H2020 5GASP Project

Grant No. 101016448

D3.1 5GASP experimentation services, middleware and multi-domain facilities continuous integration

Abstract

This document presents the 5GASP experimentation facing services, the specific implementation actions to create a multi-domain NFV/SDN fabric and the necessary facilities adaptations to support the introduced open, standardized and unified interfaces. Based on the first 5GASP reference architecture (D2.1), NetApp Onboarding and Deployments Services (NODS) are the founding pillar of the 5GASP ecosystem providing a single entry-point to NetApp developers. The establishment of NODS actors provides guidance towards the extraction of implementation requirements, internal architecture and the interaction with its adjacent components through well-defined interfaces. Furthermore, this document sets the fundamental requirements to sculpture the underlying multi-domain fabric and introduces the implementation approaches of the mature 5G facilities, constituting the overall 5GASP infrastructure.

Document properties

Document number	D3.1
Document title	5GASP experimentation services, middleware and multi-domain facilities continuous integration
Document responsible	K. Trantzas, C. Tranoris
Document editor	K. Trantzas, C. Tranoris
Editorial team	University of Patras (UoP)
Target dissemination level	PU
Status of the document	Final Version
Version	1.0

Document history

Revision	Date	Issued by	Description
0.1	24.06.21	UoP	Initial ToC draft
0.2	16.08.21	UoP	Initial draft
0.3	21.09.21	UoP	Final draft
1.0r	22.09.21	VMware, ININ	Internal review
1.0	01.10.21	UoP	Submission version

Disclaimer

This document has been produced in the context of the 5GASP Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement number 101016448.

Document authors

Company	Name	Contribution
UoP	Kostis Trantzas Christos Tranoris Ioannis Chatzis	Abstract, Introduction, Supported onboarding and deployment model, Identified Actors of NODS, 5GASP NODS architecture and design, Interaction with 5GASP ecosystem, Interaction with CI/CD service – Interface E2, Interface E1 and Interface E5, Interaction with NetApp Marketplace – Interface E3, 5GASP NODS Implementation Release 1, Implementation of E1 Interface, Implementation to support hosting network slices, Patras facility
ITAV	Diogo Gomes João Alegria José Quevedo Rui L. Aguiar	5GASP Network Orchestrator, 5GASP multi-domain network fabric, Facilities interconnection, Interfaces E4 and E7, Aveiro facility
ININ	Rudolf Sušnik Janez Sterle Luka Koršič	5GASP multi-domain network fabric Requirements, Implementation to support hosting network slices, Proposals to support vertical's needs, Ljubljana facility, Document internal review
OdinS	Jorge Gallego-Madrid Ana Hermosilla Antonio Skarmeta	Test descriptor, Interaction with CI/CD service – Interface E2, Implementation to support hosting network slices (Murcia facility), Proposals to support vertical needs (Murcia site)
UNIVBRIS	Abubakar Siddique Muqaddas, Amin Emami, Navdeep Uniyal, Xenofon Vasilakos	5GASP multi-domain NFV fabric, Bristol facility
ORANGE	Oana Badita Oproiu Elena-Madalina Marius Iordache	Identified Actors of NODS, Interaction with 5GASP ecosystem, Technologies, Bucharest Facility
VMware	Vesselin Arnaudov Radoslav Gerganov	Interconnection security aspects, Document internal review
YoGoKo	Yakub Abualhoul	Conclusion

Contents

List of Figures

List of Tables

List of Acronyms

3GPP	3rd Generation Partnership Project
5G IOPS	5G Isolated Operation for Public Safety
5GASP	5G Application & Services experimentation and certification Platform
AAU	Active Antenna Unit
AMF	Access and Mobility Management Function
API	Application Programming Interface
AUSF	Authentication Server Function
BBU	Baseband Unit
BGP	Border Gateway Protocol
BPMN	Business Process Model and Notation
BSS	Business Support Systems
CFSS	Customer Facing Service Specification
CI/CD	Continuous Integration and Continuous Deployment
CNF	Cloud-native Network Function
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
CSMF	Communications Service Management Function
CU	Centralized Unit
DevOps	Development and Operations
DHCP	Dynamic Host Configuration Protocol
DNN	Data Network Name
DNS	Domain Name System
DU	Distributed Unit
E2E	End-to-End
ELK stack	Elasticsearch, Logstash and Kibana
eMBB	Enhanced Mobile Broadband
ETSI	European Telecommunications Standards Institute
EVE	Evolution and Ecosystem
G-VNF	Gateway VNF
gNB	gNodeB
GRE	Generic Routing Encapsulation
GSMA	Global System for Mobile Communications Association
GST	General Slice Template
IaaS	Infrastructure as a Service
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPsec	Internet Protocol Security
KPI	Key Performance Indicator
L2	Layer 2
L3	Layer 3
LAN	Local Area Network

LCM	Lifecycle Management
LTR	Local Test Repository
M	Month
MAE	Mobile Automation Engine
MANO	Management and Orchestration
MCC	Mobile Country Code
MEC	Multi-Access Edge Computing
MNC	Mobile Network Code
MPLS	Multiprotocol Label Switching
MS	Milestone
MShed	MShed Museum
NEST	Network Slice Template
NetApp	Network Application
NetOr	Cross Domain Network Orchestrator
NF	Network Function
NFV	Network Function Virtualization
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NODS	NetApp Onboarding and Deployment Services
NR	New Radio
NRF	Network Repository Function
NS	Network Service
NSI	Network Slice Instance
NSA	Non Standalone Architecture
NSaaS	Network Slices as a Service
NSD	Network Service Descriptor
NSMF	Network Slice Management Function
NSSF	Network Slice Selection Function
OBU	On-Board Unit
ONAP	Open Network Automation Platform
Open5G-NFV	Open and Inter-Domain 5G NFV-based Reference
OSM	Open Source MANO
OSPF	Open Shortest Path First
OSS	Operations Support Systems
PaaS	Platform as a Service
PCF	Policy Control Function
PCRF	Policy and Charging Rules Function
PLMN	Public Land Mobile Network
PNF	Physical Network Function
PPDR	Public Protection and Disaster Relief
pRRU	pico Remote Radio Unit
Q	Quarter
qMON	Quality Monitoring System
QCI	QoS Class Identifier

QoS	Quality of Service
RAM	Random Access Memory
RAN	Radio Access Network
RESTful	Representational State Transfer
RFSS	Resource Facing Service Specification
RSU	Road Side Unit
RU	Radio Unit
SA	Standalone Architecture
SD	Slice Differentiator
SDN	Software-Defined Networking
SDO	Standard Development Organization
SDR	Software Defined Radio
SD-WAN	Software Defined WAN
SerOr	Service Orchestrator
SIM	Subscriber Identification Module
SLA	Service Level Agreement
SME	Small and Mid-size Enterprise
SMF	Session Management Function
SSH	Secure Shell Protocol
SST	Slice/Service Type
SSTP	Secure Socket Tunneling Protocol
TAC	Tracking Area Code
TCP	Transmission Control Protocol
TE	Traffic Engineering
TLS	Transport Layer Security
TM Forum	Tele Management Forum
TOSCA	Topology and Orchestration Specification for Cloud Applications
UC	Use Case
UDM	Unified Data Management
UDP	User Datagram Protocol
UDR	Unified Data Repository
UE	User Equipment
UI	User Interface
UPF	User Plane Function
vApp	vertical and cross-vertical NetApp
vCPU	Virtual Central Processing Unit
VIM	Virtualized Infrastructure Manager
VLAN	Virtual LAN
VM	Virtual Machine
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VPN	Virtual Private Network
WAN	Wide Access Network
WP	Work Package

WTC	"We The Curious"
YAML	Yet Another Markup Language
YANG	Yet Another Next Generation

Definitions

This document contains specific terms to identify elements and functions that are considered to be mandatory, strongly recommended or optional. These terms have been adopted for use similar to that in IETF RFC2119 and have the following definitions:

- **MUST** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- **MUST NOT** This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
- **SHOULD** This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- **SHOULD NOT** This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
- **MAY** This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option; except of course for the feature the option provides).

1 Introduction

1.1 Objectives of this document

The main objective of this document is to expand in implementational detail the reference architecture designed in Work Package (WP)2 and presented in D2.1 [1] and further describe the technical implementation of the internal architecture of the components introduced as core building blocks of the 5G Application & Services experimentation and certification Platform (5GASP) architecture.

The design of a common platform to accelerate the development, testing and certification of Network Applications (NetApps) is certainly one the main technical goals of the project. This document examines and sets the requirements to accomplish this goal. Initially, a unified interaction model is presented as the single-point entry point to the 5GASP platform. It is meticulously designed and entirely composed of data models introduced by widely accepted telecommunication standard bodies, e.g. Tele Management Forum (TMF), offering not only a familiar ecosystem for developers to work with, but ensuring interoperability and reproducibility with other systems. Moreover, this document identifies all the actors of the platform, in the notion of assembling the particular implementation requirements for each actor towards an outline design that satisfies them in a holistic way.

From this point on, after acquiring the requirements for the platform, the internal architecture of the platform is presented based on a service-based model. An explicit reference for each service, comprising the proposed architecture, provides a more comprehensive view to the reader of this document, while enlightening him about the technology stack that is being used. Also, this is visualised through snapshots of the current release of the 5GASP portal (phase 1 release).

Furthermore, the scope of this document is extended into the investigation of the peripheral components of the portrayed platform, i.e. the novel automated Continuous Integration/Continuous Deployment (CI/CD) toolchain, the multi-domain Network Function Virtualization/Software-Defined Networking (NFV/SDN) fabric and the public registry of Small and Mid-size Enterprises (SME)s and their registered products (marketplace). This involves the identification and definition of the communication interfaces between these components, which are extensively described.

To support the main technical objective of 5GASP, i.e. to build and operate an Open, and Inter-Domain 5G NFV-based Reference (Open5G-NFV) ecosystem of distributed of experimental facilities, our partners' testbeds are made available to SMEs seeking to experiment in developing, testing and validating their NetApps. The requirements, technologies and interconnection options to support the required reproducible environments under a seemingly single facility are investigated. Moreover, security and trust aspects associated with running 3rd party software in experimental network environments are evaluated.

Finally, this document defines the initial version of the implementation capabilities required from each testbed employed in the 5GASP fabric. The evaluation of these capabilities and the

2nd release update of the associated implementation and deployment details shall be discussed in D3.2, followed by the final release in D3.3.

1.2 Approach and methodology

As the reference architecture (D2.1) of the 5GASP platform was driven by design goals defined by key representative industrial partners from relevant infrastructure provider or NetApp sectors, WP3 was enrolled in the validation of the overall process, providing input about the implementation feasibility of the drafted architectural outline and the development process and roadmap, in general.

Started at Month (M)3, the technical work in WP3 was based on the initial requirements drafted by WP2 and aimed at the deployment of a 1st prototype version of experimentation services for architecture validation, as the first milestone. Throughout the process, discussions with involved WP2, WP3, WP4 and WP5 had led to recurring refinements upon the reference architecture and the specifications to be met by the implementation plan. Specifically, the high-level goal for the first milestone (MS)3.1 was attained on time (M6) and additionally, best practices were followed that are well established in major standardization bodies and the industry.

In parallel, having the same start line set at M3 of the project, the task to implement the multi-domain NFV/SDN fabric was strongly engaged by the partners providing their 5G experimental facilities. Evidently, the required management and orchestration stack is successfully deployed by nearly all facilities and significant effort has been put into the establishment of the interconnectivity fabric among them.

Lastly, although this deliverable is being produced shortly after the relevant task of 5GASP facilities adaptation to the 5GASP open, standardized and unified interfaces has initiated, profound research and even, some draft approaches to this direction are made, as it shall be presented throughout this document.

1.3 Document structure

This document is composed of eight (8) chapters. The introduction, being the first one, presents objectives of this documents and the approach and methodology; and the last chapter, i.e. the conclusions sums up the content discussed along with directions for future work.

Based on 5GASP's high level architecture, as introduced in D2.1, Figure 1 provides a mapping of document's chapters in relation to the architecture entities and interfaces.

Chapter 2 presents the 5GASP NetApp Onboarding and Deployment Services (NODS) requirements. These consist of the onboarding and deployment input model, the actors of the Services and the facing aspects and requirements extracted from the list of actors.

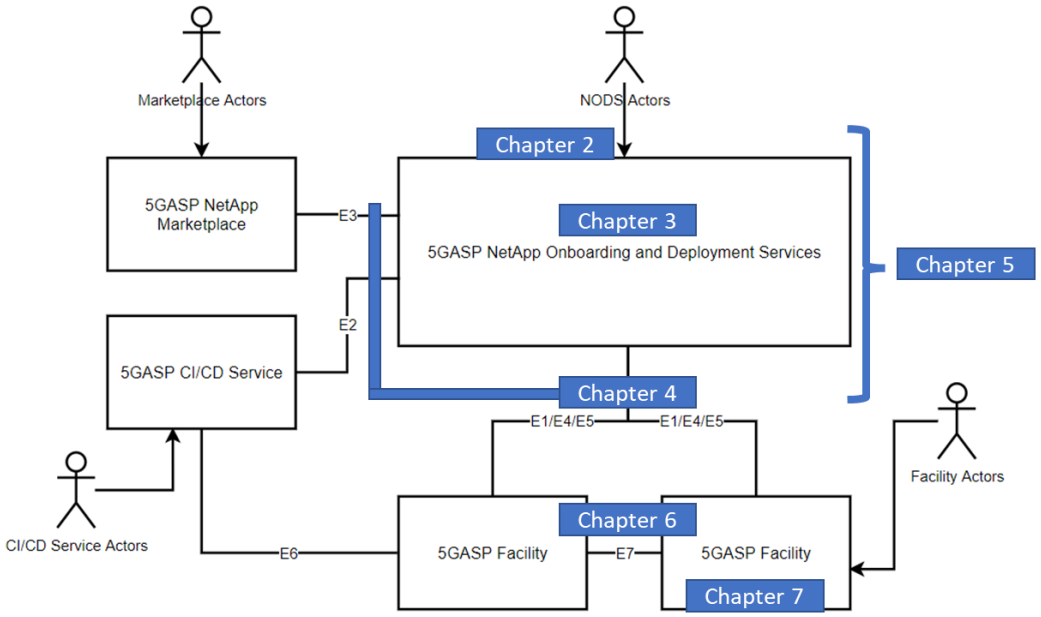
In Chapter 3, the internal architecture of NODS is introduced along with its constituent internal services, followed by the standardized northbound interfaces that are exposed.

The utilised interfaces that enable NODS interaction with the rest of the underlying 5GASP ecosystem are discussed in Chapter 4.

Having introduced a holistic perspective of NODS in previous chapters, Chapter 5 depicts the deployment of the first prototype version of experimentation service for architecture validation and the fulfilment of the previously extracted requirements.

