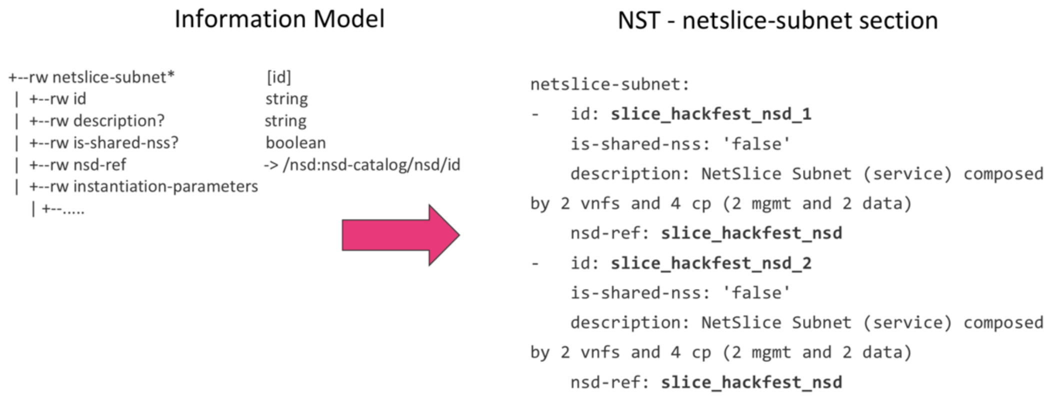
Information Model

```
module: nst
  +--rw nst* [id]
    +--rw id                          string
    +--rw name                        string
    +--rw SNSSAI-identifier
    |  +--rw slice-service-type       network-slice-type
    |  +--rw slice-differentiator?    string
    +--rw quality-of-service
    |  +--rw id                       uint16
    |  +--rw resource-type?           resource-type
    |  +--rw priority-level?          uint16
    |  +--rw packet-delay-budget?     uint16
    |  +--rw packet-error-rate?       uint16
    |  +--rw default-max-data-burst?  uint16
```

NST - id, name, and slice parameters section

```
nst:
-   id: slice_hackfest_nst
    name: slice_hackfest_nst
    SNSSAI-identifier:
        slice-service-type: eMBB
    quality-of-service:
        id: 1
```

Figure 6-17 NST information model – Part 1



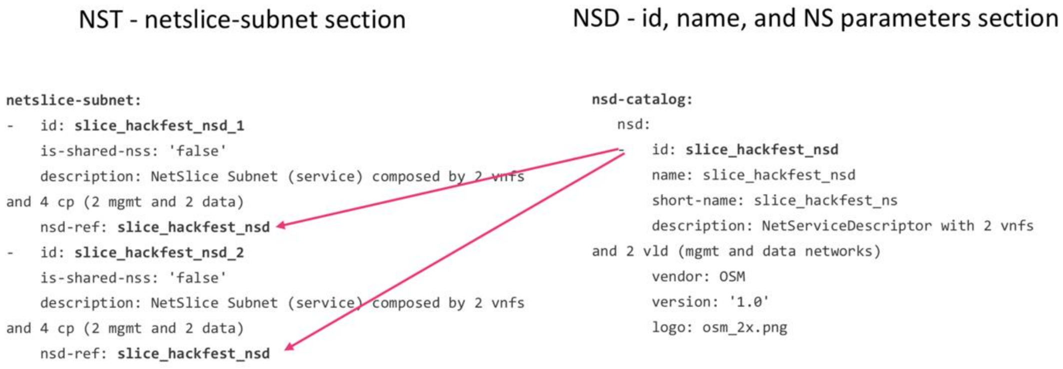
Information Model

```
+--rw netslice-subnet*               [id]
 |  +--rw id                          string
 |  +--rw description?                string
 |  +--rw is-shared-nss?              boolean
 |  +--rw nsd-ref                     -> /nsd:nsd-catalog/nsd/id
 |  +--rw instantiation-parameters
    | +--.....
```