Chapters 6 sets the imperative conditions to be met to address the facility sites interconnection into a multi-domain NFV/SDN fabric, along with its implementation proposal and the emerged security aspects.

Finally, Chapter 7 describes the expected implementation capabilities to be met by each component of the multi-domain fabric, i.e. testbeds, so as to address the specific needs of the verticals' use cases.



Legend

- **E1: Interface for communication to the NFVO (SOL005 , etc)**
- **E2: Interface for CI/CD communication**
- **E3: Interface for NetApp Marketplace interactions**
- **E4: Interface for Cross Domain Network Orchestration**
- **E5: Interface for facility and testing services management**
- **E6: Interface for facility interaction with CI/CD**
- **E7 Inter-facillity Interface connectivity**

Figure 1. Document chapters in relation to 5GASP high level architecture

2 5GASP NetApp Onboarding and Deployment Services (NODS) requirements

2.1 Supported onboarding and deployment model

As 5GASP project aims i) to support the NetApp developer to onboard its NetApp in an effortless and transparent way and ii) to provide an abstraction solution so that onboarding, activation and testing of a NetApp can be properly performed on any NFV/3rd Generation Partnership Project (3GPP) compliant 5G System, besides the 5GASP facilities. Therefore, the need of a unified onboarding and deployment model has emerged.

To that extent, the proposal of a unified standards-based model is attempted that has the form of a "**triplet**" of entities triggering a deployment order, as depicted in Figure 2. This "**triplet**" consists of the following entities:

- The NetApp artefact, that describes the NetApp to be deployed within the corresponding 5G network slice;
- The Network Slice, that needs to be activated by a target 5G facility to meet with the specific requirements of the NetApp;
- The Test Suite, described in terms of a test descriptor model, that will be executed after the activation of the NetApp.

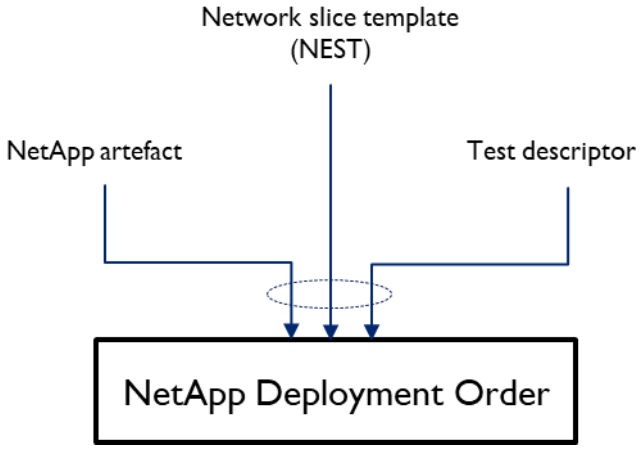


Figure 2. 5GASP onboarding and deployment model

The main aim of the proposed approach is to define each part of the unified model towards the same resource model, that provides a comprehensive description of a given service, including information on its topology and expected behaviour. That being the case, 5GASP's approach for each entity is to be defined under the TMF's Service Specification model [2], being a widely industry utilised class that outlines any type of service through a standardized set of characteristics. Functionally, it acts as a template by which Network Services (NS)s may be instantiated.

The above model entities, converted to Service Specification resource models, can be contained, and listed in respective catalogue(s). The design and maintenance of a service catalogue brings benefits both in terms of reusability and classification of the model's entities

to distinct and explicit groups. Furthermore, the introduction of a catalogue aims to incorporate these templates, which are conceived to provide a self-contained specification of offered service instances, so that their deployment and operation can be automated as much as possible.

Once all entities are expressed to the unified model, then bundling to a single entity can be achieved and progressed through underlying components for fulfilment. The concept of this unique entity can be served through TMF's Service Order model [3]. Once a Service Order is placed, the service fulfilment process is instantiated via a Service Orchestrator (SerOr) which supports the delivery process as per the requested specification, gets the full decomposition up to the required network level operations and executes them onto an administrative domain.

2.1.1 NetApp artefact

To support our TMF model adoption approach, onboarded NetApps will be referred as Resource Facing Service Specifications (RFSSs) expressing the resource aspects of the NetApps with their respective requirements. These NetApps will be described as Virtual Network Function Descriptors (VNFDs) or Network Services Descriptors (NSDs), depending on the defined model (YANG [4] or TOSCA [5]). As enhancements of the NFV architecture towards "Cloud-native" are currently attempted, 5GASP aims to support Cloud-native Network Functions (CNFs), and simultaneously provide effortless transformation for already containerized applications to Virtual Network Functions (VNFs) via Kubernetes Helm Charts, leveraging deployment schemes described in [6]. Prior to a successful onboarding, a set of elementary pre-flight tests, i.e. syntax, semantics, and reference checking will be conducted to certify the validity of the descriptor to be onboarded. Should this step be successful, then the hosting network slice requirements shall be introduced.

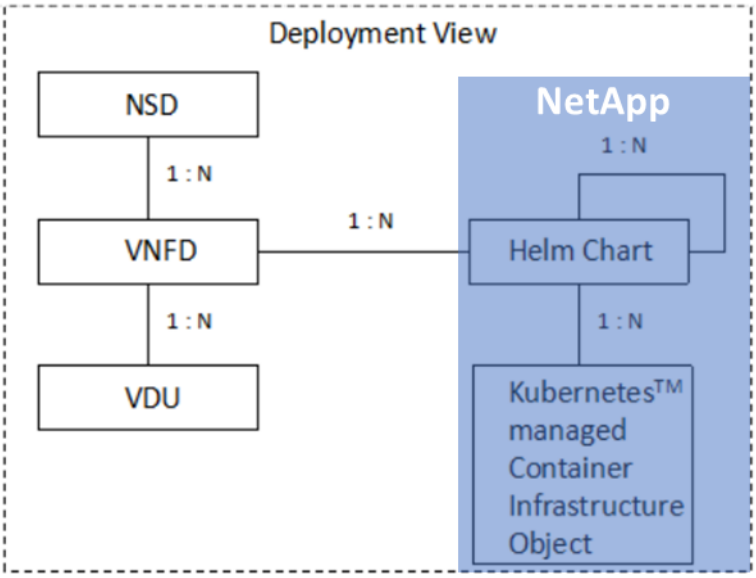


Figure 3. NetApp deployment object mapping (edited from [6])

2.1.2 Network slice template

The hosting network slice requirements need to be defined in a standardized way as well. While onboarding the NetApp, two options can be distinguished for the Network Slice specification to be achieved: i) the developer may select the accommodating site to onboard its NetApp through a list of explicitly reported network slice specifications, or ii) 5GASP system may automatically appoint each NetApp to a target site, based on more abstract network requirements input from the developer. Whichever the appointed case may be, network requirements will be fully aligned with Global System for Mobile Communications Association (GSMA)'s Generic Slice Template (GST) [7] properties and each designated site will provide information on the range of network requirements it supports in form of Network Slice Templates (NESTs). Therefore, these templates will be either available to the developer to choose from or there will be automatically allocated depending on the NetApp, fulfilling its requirements. Each one of the populated properties in NESTs will be perceived as a Service Specification Characteristic [2] resource class, representing a key feature of the Service Specification describing the hosting network slice. An example of a facility provided NEST is depicted in Figure 4.

Example NEST	
Area of service	GR
Area of service: Region specification	Patras
Downlink maximum throughput per UE	128 Kbps
Uplink maximum throughput per UE	10000 Kbps
Isolation level	Virtual resources isolation
Slice quality of service parameters: 3GPP 5QI	69
Supported device velocity	0-1 km/h
User data access	Termination in the private network

Figure 4. Example of a NEST

2.1.3 Test descriptor

As there are not any current standards to define how to describe the testing and validation of NetApps, we have defined a new test descriptor model. It is based on ideas from past projects such as 5GEVE [8] and 5GTANGO [9], and designed to cover the needs of 5GASP. As the project is in its first year, the design is a first version of the test descriptor, according to the requirements we have accumulated from the initial study and design of the 5GASP architecture and the CI/CD service. Also, it supports the three types of tests defined by the project: infrastructure tests, custom tests, and custom test VNFs.

The 5GASP test descriptor will be onboarded (see Figure 5) to 5GASP NODS attached to a TMF Service Test Specification [10], following the TMF standards on which 5GASP NODS is based. The Service Test Specification will contain the needed information to place a test with all the necessary parameters. The specific testing procedure will be described by the 5GASP test descriptor, however, there are deployment data that are required to manage the tests. This information cannot be known beforehand, i.e., in the test descriptor design and development steps.

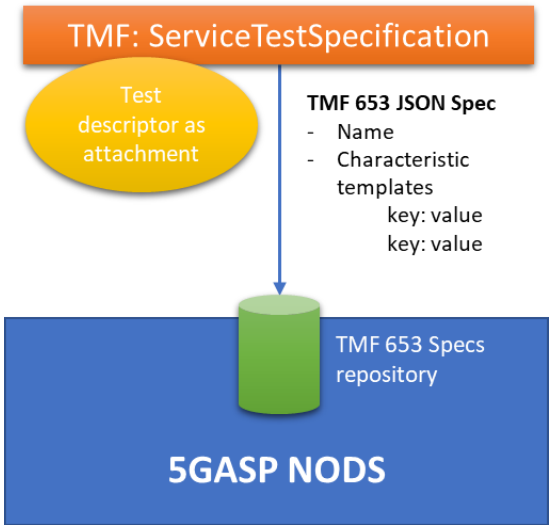


Figure 5. Test descriptor onboarding

The idea is to start working with this preliminary version of the 5GASP test descriptor and evaluate its capabilities with the first test procedures designed by the NetApp owners. Then, with the feedback obtained from the experimenters and from the 5GASP framework developers, we will modify and improve the test descriptor to better suit the project demands.

In this way, the aim was to design a simple, easy-to-use test descriptor that follows the general testing procedure of the project. This test descriptor will include information about the desired testing for a certain NetApp. Therefore, it will be related to a NetApp in a 1-on-1 way, although it can reference multiple individual tests. The test descriptor has been designed to describe the tests that a NetApp must pass in three phases: (i) setup, (ii) execution, and (iii) validation. This is because it perfectly fits the testing pipeline lifecycle, where the pipeline has to be established and prepared, then the tests are triggered and the pipeline executed, and finally the results are gathered and validated. Also, it includes identifiers and metadata about the test descriptor.

To ease the adoption and the generalization of the descriptor we have adopted the approach of parametrizing the configuration values of the tests. In this way, the custom tests will be developed having in mind that the configuration variables will be provided by the test descriptor as parameters. By doing so, the tests will be generic and totally reusable. Whenever an experimenter desires to use a custom test from the repository, the information about the configuration parameters will be pre-provided and the developer will be able to configure the testcase to fit the needs of the NetApp under evaluation. The implementation aspects of the test descriptor are thoroughly detailed in D5.1.

2.2 Identified Actors of NODS

This section aims to pinpoint the potential NODS actors and distinguish their designated use cases in relation to their interaction with it. This input will be used as foundation to extract NODS requirements and shape its architecture further in this document.

2.2.1 Anonymous user

Anonymous user accesses the public resources of NODS without providing a username or password. The supported use cases for the role are presented in Table 1.

Use Case ID	Title	Description
#01	Signing up to NODS	Either creates new account or uses other federated providers, e.g. github
#02	Browsing the public catalogue of services	Browses available supported Customer Facing Service Specifications (CFSSs), i.e. NetApps, Network Slices, Test Descriptors
#03	Signing in to NODS	Logs in to NODS through the corresponding form

Table 1. Anonymous user Use Cases

2.2.2 5GASP NetApp Developer

5GASP NetApp Developer is in charge with developing vertical and cross-vertical NetApps (vApps) uploaded on 5GASP Portal that, after testing and validation, are published on the NetApp store. The supported use cases for the role are presented in Table 2.

Use Case ID	Title	Description
#01	Account management	Browses and manages personal account information
#02	NetApps packages onboarding	Onboards VNF, NSD artefacts and archives
#03	Onboarded packages management	Manages own onboarded VNF, NSD artefacts and archives
#04	NetApp management	Manages the constructed RFSSs of own NetApps out of the onboarded packages
#05	Service Test Specifications creation and (optional) custom test descriptors onboarding	Designs the Service Test Specifications defining the mandatory parameters for CI/CD service to execute the tests. Optionally provides YAML archives (CI/CD artefacts). Alternatively selects tests from the available predefined pool.
#06	Browsing the NODS catalogue of available services	Available services refer to NetApp deployment and testing, e.g. Test a NetApp in a basic Enhanced Mobile Broadband (eMBB) slice
#07	NetApp deployment request to NODS	Deployment requests are captured through a Service Order which includes: • NEST selection, describing the hosting network slice of the NetApp – Area of Service (either automatically based on NetApp requirements or chosen by the developer) • NetApp CFSS

		• Selection of Key Performance Indicators (KPIs) to be measured from the list of available KPIs per testbed or selection of custom tests available in test repository
#08	Service Order view and management	A list of current and past Service Orders is available, along with detailed information per Service Order through its lifecycle
#09	Service Inventory view	A list of services linked to Service Orders are available, along with their run-time information
#10	Notifications about the deployment phases of the NetApp	Notified via email throughout the NetApp deployment lifecycle
#11	Notifications about errors on any phase of the NetApp's deployment	Notified for errors via email throughout the NetApp deployment lifecycle
#12	Access to test results	Selecting a completed NetApp deployment Service Order grants access to the CI/CD phase results

Table 2. 5GASP NetApp Developer Use Cases

There are two main procedures derived from the table above, concerning the 5GASP NetApp Developer; the onboarding and the ordering procedure, which are depicted as activity diagrams in Figure 6 and Figure 7, respectively.