NST - netslice-subnet section

```
netslice-subnet:
-   id: slice_hackfest_nsd_1
    is-shared-nss: 'false'
    description: NetSlice Subnet (service) composed
by 2 vnfs and 4 cp (2 mgmt and 2 data)
    nsd-ref: slice_hackfest_nsd
-   id: slice_hackfest_nsd_2
    is-shared-nss: 'false'
    description: NetSlice Subnet (service) composed
by 2 vnfs and 4 cp (2 mgmt and 2 data)
    nsd-ref: slice_hackfest_nsd
```

Figure 6-18 NST information model - Part 2



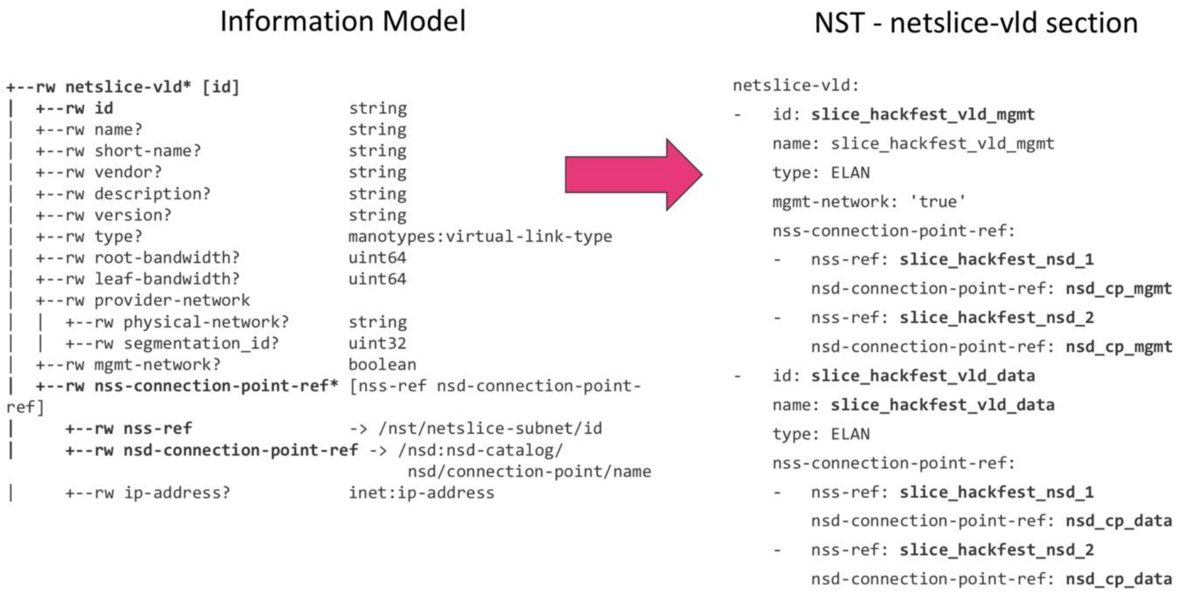
NST - netslice-subnet section

```
netslice-subnet:
-   id: slice_hackfest_nsd_1
    is-shared-nss: 'false'
    description: NetSlice Subnet (service) composed by 2 vnfs
and 4 cp (2 mgmt and 2 data)
    nsd-ref: slice_hackfest_nsd
-   id: slice_hackfest_nsd_2
    is-shared-nss: 'false'
    description: NetSlice Subnet (service) composed by 2 vnfs
and 4 cp (2 mgmt and 2 data)
    nsd-ref: slice_hackfest_nsd
```

NSD - id, name, and NS parameters section

```
nsd-catalog:
  nsd:
  -   id: slice_hackfest_nsd
      name: slice_hackfest_nsd
      short-name: slice_hackfest_ns
      description: NetServiceDescriptor with 2 vnfs
and 2 vld (mgmt and data networks)
      vendor: OSM
      version: '1.0'
      logo: osm_2x.png
```

Figure 6-19 NST information model - Part 2 (focus on NSD)



Information Model

```
+--rw netslice-vld* [id]
|  +--rw id                        string
|  +--rw name?                     string
|  +--rw short-name?               string
|  +--rw vendor?                   string
|  +--rw description?              string
|  +--rw version?                  string
|  +--rw type?                     manotypes:virtual-link-type
|  +--rw root-bandwidth?           uint64
|  +--rw leaf-bandwidth?           uint64
|  +--rw provider-network
|  |  +--rw physical-network?      string
|  |  +--rw segmentation_id?       uint32
|  +--rw mgmt-network?             boolean
|  +--rw nss-connection-point-ref* [nss-ref nsd-connection-point-
ref]
|     +--rw nss-ref                -> /nst/netslice-subnet/id
|     +--rw nsd-connection-point-ref -> /nsd:nsd-catalog/
                                        nsd/connection-point/name
|     +--rw ip-address?            inet:ip-address
```

NST - netslice-vld section

```
netslice-vld:
-   id: slice_hackfest_vld_mgmt
    name: slice_hackfest_vld_mgmt
    type: ELAN
    mgmt-network: 'true'
    nss-connection-point-ref:
    -   nss-ref: slice_hackfest_nsd_1
        nsd-connection-point-ref: nsd_cp_mgmt
    -   nss-ref: slice_hackfest_nsd_2
        nsd-connection-point-ref: nsd_cp_mgmt
-   id: slice_hackfest_vld_data
    name: slice_hackfest_vld_data
    type: ELAN
    nss-connection-point-ref:
    -   nss-ref: slice_hackfest_nsd_1
        nsd-connection-point-ref: nsd_cp_data
    -   nss-ref: slice_hackfest_nsd_2
        nsd-connection-point-ref: nsd_cp_data
```

Figure 6-20 NST information model - Part 4

Annex I SONATA

I.1 SONATA NST Design

The NST can be split into 3 major parts: (1) identification and basic information; (2) the list of NSs (NSDs) and respective details; and (3) the interconnection among NSs.

```
{
   "name":"NST_3_Example1",
   "description":"This is the description of a NST.",
   "version":"3.0",
   "author":"CTTC",
   "vendor":"5GTango",
   "SNSSAI_identifier":{
      "slice-service-type":"eMBB"
   },
   "onboardingState":"ENABLED",
   "operationalState":"ENABLED",
   "usageState":"NOT_IN_USE",
   "5qi_value":3,
   "slice_ns_subnets":[
      {
         "id":"Service_subnet_1",
         "nsd-name":"ns-2-vnf-2-vdu",
         "nsd-vendor":"eu.5gtango",
         "nsd-version":"0.1",
         "sla-name":"None",
         "sla-ref":"None",
         "is-shared":false
      },
      {
         "id":"Service_subnet_2",
         "nsd-name":"ns-2-vnf-2-vdu",
         "nsd-vendor":"eu.5gtango",
         "nsd-version":"0.1",
         "sla-name":"None",
         "sla-ref":"None",
         "is-shared":true
      },
      {
         "id":"Service_subnet_3",
         "nsd-name":"ns-2-vnf-2-vdu",
         "nsd-vendor":"eu.5gtango",
         "nsd-version":"0.1",
         "sla-name":"None",
         "sla-ref":"None",
         "is-shared":false
      }
   ],
   "slice_vld":[
      {
         "id":"mgmt",
         "name":"mgmt",
         "mgmt-network":true,
         "type":"E-LAN",
         "nsd-connection-point-ref":[
            {
               "subnet-ref":"Service_subnet_1",
               "nsd-cp-ref":"mgmt"
            },
            {
               "subnet-ref":"Service_subnet_2",
               "nsd-cp-ref":"mgmt"
            },
            {
               "subnet-ref":"Service_subnet_3",
               "nsd-cp-ref":"mgmt"
            }
         ]
      },
      {
         "id":"slice_input",
         "name":"slice_input",
         "type":"E-LAN",
         "nsd-connection-point-ref":[
            {
               "subnet-ref":"Service_subnet_1",
               "nsd-cp-ref":"input"
            }
         ]
      },
      {
         "id":"slice_output",
         "name":"slice_output",
         "mgmt-network":true,
         "type":"E-LAN",
         "nsd-connection-point-ref":[
            {
               "subnet-ref":"Service_subnet_3",
               "nsd-cp-ref":"output"
            }
         ]
      }
   ]
}
```