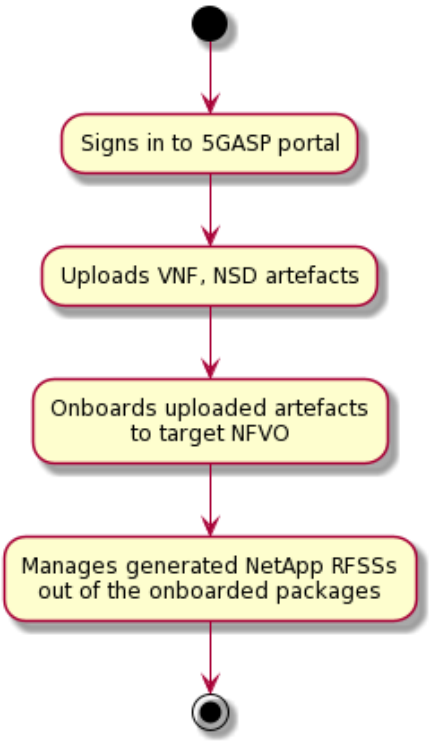


Figure 6. NetApp Developer's onboarding procedure

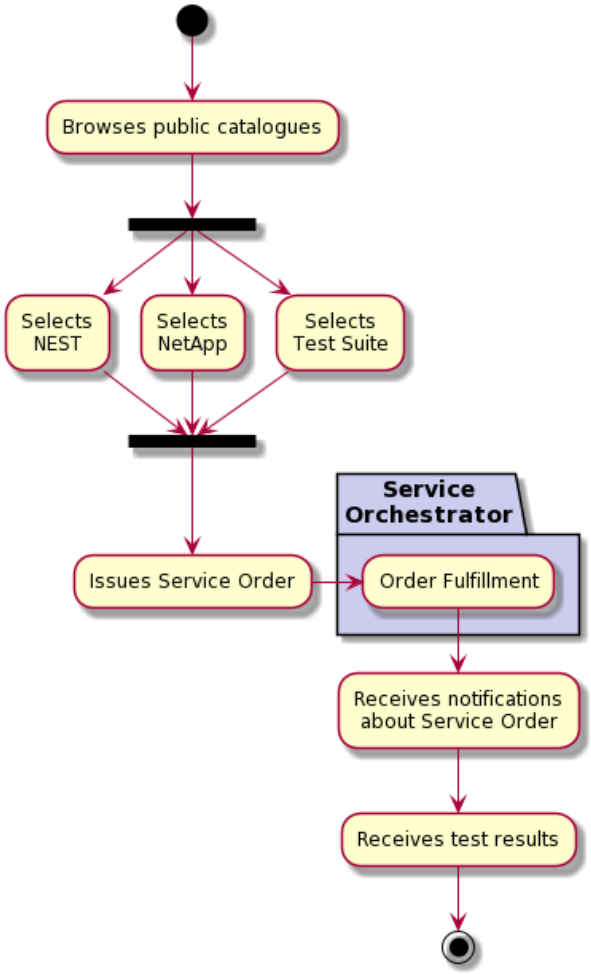


Figure 7. NetApp Developer's ordering procedure

2.2.3 5GASP NF Developer

5GASP Network Function (NF) Developer has the role of designing and onboarding the VNFs and CNFs. It is a similar kind of profile to the NetApp developer but applied to different technical area. While NetApp Developer focuses on service applications, the NF Developer is focused on network related functions. The supported use cases for the role are presented in Table 3.

Use Case ID	Title	Description
#01	Account management	See UC #01 of 5GASP NetApp Developer
#02	NetApps packages onboarding	See UC #02 of 5GASP NetApp Developer
#03	Onboarded packages management	See UC #03 of 5GASP NetApp Developer
#04	Notifications about errors on any phase of the NetApp's deployment	See UC #12 of 5GASP NetApp Developer

Table 3. 5GASP NF Developer Use Cases

2.2.4 Service Designer

The Service Designer role refers to the developer of services composed by 5GASP NetApps and VNFs, including onboarding, service specifications and slices. The supported use cases for the role are presented in Table 4.

Use Case ID	Title	Description
#01	Account management	See UC #01 of 5GASP NetApp Developer
#02	Onboarded packages management	Manages all onboarded VNF, NSD artefacts and archives
#03	NetApp management	Manages the constructed RFSSs of all NetApps out of the onboarded packages
#04	NetApp designing based on other NetApps in the catalogue	Accesses the NetApp catalogue and designs new CFSS bundles containing more than one NetApp
#05	Service Test Specifications creation and custom test descriptors onboarding	See UC #05 of 5GASP NetApp Developer. Provision of YAML archives is mandatory.
#06	Browsing the NODS catalogue of available services	See UC #06 of 5GASP NetApp Developer
#07	NetApp deployment request to NODS	See UC #07 of 5GASP NetApp Developer
#08	Service Order view and management	See UC #08 of 5GASP NetApp Developer
#09	Service Inventory view	See UC #09 of 5GASP NetApp Developer
#10	Notifications about the deployment phases of the NetApp	See UC #10 of 5GASP NetApp Developer
#11	Notifications about errors on any phase of the NetApp's deployment	See UC #11 of 5GASP NetApp Developer
#12	Access to test results	See UC #12 of 5GASP NetApp Developer

Table 4. Service Designer Use Cases

The main process derived from the table above, concerning the Service Designer, is the service design procedure depicted as an activity diagram in Figure 8.

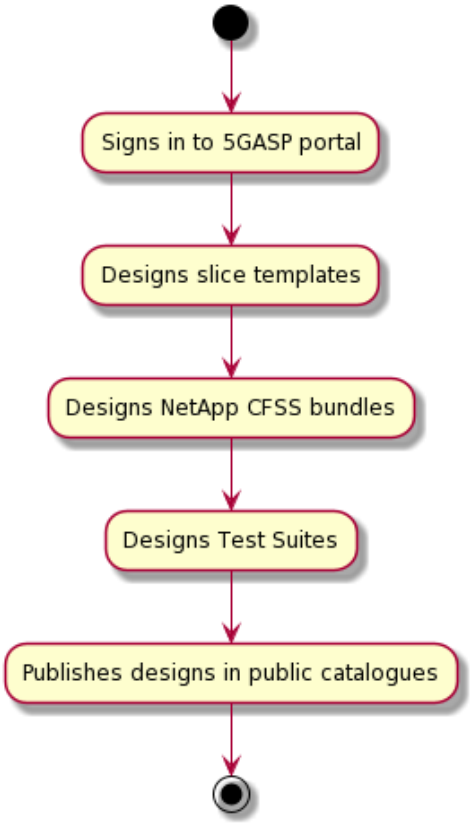


Figure 8. Service Designer's design procedure

2.2.5 Service Provider

The Service Provider offers the service to the verticals' end users.

The use cases of Service Provider are identical to the Service Designer's with the addition of the below. The supplementary use case for the role is presented in Table 5.

Use Case ID	Title	Description
#13	Publication and support of services	Has the responsibility of publishing and supported the offered services in catalogues

Table 5. Service Provider supplementary Use Case

2.2.6 5GASP NODS Platform Administrator

The 5GASP NODS Platform Administrator role refers to the administration of 5G services, operation and maintenance.

The administrator can perform all the previously described use cases but can also support the supplementary use cases presented in Table 6.

Use Case ID	Title	Description
#01	Testbeds management	Has access to exposed services' Application Programming Interfaces (APIs) from testbeds

#02	User account management	Manages platform's user accounts
#03	System messages management	Has access to and handles system information, alerts and errors
#04	Issue management system administration	Has access to and administers the issue management system
#05	Catalogue management	Can create, alter and delete service catalogues

Table 6. 5GASP NODS Platform Administrator supplementary Use Cases

3 5GASP NetApp Onboarding and Deployment Services (NODS) architecture and design

3.1 NODS architecture

This section, based upon the requirements that support the methodology of 5GASP as extracted from the use cases per actor of the previous section (see section 2.2), presents the internal design of 5GASP NODS. It follows the principals of a service-based architecture, as depicted in Figure 9, employing a publish/subscribe (message bus) system, thus offering modularity towards potential system extensions and updates.

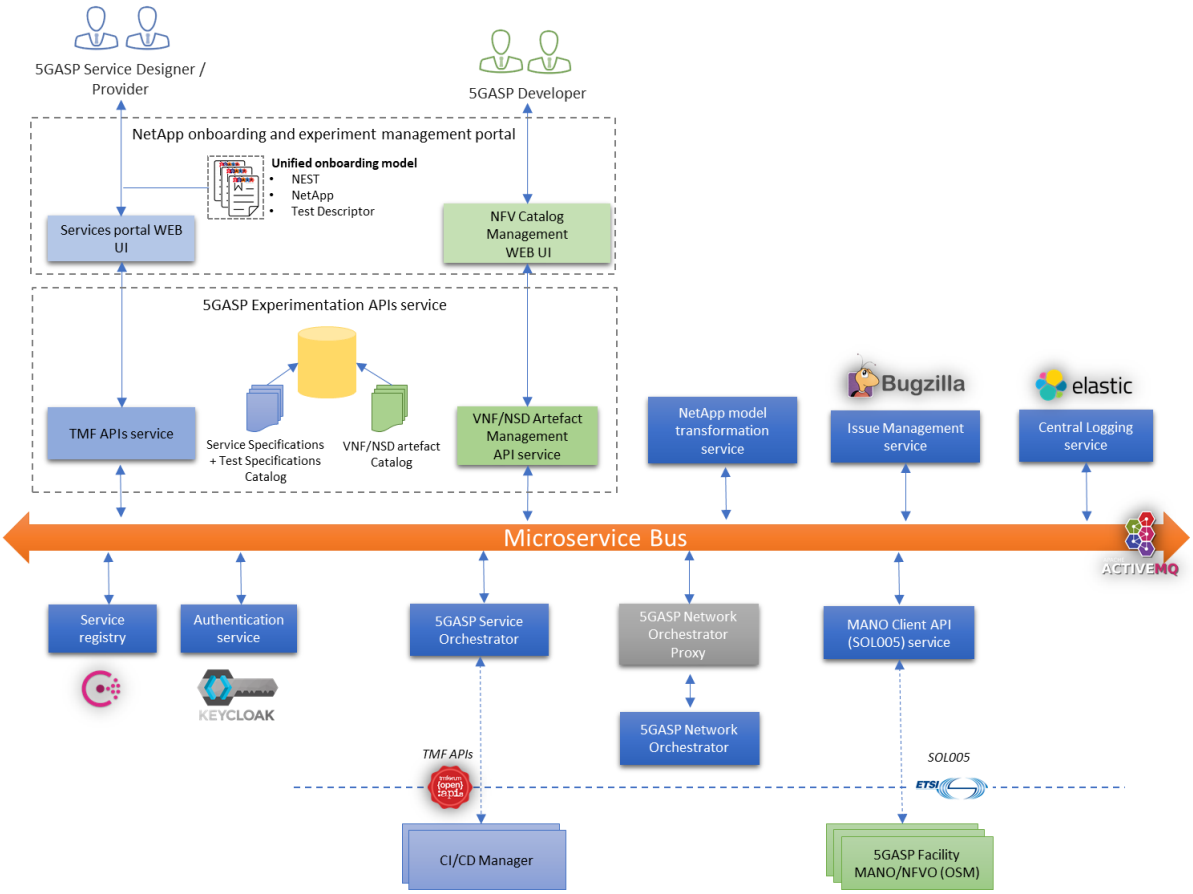


Figure 9. 5GASP NODS architecture

In summary, the 5GASP NODS platform comprises of the following services:

- The NetApp onboarding and experiment management portal,
- The 5GASP experimentation APIs service,
- The service registry,
- The authentication service,
- The model transformation service,
- The issue management service,
- The central logging service,
- The 5GASP Service Orchestrator,

- The 5GASP Network Orchestrator,
- The Management and Orchestration (MANO) client API service,
- The microservices bus.

The 5GASP NODS portal solution is based on the open-source project Openslice [11], further supporting its development to meet the project's requirements. Openslice employs a service-based architecture and design which utilises 3GPP and European Telecommunications Standards Institute (ETSI) standards, along with TM Forum Open APIs to deliver Network Slices as Services (NSaaS). Also, it offers a user-friendly UI, multi-tenancy, support for onboarding VNFs to a target NFV Orchestrator (NFVO) and an Open API based on TMF family of APIs making it a perfect candidate for facilitating the project's needs.

Openslice is currently aligned with TMF909's proposed API Component Suite [12] in support of a set of Operational Domains exposing and managing "Network" Services. This component suite not only covers the functionalities required by Operational Domains to interwork with OSS/BSS applications and/or other domains from service providers or 3rd parties, but it also accommodates one of the key requirements of Openslice, that is the reusability of API functionality rather than using a large set of specific APIs.

Openslice's key contribution is that allows SMEs or organisations to implement private network scenarios that interact with public large-scale networks in a standardized manner.

3.2 Internal services design

Following the previous section, which concisely mentioned the internal components of 5GASP NODS architecture, all the internal services of 5GASP NODS are introduced and further presented below.

3.2.1 NetApp onboarding and experiment management portal

The NetApp onboarding and experiment management portal provides a single entry-point to relevant actors. Its mission is to add an abstraction layer on top of the interconnected individual administrative domains, hiding the mechanisms, protocols and technologies adopted in every site, exposing the facility as a unified platform through a user-friendly User Interface (UI). As it has been reproduced in Figure 10, the portal consists of:

- A NFV catalogue management web UI,
- A services portal web UI.

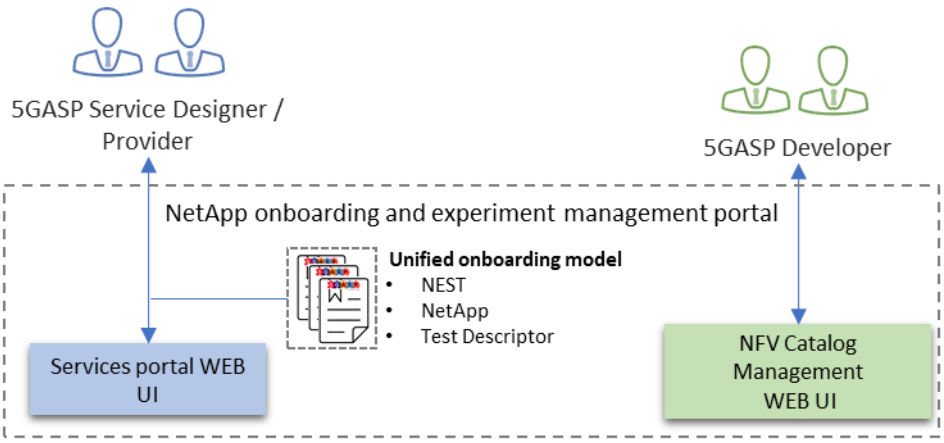


Figure 10. NetApp onboarding and experiment management portal as a single entry-point

The NFV catalogue management UI allows users to onboard and manage the NetApp packages, i.e. VNF/NSD artefacts. Subsequently, the packages are assembled in corresponding catalogues. Also, the 5GASP platform administrator is provided with a management interface of the underlying administrative domains, thus allowing the incorporation of the individual exposed services per domain to the unified facility.

The services portal web UI allows respective actors to leverage the automatically constructed RFSSs, out of the onboarded artefacts, combining them to design new CFSSs. These CFSSs are then exposed to relevant catalogues constituting the NetApp entity of the onboarding model (Section 2.1). In a similar way, services portal enables the outlining of the Service Test specifications and the provision of the test descriptors. Moreover, service providers can design and offer the available network slices (NESTs), thus putting together the whole onboarding model. All the aforementioned entities are assembled and exposed under relative catalogues, which are made available to users. Consequently, portal users can issue NetApp deployment requests, composed of catalogues' entities, receive notification throughout the whole deployment lifecycle and eventually be granted access to the test results.

3.2.2 5GASP experimentation APIs service

The 5GASP experimentation APIs service is composed by a 3rd party VNF/NSD management API as well as TMF family APIs, as seen in Figure 11.

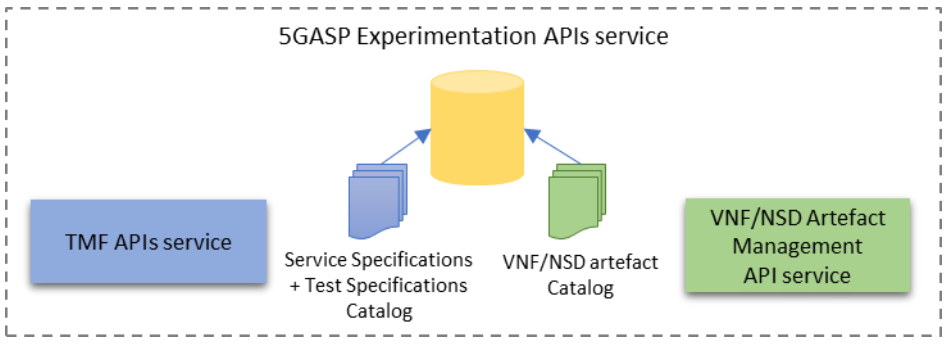


Figure 11. 5GASP experimentation APIs service components

The VNF/NSD management API service is charged with the support of actions performed in NFV catalogue management side of the portal. Specifically, it supports the infrastructure as

well as the NFV/NSD artefact management requirements. Also, it maintains the respective VNF/NSD artefact catalogue.

On the other hand, TMF APIs service offers TM Forum's Open APIs to allow consumption of service catalogues' exposed capabilities. The employed Open APIs include Service Catalogue, Ordering, Inventory, Resource Catalogue, Service Test, Product Catalogue management APIs. Evidently, this Open APIs family supports the service side of the portal. All actions performed are reflected to TM Forum's resource models.

3.2.3 Service registry

This registry provides a one stop solution for typical procedures in microservice architectures, including service (self) registration, discovery, key-value store and load balancing. These requirements are met by employing Consul [13]. Consul offers a service mesh solution which facilitates all the above features either individually or all together under a full-service mesh.

In brief, the key features supported are:

- Service discovery: Consul clients can register a service, even themselves, and enable other clients to discover providers of a given service.
- Health check: Several health checks are provided, associated with the health of a service as a whole or with a local node's resource utilization. This information can be leveraged for cluster health estimation or traffic redirection away from unhealthy hosts.
- Key-value store: Consul offers a central hierarchical key-value store repository that can serve a number of purposes, e.g. dynamic configuration, coordination, etc.

3.2.4 Authentication service

This service provides the authentication and authorization management capabilities, which enables the management of users and roles that are allowed to have access both to the portal and the Representational State Transfer (REST) APIs. This allows an administrator to define which users belong to particular roles and, hence, assign permissions to allow access to specific resources and actions. All APIs (expect the grant token request) require a Bearer token in request header, indicating an authorized user. Authentication is based on oAuth2 [14], which is implemented by employing a Keycloak [15] server.

3.2.5 NetApp model transformation service

The NetApp model transformation service is responsible to transform and enhance prototype NetApps that are not 5GASP ready, to be properly packaged for onboarding. One such example is to transform to other supported NFV onboarding models, supporting both modelling and packaging according to the ETSI standards SOL006 (YANG-based model) [16] and SOL001 (TOSCA-based model) [17]. It is expected that this component will be delivered later in the platform, since more experience needs to be acquired concerning the onboarding and CI/CD operation of existing NetApps.

3.2.6 Issue management service

For issue management support, NODS will rely on Bugzilla [18]. Bugzilla is a ticketing tool that allows issue reporting and tracking via tickets to all relevant stakeholders. Figure 12 displays

the overall issue management service architecture integrating Bugzilla as its core and how this tool interacts with other NODS services presenting some distinctive scenarios. It should be noted that Bugzilla tickets will not only be used for bugs/errors, but also for general requests, e.g. Service Order procedure.

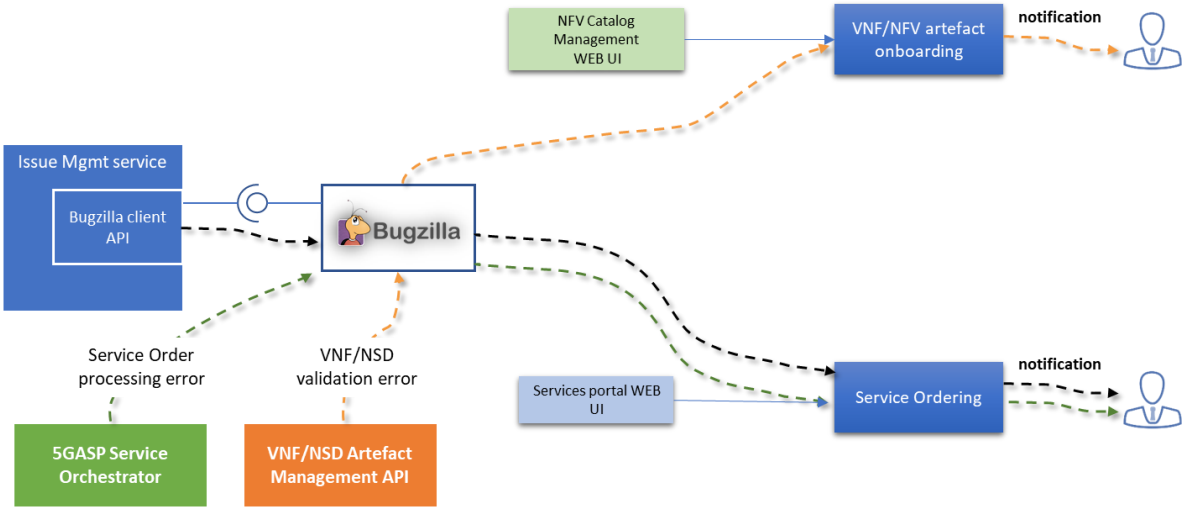


Figure 12. Issue Management service architecture

3.2.7 Central logging service

NODS follows the centralized log management concept, i.e. a type of logging solution system that consolidates the log data from different services and pushes it to a central, accessible and easy-to-use interface. For that reason, ELK stack [19] is elected as an open-source centralized logging solution, which is based on Elasticsearch for collecting, parsing and storing logs. The End-to-End (E2E) ELK stack is formed by employing Elasticsearch, Logstash and Kibana towards a real-time data analytics tool that provides insights from any type of structured and unstructured data source.

3.2.8 5GASP Service Orchestrator

5GASP Service Orchestrator (SerOr) is the principal component of NODS architecture. It is engaged as soon as an issued NetApp Deployment Order (see Section 2.1) is captured. Henceforth, the SerOr is tasked with the support of the deployment decisions to facilities, along with the choreography between employed components of the automation circle, e.g. CI/CD Service. Moreover, it is aware of all the underlying facilities and is tightly coupled with 5GASP Network Orchestrator (see following Section 3.2.9) to lay the groundwork for establishing the inter-domain fabric.

The SerOr uses open-source Flowable business process engine [20] to outline multiple orchestration cases. For example, depicted in Figure 13, the Order fulfilment process is expressed in the form of a BPMN [21] diagram which is consumed by the process engine. In short, the process first checks which tasks can be automatically orchestrated by a NFVO and which require human interaction, if any. Consequently, a local service orchestration process is comprised of inter-domain and cross-domain orchestration requests, which concludes the order fulfilment process, upon its completion.

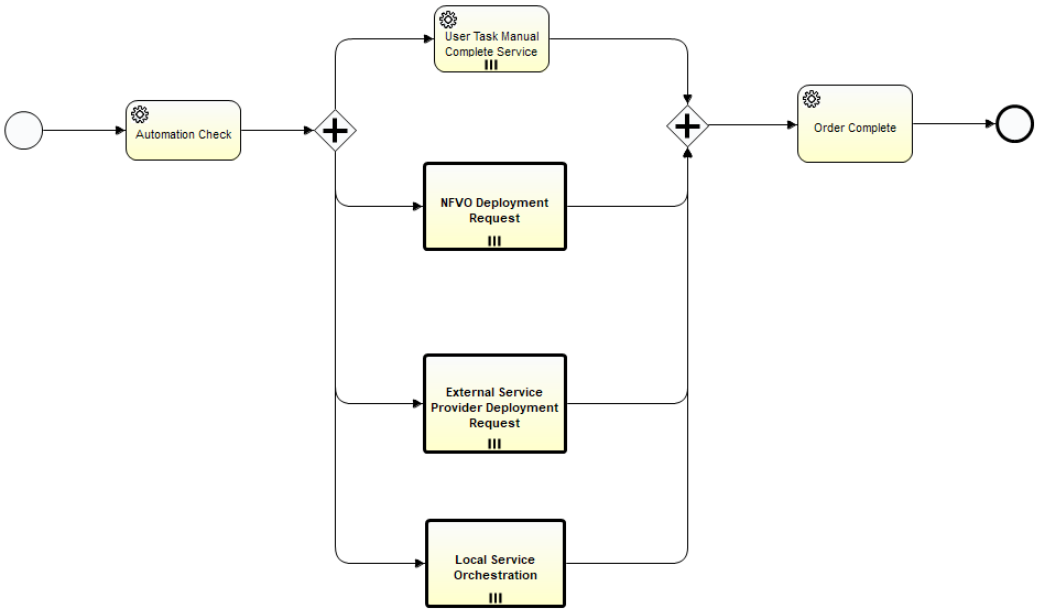


Figure 13. Order Fulfilment process diagram

The above example, along with the remaining cases described by Flowable engine offer a sturdy foundation to support several fundamental orchestration schemes, but it is evident that these designs are static. As 5GASP needs to support dynamic orchestration patterns as well, e.g. test suite must be executed only after the successful deployment of the hosting network slice and the NetApp, the concept of the on-demand orchestration designing is adopted. Specifically, multiple landmark phases are introduced within the Network Slice Instance lifecycle, as seen in Figure 14, i.e. pre-provision, after-activation, supervision, after-deactivation phases.

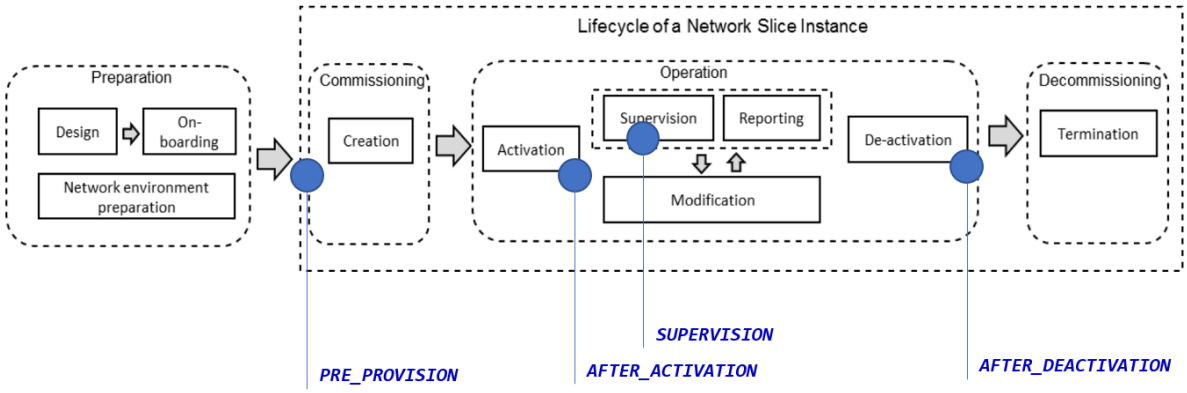


Figure 14. LCM rules phases in relation to the lifecycle of a Network Slice Instance (image edited from [22])

Accordingly, dynamic orchestration is achieved by incorporating lifecycle management (LCM) rules, that execute a specific logic, into the aforementioned phases. These rules are designed in a particular user interface using Blockly library [23], which generates syntactically correct code from interlocking blocks, and then forwards the output code into the orchestration pipeline, as depicted in Figure 15.

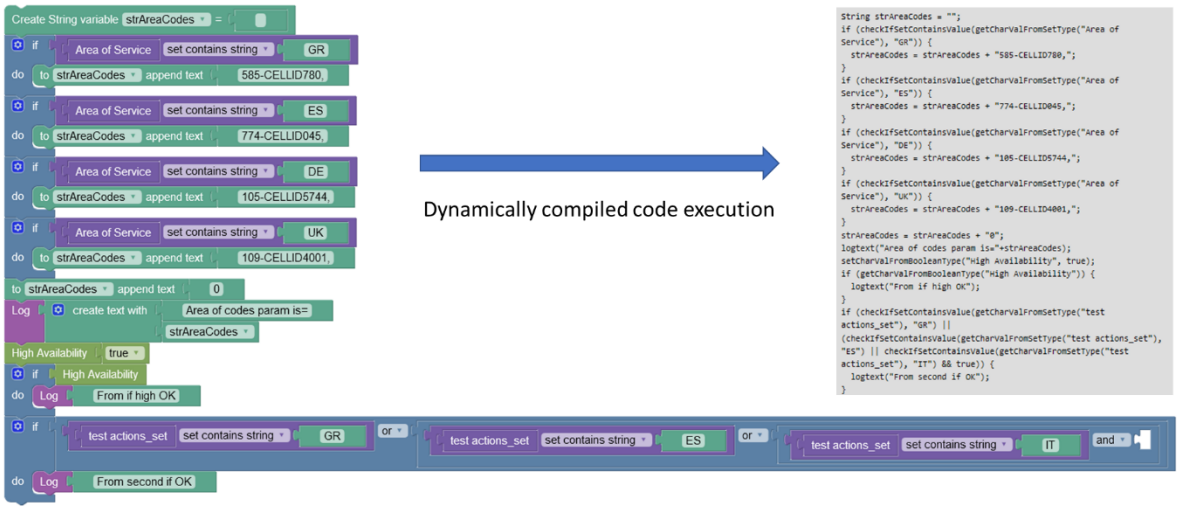


Figure 15. LCM rule design and output code

3.2.9 5GASP Network Orchestrator

Network Orchestrator (NetOr) is by itself a complex service, being composed of different micro-services (Figure 16). By combining that micro-service-oriented architecture with an event-driven approach, this system provides scalability, flexibility, modularity, and efficiency requisites. That is possible by each component managing a different set of entities and functions, exchanging event messages through a centralized message bus to establish the NetOr's internal communication channels.