The initial part of the NST identifies the Network Slice by a name, description, version, author and vendor and type (SNSSAI_identifier), and other information like a brief description. In addition, the *usageState* indicates whether the NST is in use; i.e. has NSIs and other flags for future use.

```
   "name":"NST_3_Example1",
   "description":"This is the description of a NST.",
   "version":"3.0",
   "author":"CTTC",
   "vendor":"5GTango",
   "SNSSAI_identifier":{
      "slice-service-type":"eMBB"
   },
   "onboardingState":"ENABLED",
   "operationalState":"ENABLED",
   "usageState":"NOT_IN_USE",
   "5qi_value":3,
```

The second part lists the NSs that comprises that Network Slice (three), identifying the NS (NSD) with id, name, vendor, version, etc. (*id, nsd-name, nsd-vendor, nsd-version*), as well as the SLA to be used (*sla-name/sla-ref*) and indicating whether the NS is to be shared among multiple NSIs or is instantiated one per Network Slice (*is-shared*).

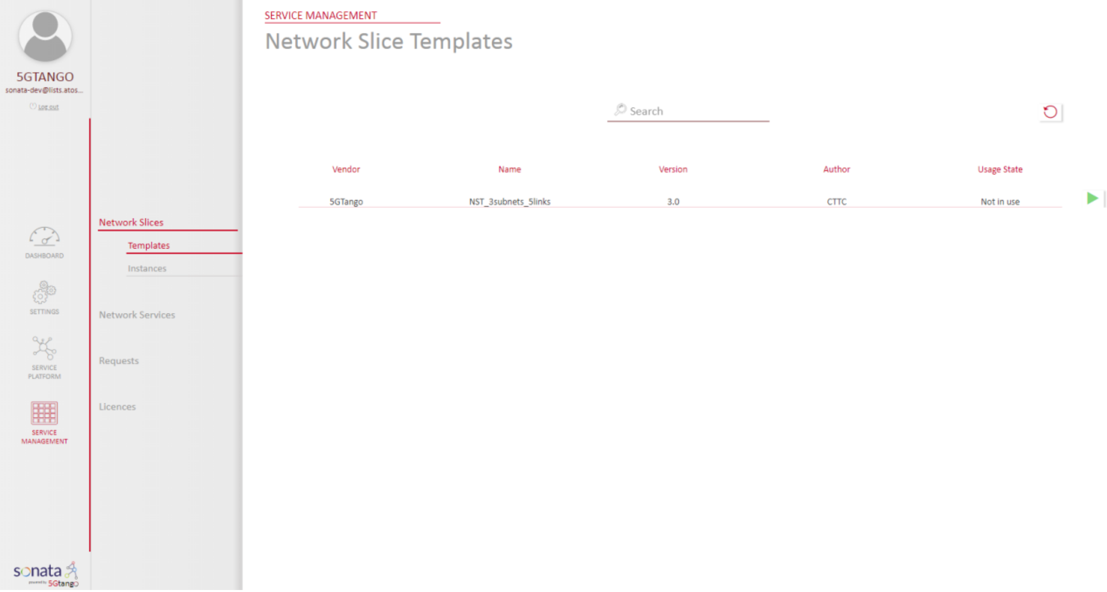
```
   "slice_ns_subnets":[
      {
         "id":"Service_subnet_1",
         "nsd-name":"ns-2-vnf-2-vdu",
         "nsd-vendor":"eu.5gtango",
         "nsd-version":"0.1",
         "sla-name":"None",
         "sla-ref":"None",
         "is-shared":false
      },
      {
         "id":"Service_subnet_2",
         "nsd-name":"ns-2-vnf-2-vdu",
         "nsd-vendor":"eu.5gtango",
         "nsd-version":"0.1",
         "sla-name":"None",
         "sla-ref":"None",
         "is-shared":true
      },
      {
         "id":"Service_subnet_3",
         "nsd-name":"ns-2-vnf-2-vdu",
         "nsd-vendor":"eu.5gtango",
         "nsd-version":"0.1",
         "sla-name":"None",
         "sla-ref":"None",
         "is-shared":false
      }
   ],
```