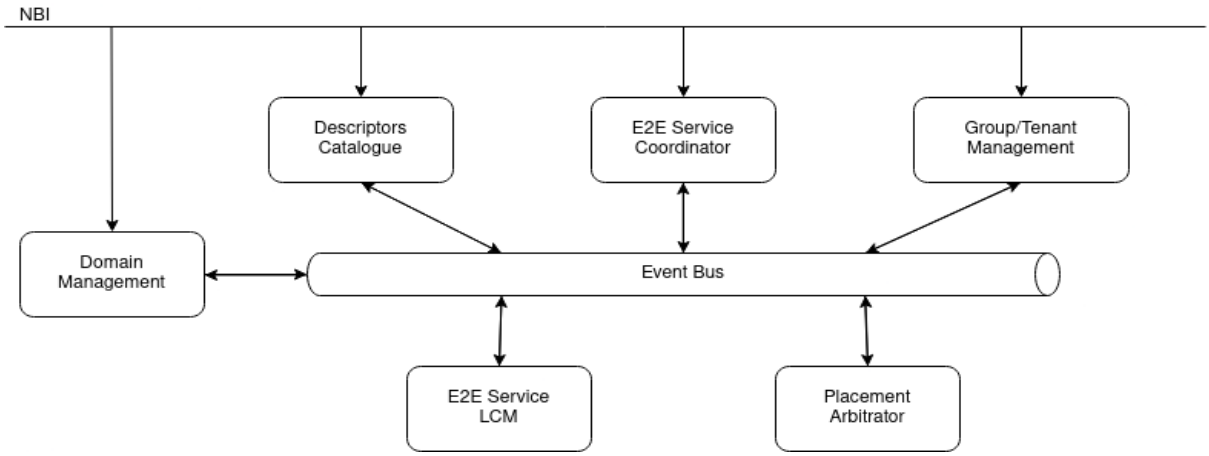


Figure 16. NetOr high-level architecture

With distinct and isolated components, the NetOr system can scale a unique micro-service if needed, adding new workers to manage certain operations and entities. The service considers that scalability possibility, meaning that all micro-services are stateless and thus they don't keep any runtime information. When that functionality is needed, the component uses a memory cache to guarantee that all data is persisted at all times, also serving as a fault-tolerant mechanism capable of handling system crashes.

The NetOr is flexible since it allows the seamless substitution of any sub-component, given that the new component provides the same functionalities and follows the same interface

and standards. Similarly, the system is modular because it allows the addition and removal of micro-services with minimal impact on the rest of the service.

By default, a micro-service-oriented system favours asynchronous communications. Compared to a sequential approach, asynchronous communications allow parallel processing, which may improve the performance and efficiency of a given process. Although allowing parallelization, one drawback from this approach is the constant and increased exchange of administrative messages through the network.

Internally, the system has six main components: the Descriptors Catalogue, the Group/Tenant Manager, the Domain Manager, the E2E Service Coordinator, the E2E Service LCM Manager, and the Placement Arbitrator.

3.2.10 MANO Client API service

The MANO Client API service is an intermediate component that facilitates the communication between NODS and the underlying MANO components, which reside in the engaged facilities. Currently, as the onboarding and packaging model followed is based on YANG model (OSM-powered [24] facilities), the information that needs to be exchanged relies on ETSI SOL005 interface [25] and thus, a corresponding plugin charged with the translation of resource requirements into lifecycle actions in the MANO domain is provided. Later in the project, as TOSCA model is expected to be also supported, the same plugin will support the information exchange with ONAP-powered [26] facilities.

3.2.11 Microservices bus

Since NODS is based on a service-based architecture, a messaging bus is needed to support the internal exchange of messages between micro-services in a loose-coupling manner. Current implementation is based on ActiveMQ [27] and Camel [28] , so micro-services can be implemented in any programming language.

3.3 Northbound standardized interfaces

One of 5GASP objectives is to provide northbound standardized solutions for experiment requests. That is technically possible by encapsulating models from a major Standard Development Organization (SDO), namely the TM Forum. Hence, each entity of the onboarding triplet (NetApp Artifact, Test Descriptor, NEST information) is admitted in a relative catalogue and ultimately exposed publicly, as depicted in Figure 17.

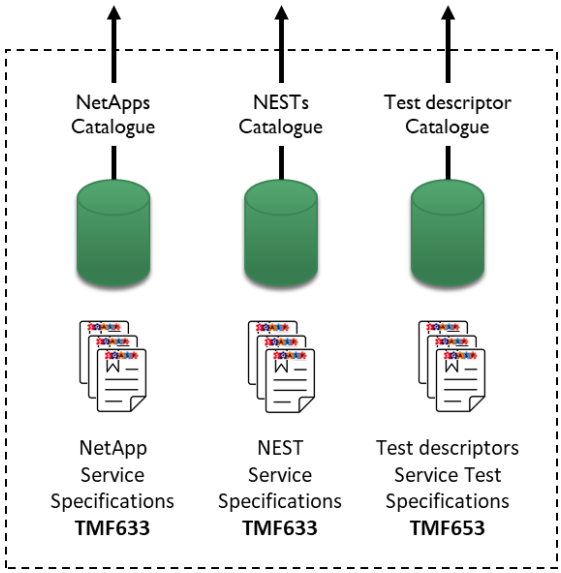


Figure 17. Northbound standardized resource models

As already mentioned, NetApps and NESTs are expressed as Service Specifications, which reside into Service Catalogues based on the relative TMF's resource model [2]. Service Catalogues provide additional grouping through underlying categories. The TMF's Service Catalogue model is elected as the most befitting approach as it is extensively embraced by the telecommunications' industry.

The test descriptor catalogue interface is under development because the test descriptor is still being designed. As detailed in Section 2.1.3, the first version of the descriptor has just been drafted. It is expected that the feedback gathered from the usage of this first approach by the project partners will allow the enhancement of the descriptor. In consequence, the interface for the catalogue remains to be designed, awaiting a more mature version of the descriptor. Nevertheless, there is an agreement to use TMF's Service Test model for infrastructure tests, which can be the first step in the catalogue design.

4 Interaction with 5GASP ecosystem

As it has been set in D2.1, one of the main goals of the 5GASP is to achieve secure/trusted service provisioning and effortless operation taking advantage of experimental facilities featuring virtualized functions. This is achieved by allowing access to a multi-domain fabric in which NetApps can be deployed in minutes across several domains, following a software driven process of testing and validation towards certification.

The 5GASP system interlocks several internal services, alongside their internal and public interfaces to onboard, deploy, facilitate and offer NetApps. The high-level interaction of the actors of these services is depicted in Figure 18.

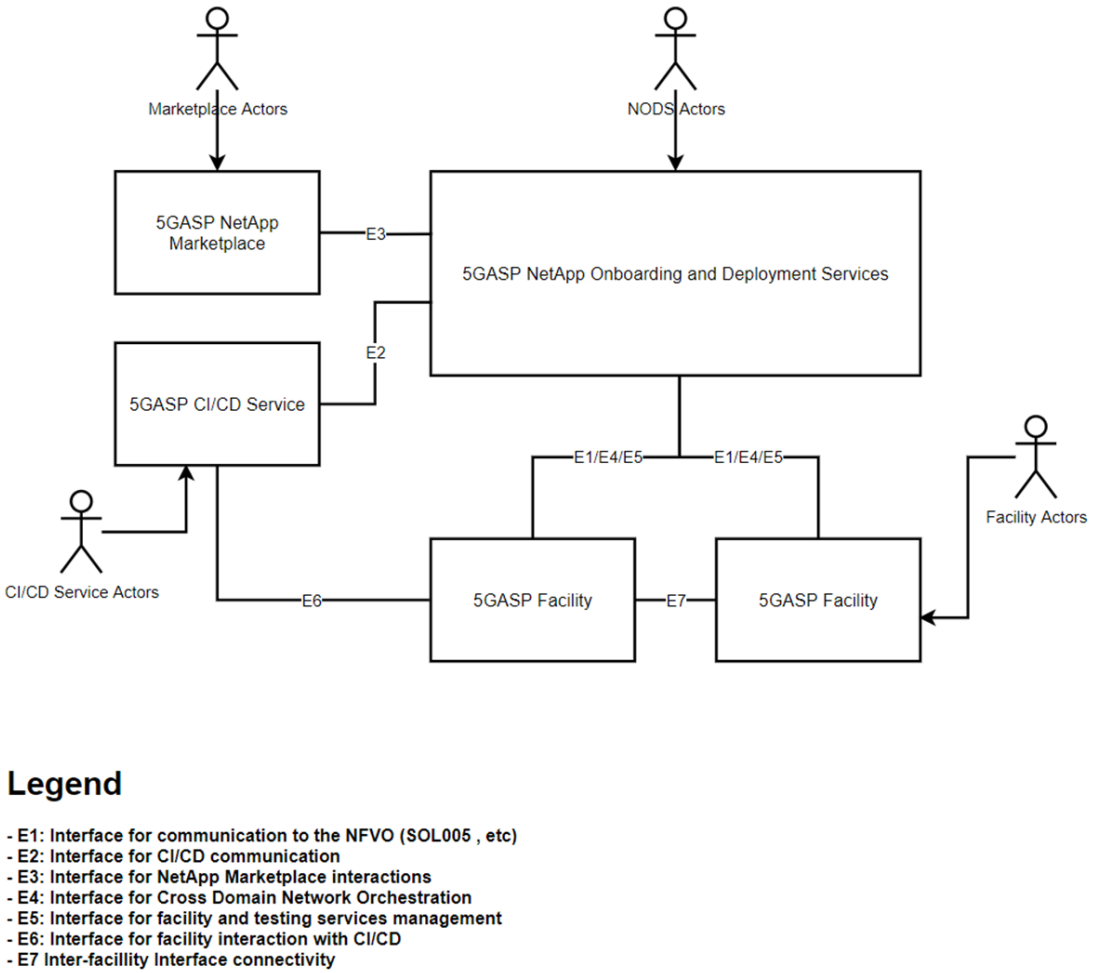


Figure 18. 5GASP high level architecture [1]

As it can be extracted from the figure above, 5GASP follows a service-based architecture. Moreover, 5GASP employs NODS as its central entity (see Section 3). We will define further in this chapter, the external interfaces, that are provided and consumed by the various 5GASP services, related to their interaction with the main component of the architecture: Interaction with CI/CD service (Interface E2 in subsection 4.1), Interaction with Facilities (E1 interface in subsection 4.2.1, E4 interface in subsection 4.2.2, E5 interface in subsection 4.2.3 and E7

interface in subsection 4.2.4) and Interaction with NetApp Marketplace (in subsection 4.3 Interface E3).

Specifically, a greater level of comprehension of the interfaces involved can be achieved through the following interaction diagrams, which describe the nominal use case of deploying a host slice and a NetApp within it (Figure 19) and the nominal test phase (Figure 20) that completes the automation circle. A more detailed overview of the onboarding procedure (Step 2 in Figure 19) can be seen in Figure 21.

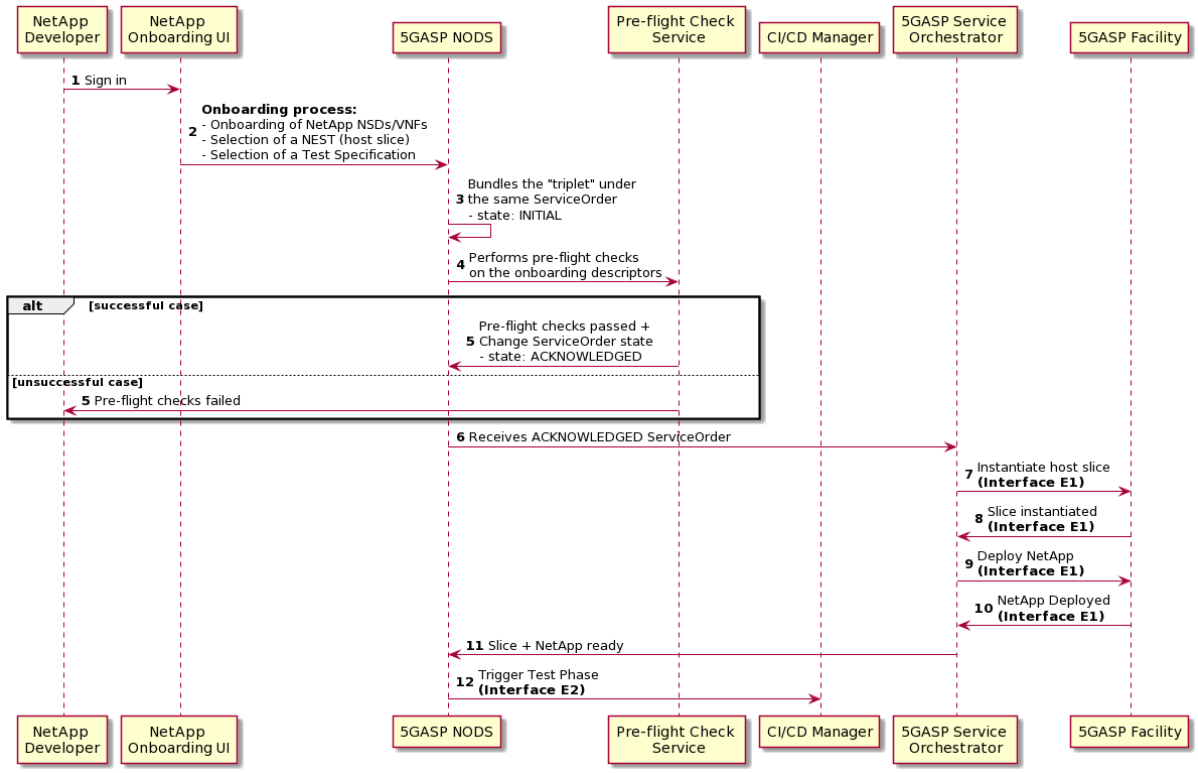


Figure 19. Network slice and NetApp deployment interaction diagram

The nominal deployment process (Figure 19) is initiated by the NetApp Developer signing into a specifically designed User Interface (step 1). This interface can be integrated inside NODS or exist on its own, as the interaction is achieved through standardized interfaces and data models. Next, developer drafts the unified standards-based onboarding model by providing its selections considering NEST information, NetApp Artifact, Testing Descriptor (step 2). Following, 5GASP NODS is charged with bundling these entities under a single service order, rendering its state as “INITIAL” (step 3). At this step, pre-flight checks are triggered and operated over the provided descriptors (step 4). It should be noted that the pre-flight check enabling service can either be once again integrated in NODS or alternatively, it can be employed as an external service. Upon successful completion of pre-flight checks, the service order’s state is switched to “ACKNOWLEDGED” (step 5). 5GASP Service Orchestrator, which is polling for acknowledged service orders, is then handed over the onboarding model (step 6) and orchestration process is initiated. Specifically, Service Orchestrator instantiates the host network slice and deploys the NetApp, both described in the onboarding model, through its internal orchestration graphs (step 7-10) leveraging interface E1 (Section 4.2.1) to

communicate with involved facilities. Once, the orchestration procedure is fulfilled, NODS is notified (step 11) and the test phase is triggered through CI/CD manager (step 12).

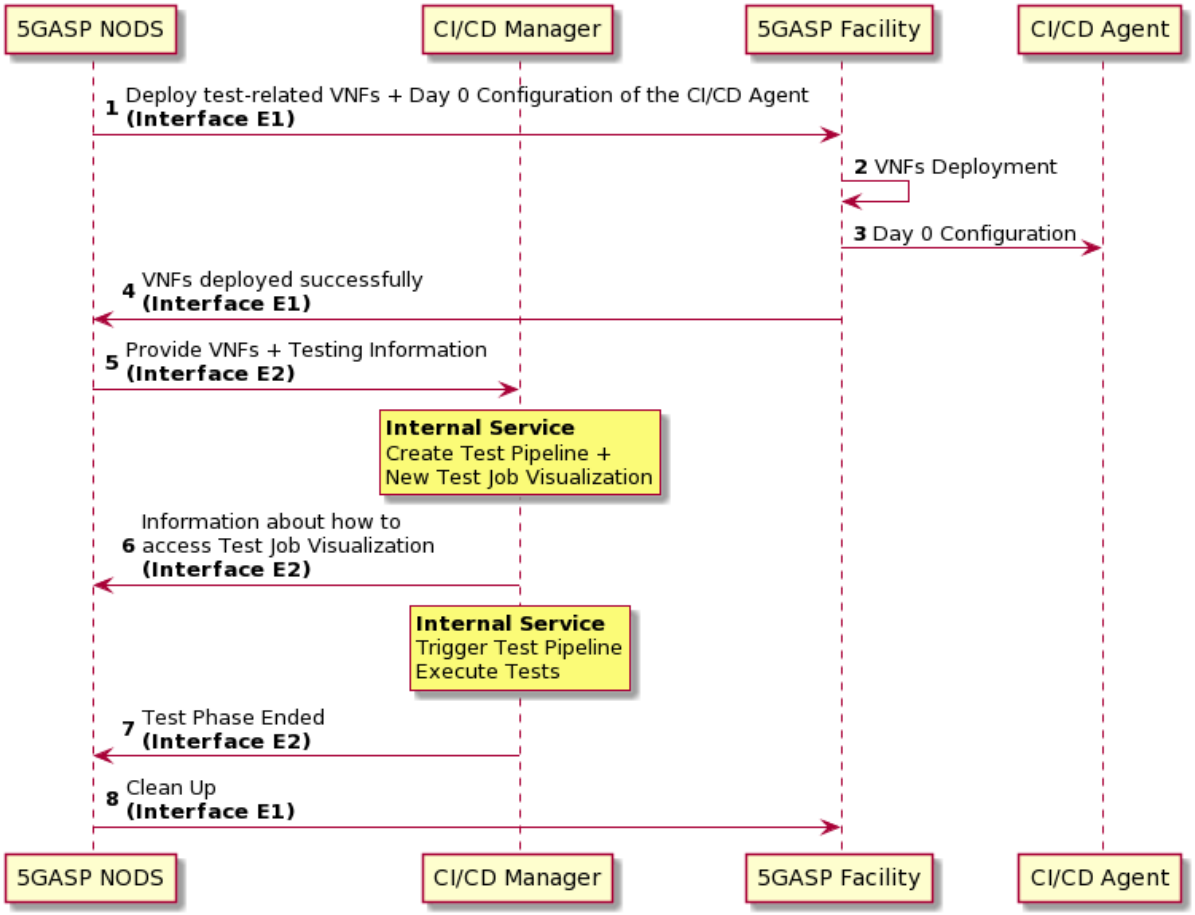


Figure 20. Testing phase interaction diagram

The above figure (Figure 20) depicts the remaining of the nominal deployment process directing its focus on the testing phase, which completes the deployment circle. After test phase is triggered, 5GASP NODS deploys test related VNFs, i.e. CI/CD Agents along with their initial configuration (step 1-3), based upon equivalent orchestration graphs as the ones already described. As NODS is notified upon the successful deployment of the said CI/CD Agents through Interface E1 (step 4), Interface E2 is then employed and the flow is switched to CI/CD manager (step 5). Here, NODS provides CI/CD Manager with the related deployment information assembled from the previous steps. Following, an internal service of CI/CD manager outlines the test pipeline along with the test job visualization. At this step, visualization information is forwarded to NODS (step 6) simultaneously with testing instantiation and execution. Eventually, as the testing phase ends and NODS is alerted (step 7), a clean-up process is initiated in the involved facilities for any obsolete remaining testing artefact.

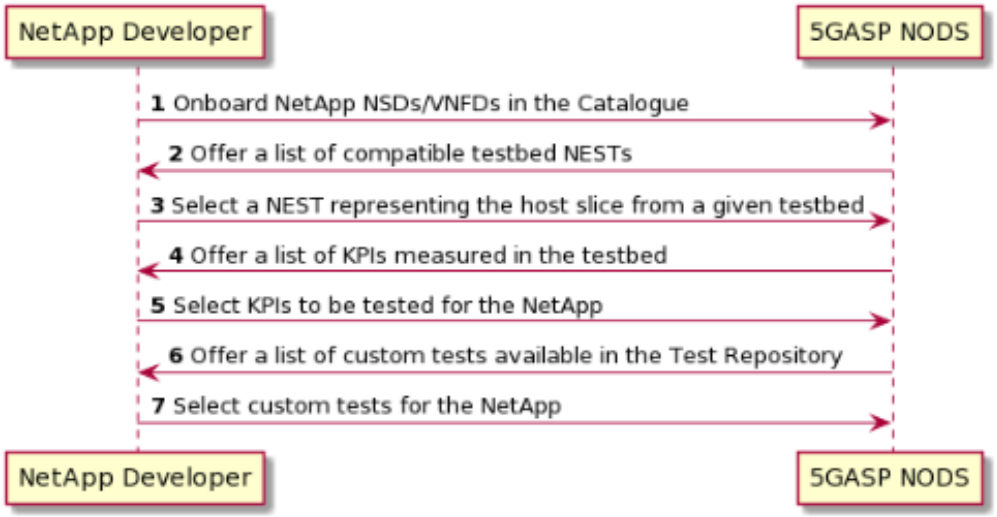


Figure 21. NetApp onboarding interaction diagram

4.1 Interaction with CI/CD service – Interface E2

After the onboarding of the triplet (NetApp Artifact, Test Descriptor, NEST information) on 5GASP NODS, the VNFs of the NetApp will be deployed via OSM as running Network Services (NSs). Alongside this, NODS will drive the initial configurations on the CI/CD Agent deployed in the testbed where the NetApp was deployed.

After the deployment of the NetApp and the initial configurations of the CI/CD Agent, the testing process can start. To do so, NODS must render and extend the Service Test Specification, referring to the corresponding test descriptor, initially provided by the NetApp developer, e.g. adding the assigned IPs of each deployed VNF. For that reason, NODS is expected to query its active service inventory to obtain the required information. At this stage, the convenience of designing NODS to support TMF's Service Inventory [29] is harnessed through the introduction of the Service resource model, which encapsulates all the aforementioned information, as seen in Figure 22.

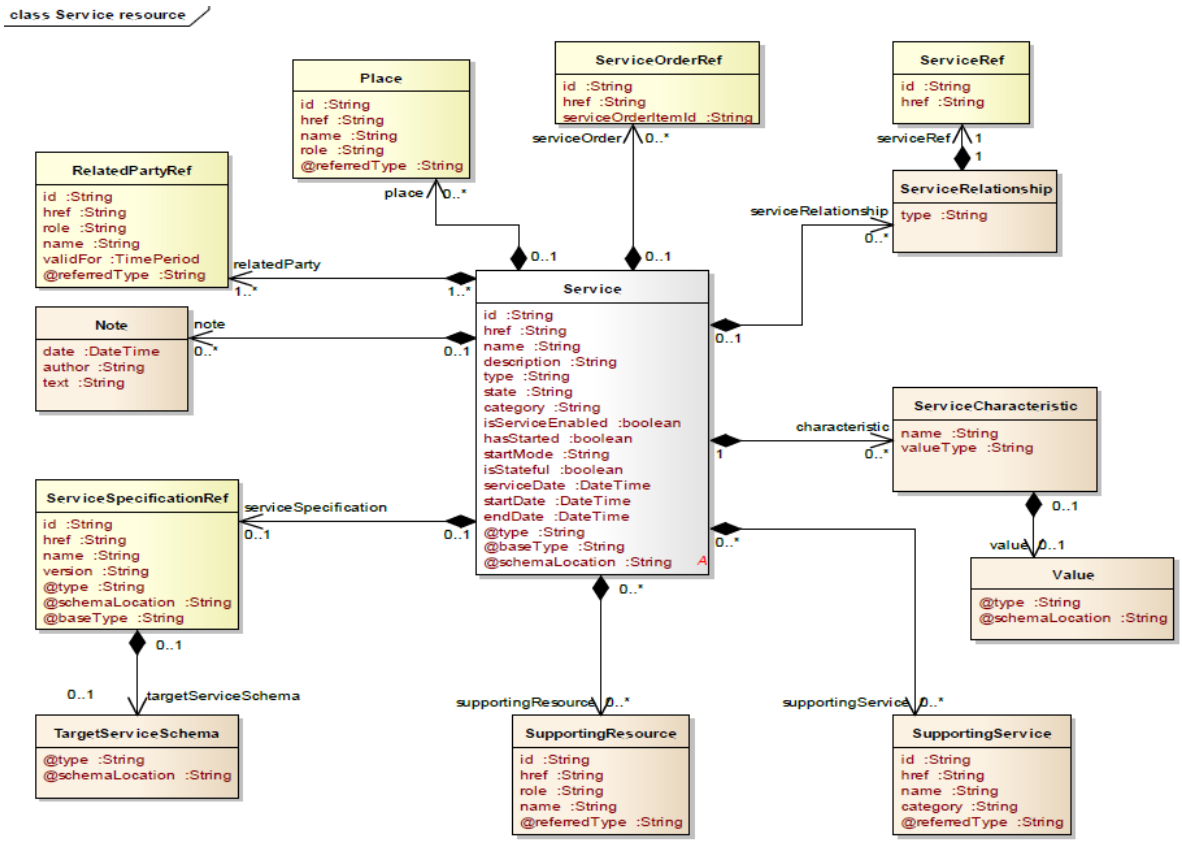


Figure 22. Service Resource model [29]

As the deployment information cannot be known beforehand, i.e. in the test descriptor design and development phase, the Service Test Specification will contain empty key-value pairs of the required data (IPs, VNF IDs, NS IDs, etc) that will be filled by NODS once the NetApp has been deployed and activated. The values are not directly included in the test descriptor to detach NODS from the model and specification details of test related components.

At this point, NODS will create a TMF Service Test instance from the Service Test Specification, housing the attached test descriptor augmented with deployment data. This object will then be forwarded to the CI/CD Manager. By doing this, NODS triggers a validation process on the CI/CD Manager, which is the entity responsible for all testing processes and the communication with the other CI/CD Service's components.

The CI/CD Manager starts by creating a Jenkins Pipeline Configuration File, based on the received testing descriptor, and submits it to the CI/CD Agent responsible for testing the NetApp. Before this, the CI/CD Agent had already registered itself in the CI/CD Manager. Then, the testing process begins with the execution of several tests on the NetApps and concludes by the collection of the test outputs. This stage is meticulously described in D5.1 and thus, only an abstract outline is offered in this document.

After the testing process is finalized, the CI/CD Manager will inform NODS that the testing and validation process has ended, and the results will be submitted using TMF Service Test entity. Specifically, the developer should be able to access a Service Tests repository, which is updated with test results. The overall procedure is illustrated in Figure 23.

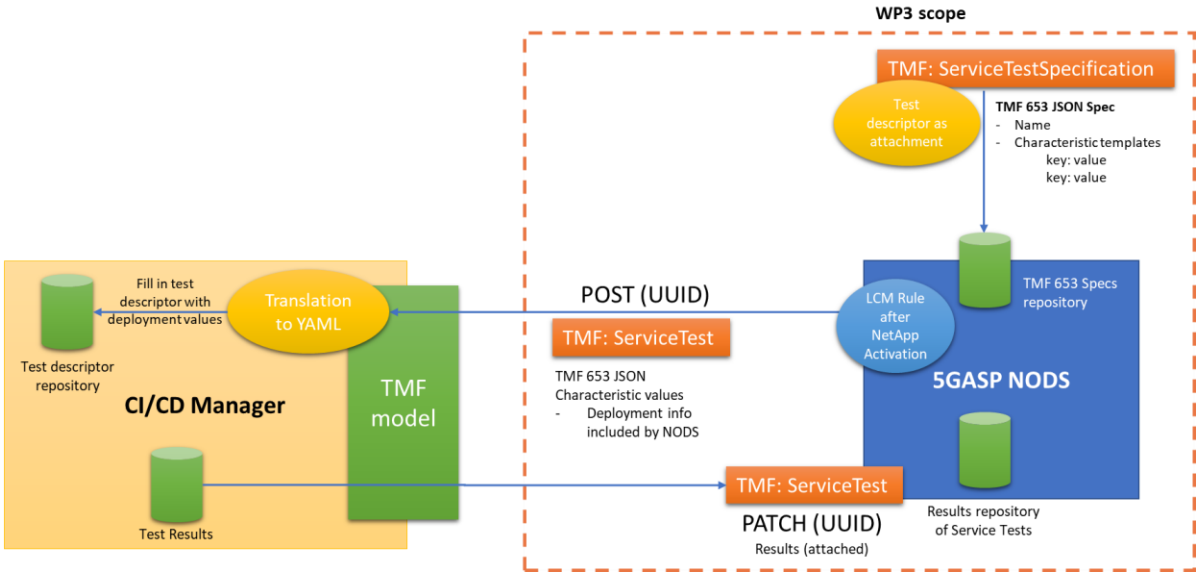


Figure 23. Test descriptor choreography

4.2 Interaction with Facilities – Interface E1/E4/E5/E7

This section covers the southbound interfaces of the portal, introduced in 5.2.5.1 (Interface E1), 5.2.5.4 (Interface E4), 5.2.5.5 (Interface E5) and 5.2.5.7 (Interface E7) subsections of the D2.1.

4.2.1 Interface E1

Taking into consideration the different technological stacks deployed in each facility, and the available standard interfaces, the following segment explores the different options provided for interconnection at the services level between the global portal and the underlying facilities, as depicted in Figure 24.