The final part lists the links that interconnect NSs among them. Each link is identified by an *id*, a *name* and a *type*, and describes the connections points of the NSs associated to this link. Links can be local of remote depending in the where each NS is deployed.

```
   "slice_vld":[
      {
         "id":"mgmt",
         "name":"mgmt",
         "mgmt-network":true,
         "type":"E-LAN",
         "nsd-connection-point-ref":[
            {
               "subnet-ref":"Service_subnet_1",
               "nsd-cp-ref":"mgmt"
            },
            {
               "subnet-ref":"Service_subnet_2",
               "nsd-cp-ref":"mgmt"
            },
            {
               "subnet-ref":"Service_subnet_3",
               "nsd-cp-ref":"mgmt"
            }
         ]
      },
      {
         "id":"slice_input",
         "name":"slice_input",
         "type":"E-LAN",
         "nsd-connection-point-ref":[
            {
               "subnet-ref":"Service_subnet_1",
               "nsd-cp-ref":"input"
            }
         ]
      },
      {
         "id":"slice_output",
         "name":"slice_output",
         "mgmt-network":true,
         "type":"E-LAN",
         "nsd-connection-point-ref":[
            {
               "subnet-ref":"Service_subnet_3",
               "nsd-cp-ref":"output"
            }
         ]
      }
   ]
```

I.2 SONATA Catalogue Management Interface

The information of the Catalogue can be visualized on the Portal, listing the available NSDs and NSTs. This information can be seen in the Portal by both the Slice designer and the customer. Figure 6-21 and Figure 6-22 depict the Portal look and feel for listing NSTs and NSDs, respectively (customer view). Associated to each template, the Portal enables (green arrow) the deployment of instance of that Descriptor (Service instance) or Template (Network Slice Instance).

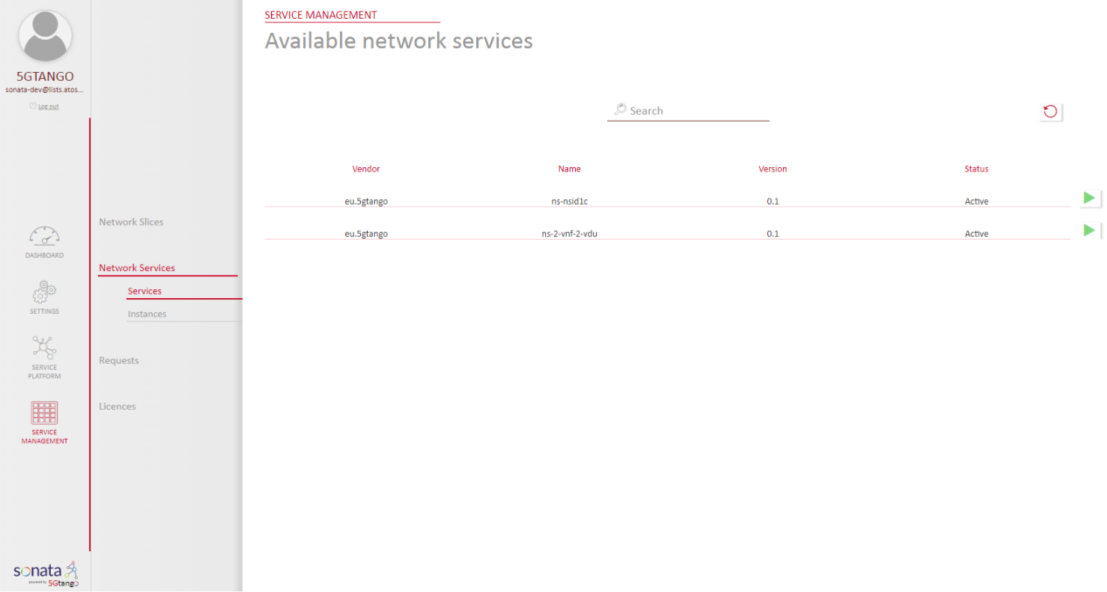


SERVICE MANAGEMENT

Network Slice Templates

Vendor	Name	Version	Author	Usage State
5GTango	NST_3subnets_5links	3.0	CTTC	Not in use

Figure 6-21 SONATA Release 5.0 Portal: NSTs listing.



SERVICE MANAGEMENT

Available network services

Vendor	Name	Version	Status
eu.5gtango	ns-nsid1c	0.1	Active
eu.5gtango	ns-2-vnf-2-vdu	0.1	Active

Figure 6-22 SONATA Release 5.0 Portal: NSDs Listing.