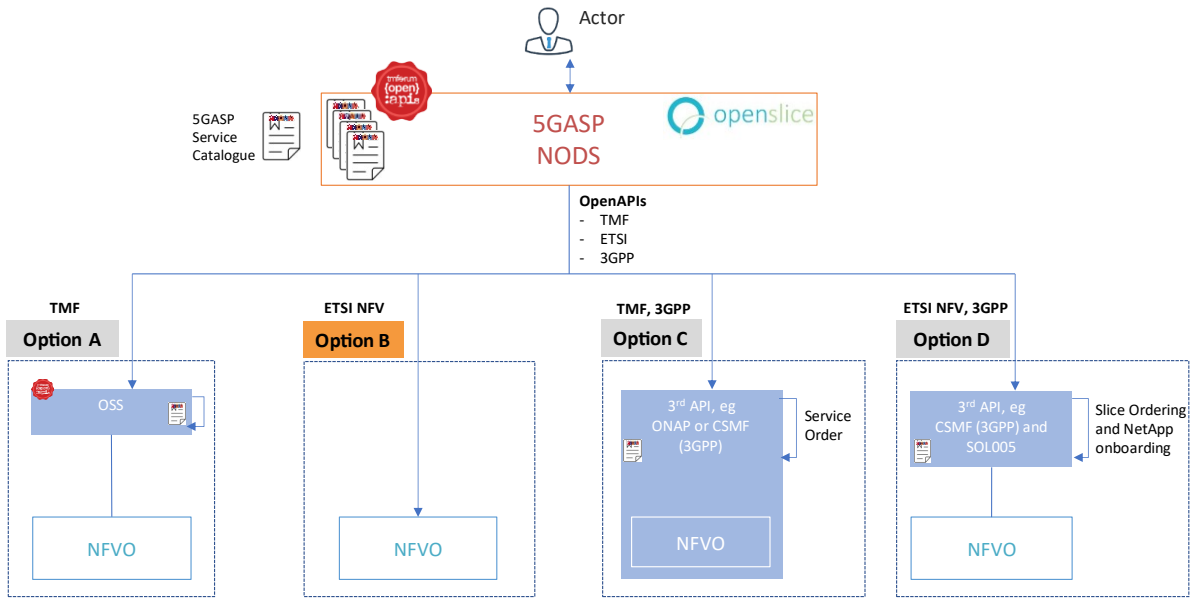


Figure 24. Portal communication with underlying facilities options

Option A: Single portal on each facility site

In option A, each facility site deploys its own Operations Support Systems (OSS) portal solution.

The deployed portal must implement TMF Open APIs that enable items exchange with the global portal. Specifically, the exposure of supported Service Specifications, the handling of an incoming Service Order and the display of the Service Inventory are a set of minimum required actions that must be supported in this deployment. To that extend, a local NODS instance can be utilized for that purpose, as it already supports the aforementioned procedures.

In this option, the connectivity with the underlying NFVO is achieved through an ETSI SOL005 client.

Option B: Global portal as an E2E Service Orchestrator

In option B, the global NODS portal is used as an E2E Service Orchestrator solution.

This comes across as the simplest option to start with, thus nominating it as the appointed solution in the early stages of the project. Each facility site must implement an ETSI-NFV compliant MANO stack (e.g. OSM) and ensure its exposure to the global portal.

The global portal acts as an E2E Service Orchestrator and is responsive for orchestrating and fulfilling the E2E service.

Option C: Facility exposes non ETSI-NFV APIs

Option C enables facilities to implement their own comprehensive platform for orchestration, management and automation of network services (e.g. ONAP).

The facility's implementation should support TMF and/or 3GPP APIs to communicate with NODS. Facility captures the referenced Service Order from NODS, processes it internally as a newly placed order and hands it over to its NFVO. In cross-domain use cases, the orchestration can i) either be supervised hierarchically by NODS, expecting a fulfilled Service Order from the underlying facility so as to proceed with the orchestration, or ii) it can be completely handed over to the facility, should a mesh connection be supported with other facilities.

Option D: Facility exposes ETSI-NFV or 3GPP compliant APIs over other management service

In option D, the global NODS portal would be able to interact through ETSI-NFV and/or 3GPP compliant APIs with the underlying management service.

The definition of a management service can either i) follow the 3GPP concept of 5G network slicing management framework which is built upon standardized management interfaces of network slicing/function management services [30], or ii) any other ETSI-NFV compliant architecture providing NetApp onboarding and slice management.

Again, in this option, the connectivity with the underlying NFVO is achieved through an ETSI SOL005 client.

4.2.2 Interface E4

The proposed interconnection implementation follows the architecture presented in Figure 25. The NetOr service will act as the Communications Service Management Function (CSMF) [22] and will communicate with lower-level Network Slice Management Functions (NSMFs) [22]. The NetOr is then responsible for creating and configuring a Virtual Private Network (VPN) tunnel between the independent administrative domains, creating a secure communication channel.

To successfully achieve the multi-domain scenarios, not only do the network resources need to be aligned with the mechanism, but a centralized service orchestrating agent should also exist, ideally outside all domains in question, to allow the connection of the tunnel peers. That agent will receive and process the dynamic tunnel endpoints' information and exchange it with the remaining peers, effectively completing the tunnel configuration. NetOr is the service serving as that centralized service orchestrating agent, gathering and redistributing the inter-domain information.

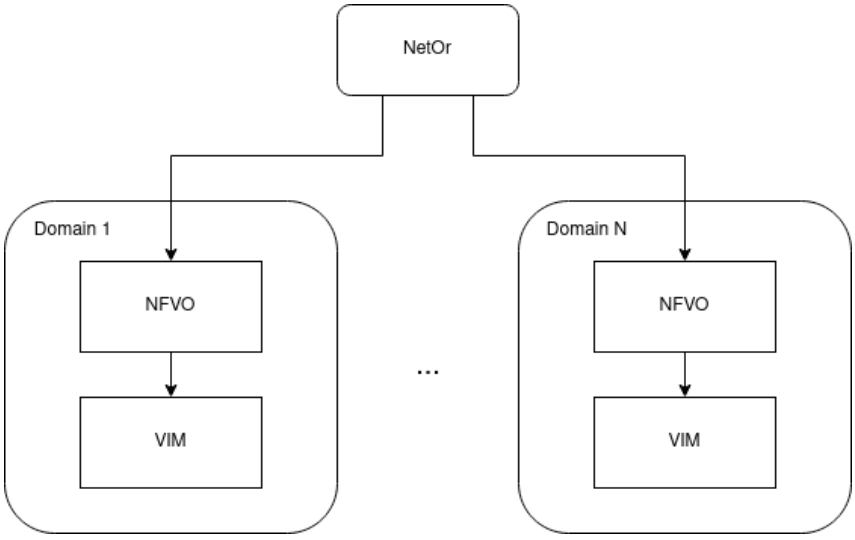


Figure 25. Proposed interconnection architecture

As presented in Figure 26, the process starts by instantiating a final Service based on blueprints, descriptors, and templates that support and activate the multi-domain mechanism. With that request, the system creates all necessary management entities and instantiates the network resources (NSIs or NSs) in the respective domains. After instantiating the Service, there is continuous polling over the status and information of its components. When the NetOr system verifies that a Service component is "running" (meaning that it is deployed and configured), it triggers a runtime operation to fetch the tunnel information, such as the IPs of the tunnel peer machine and the self-generated tunnel public key. Once all the Service components are "running" and their tunnel information has been fetched, the NetOr proceeds to exchange that information between peers, effectively providing the necessary data to peers and configuring the tunnel in the process. Only the NetOr knows all tunnel peers since it instantiated them independently on different domains.

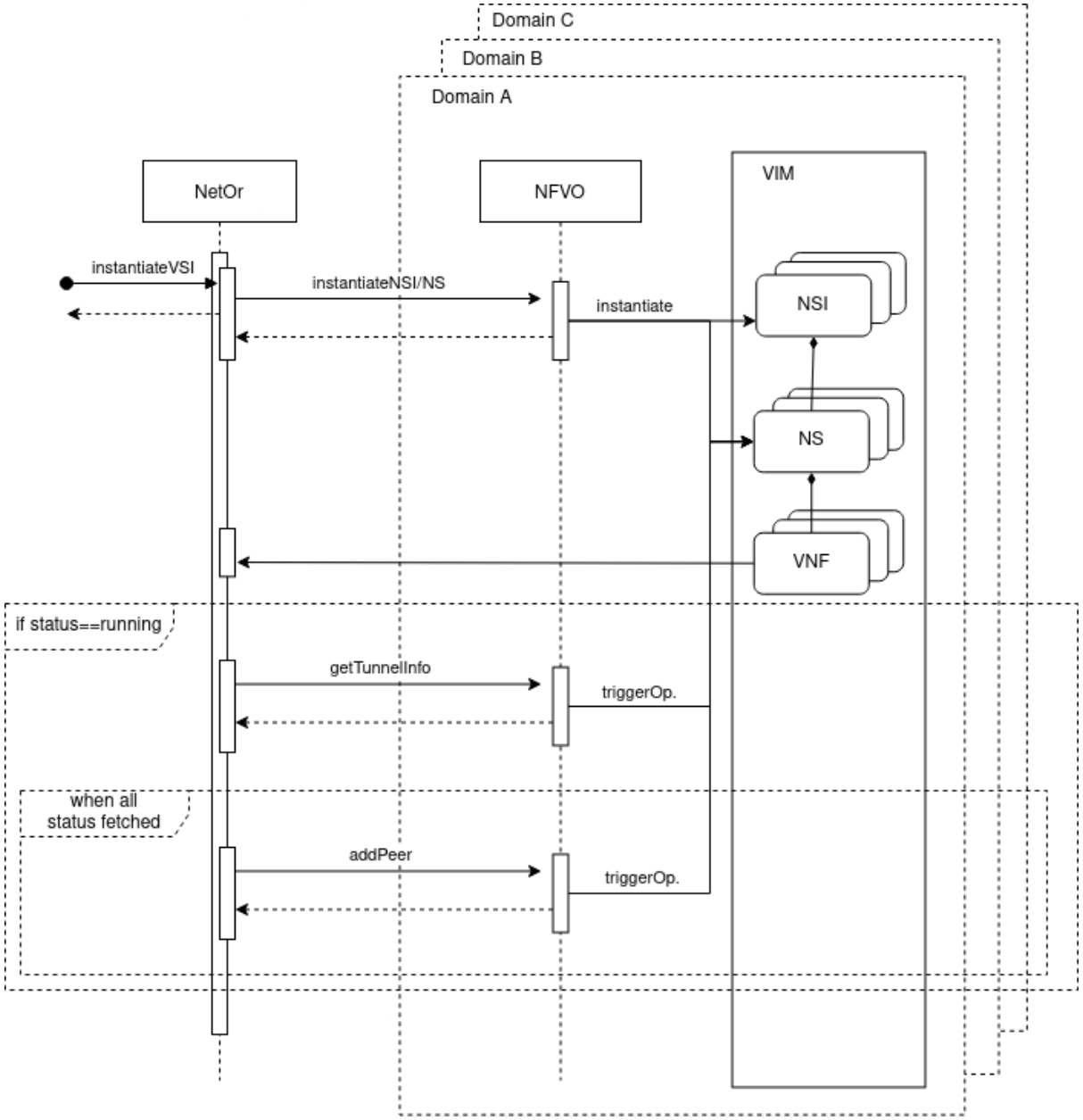


Figure 26. Data flow between NetOr and respective domains

4.2.3 Interface E5

During the first development phase, 5GASP developers will only be able to test their NetApps against some pre-defined tests. Towards later project phases, when custom test execution would be supported as well, the interface E5 shall be employed to onboard the corresponding test descriptors to a target testbed, and specifically to its Local Test Repository (LTR). It should be noted that E5 does not need to maintain a separate interface towards the testbeds, but it can provide its capabilities on top of an already established interface, i.e. E1.

4.2.4 Interface E7

The intended generic multi-domain scenario can be like the one presented in Figure 27, where the intended final service is achieved by an E2E multi-domain network slice, where its subnets

are hosted in distinct administrative domains, interconnected with the aid of a VPN tunnel between said subnets.

This multi-domain mechanism is scalable since it allows the interconnection of as many administrative domains as necessary, needing only to configure the corresponding subnets in the E2E multi-domain Network Slice.

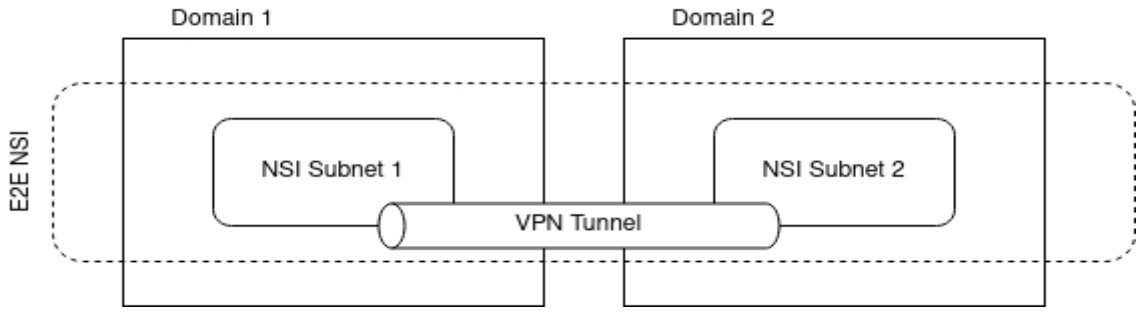


Figure 27. Generic multi-domain scenario

4.3 Interaction with NetApp Marketplace – Interface E3

This interface is utilised for publishing the NetApp to the NetApp Marketplace and therefore making it publicly available, after the DevOps experimentation and certification readiness lifecycle is successfully completed. TMF's Product resource model [31] will be used to accomplish this required interconnection, as it widely observed and employed in telecommunication industry.

Thus, it is expected that the most suitable resource model of the TMF's Product family to describe a NetApp Marketplace asset is Product Offering. Not only it incorporates adequate resources to describe a NetApp offering, but also ensures the interoperability among other industry implementations.

Taking into consideration the various properties of the aforementioned model, the highlighted ones (see Figure 28) will be leveraged to describe a Marketplace asset. Briefly, a Marketplace asset might contain an attachment (e.g. logo, images, certification links or files), topological information about the offered deployment, pricing, asset's specific characteristics, Service Level Agreement (SLA) reference and lastly, a reference to the actual services ordered and employed i.e. hosting network slice, NetApp and test descriptor. Notable mention should be made of the latter entity, namely Service Candidate Ref of the Product Offering resource model. This entity associates the product offering with the Service Specifications constituting the onboarding and deployment model (see section 2.1).

To end up, utilising TMF's Product aims at:

- Consistency between the ordering and deployment model
- Introduction of business aspects, such as pricing, product options, market segment
- Imposing an abstraction layer between customer and service provider
- Effortlessly interacting with other production systems

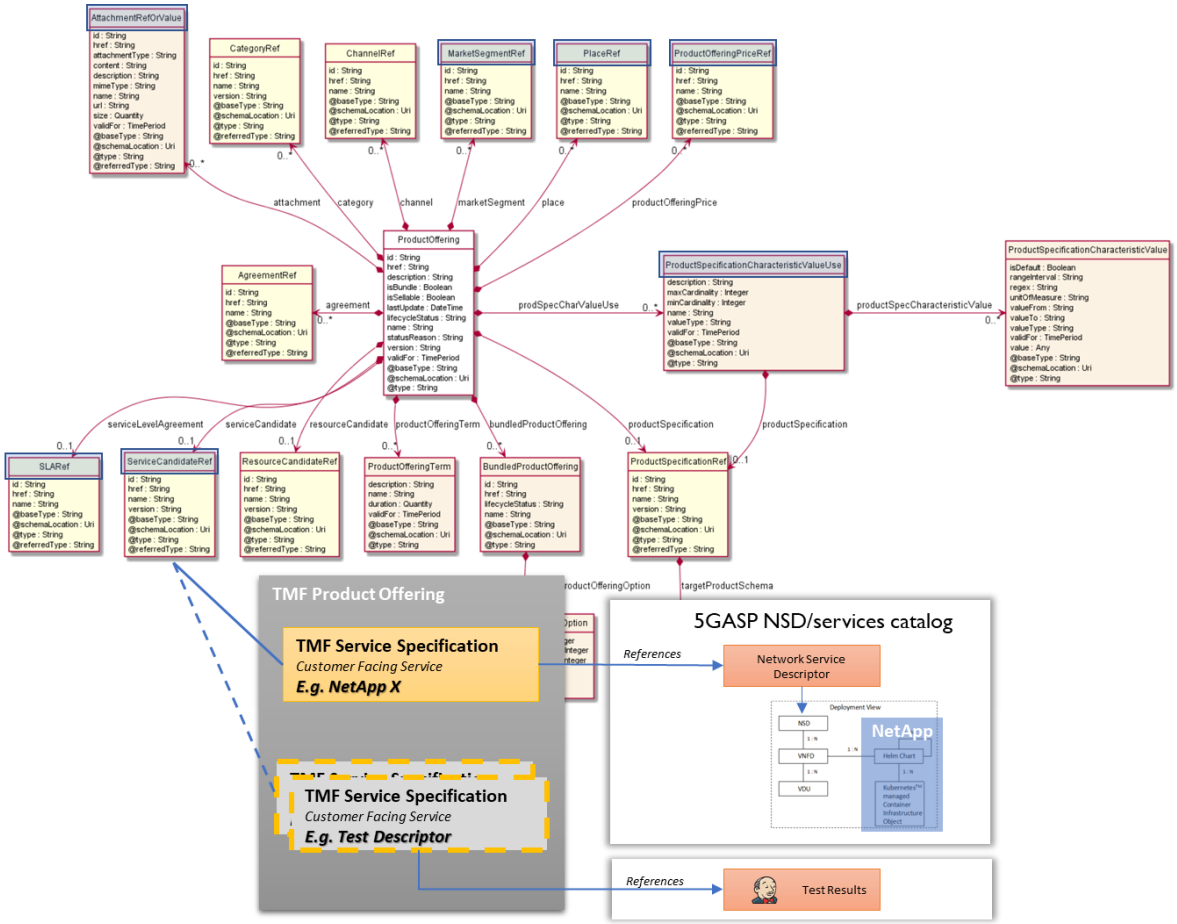


Figure 28. Product Offering resource model

5 5GASP NODS Implementation Release 1

This section is dedicated to implementation aspects based upon the requirements, architecture and interaction of NODS with the overall 5GASP ecosystem introduced in the previous sections. Specifically, it reflects MS3.1's deployment of 1st prototype version of experimentation services for architecture validation.

To begin with, the capability to manage underlying testbeds of the overall facility is being offered, only available only to NODS Platform Administrator. The management involves the employment of the exposed testbed's services API to add, edit or delete a testbed from the multi-domain fabric (Figure 29).

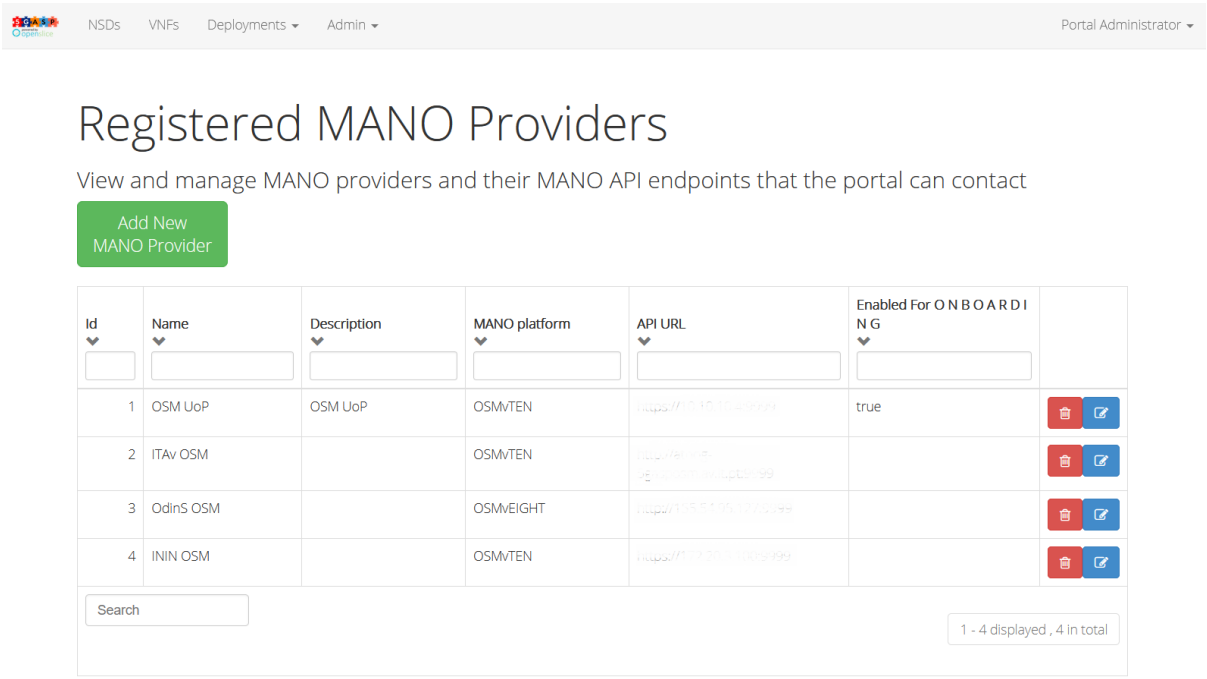


Figure 29. Testbed management UI

Users, i.e. NetApp/NF Developers, are expected to onboard VNF/NSD artefacts and archives. For that reason, a dedicated interface is employed incorporating information as packaging format (e.g. OSM v.10), category filtering and an archive uploading form. The uploading form, as well as the listing of the onboarding archives is illustrated in Figure 30 and Figure 31, respectively.

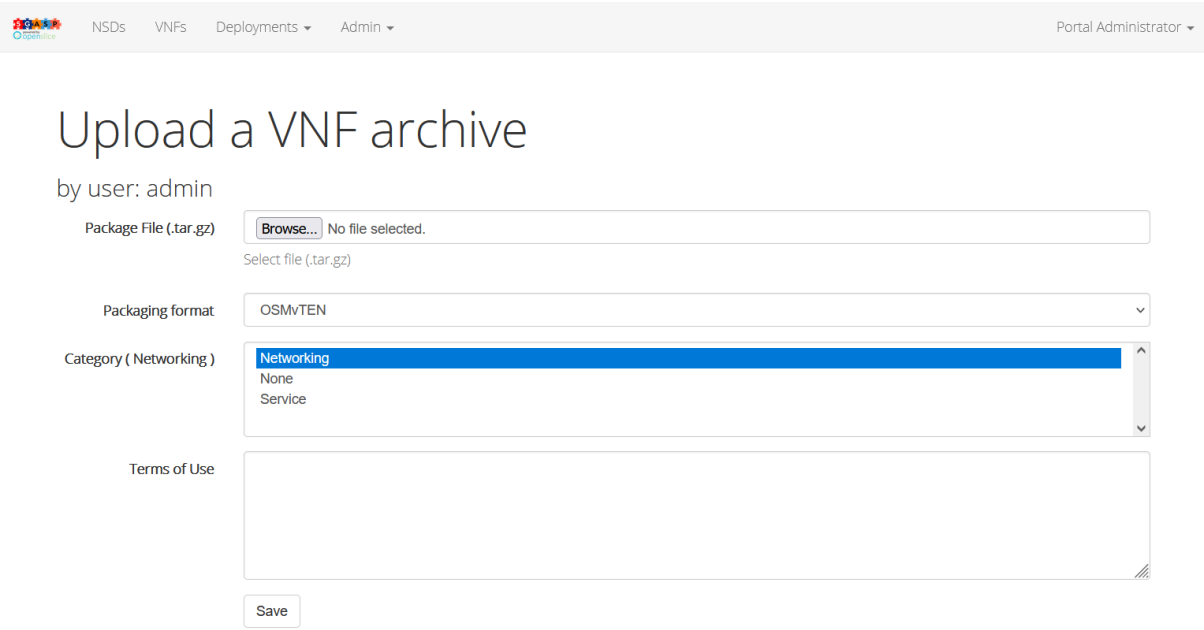


Figure 30. VNF/NSD archives uploading UI

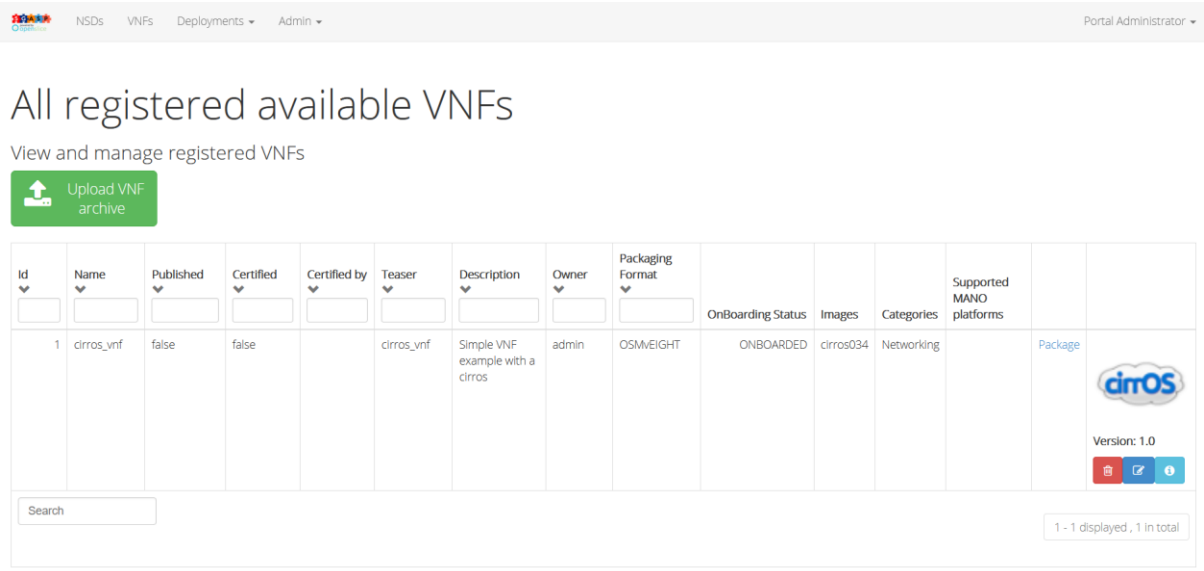


Figure 31. Onboarded VNF/NSD archives listing UI

Once an archive is successfully onboarded, an automated procedure which constructs RFSSs of NetApps out of the onboarded packages is instantiated. At this point, the nominal use case would require the Service Designer to outline CFSSs out of the constructed RFSSs (Figure 32), define their unique characteristics (Figure 33) and eventually publish them to publicly available catalogues (Figure 34).

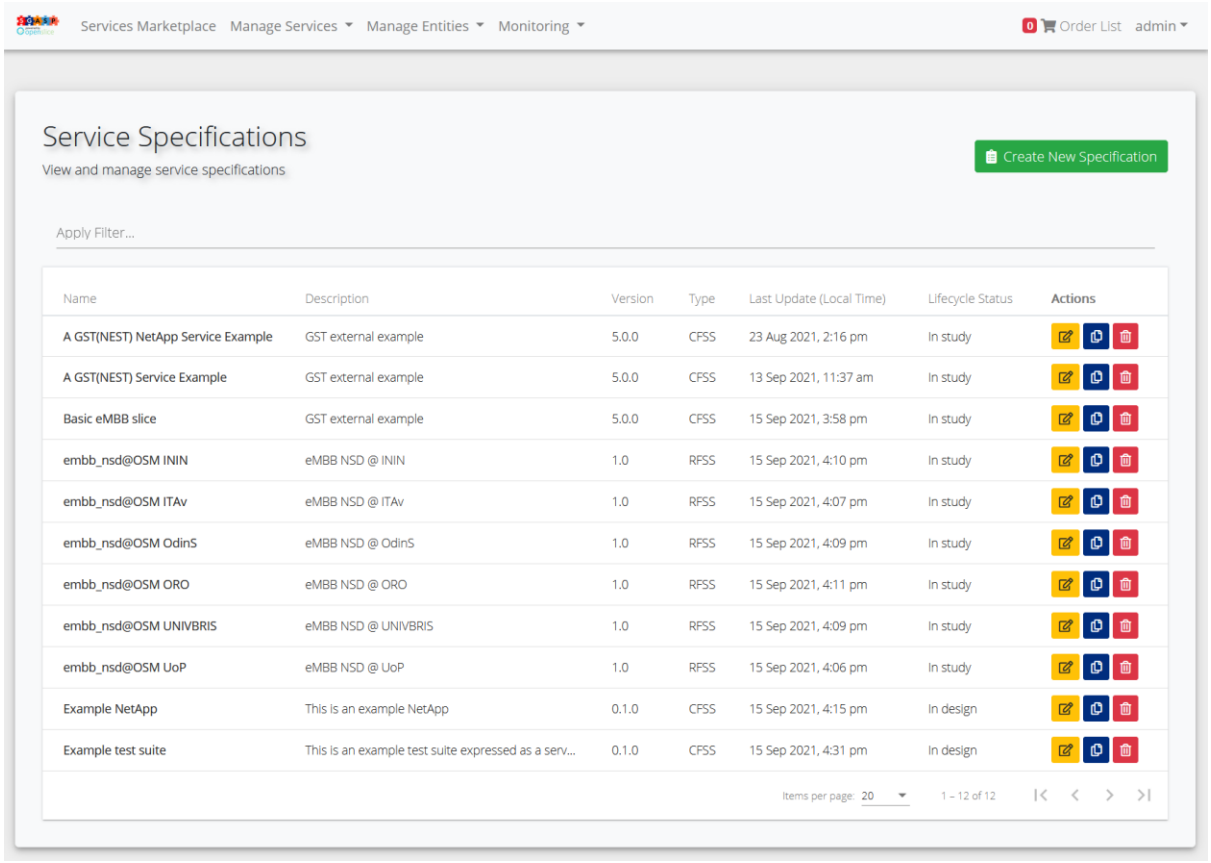


Figure 32. Service Specifications listing UI

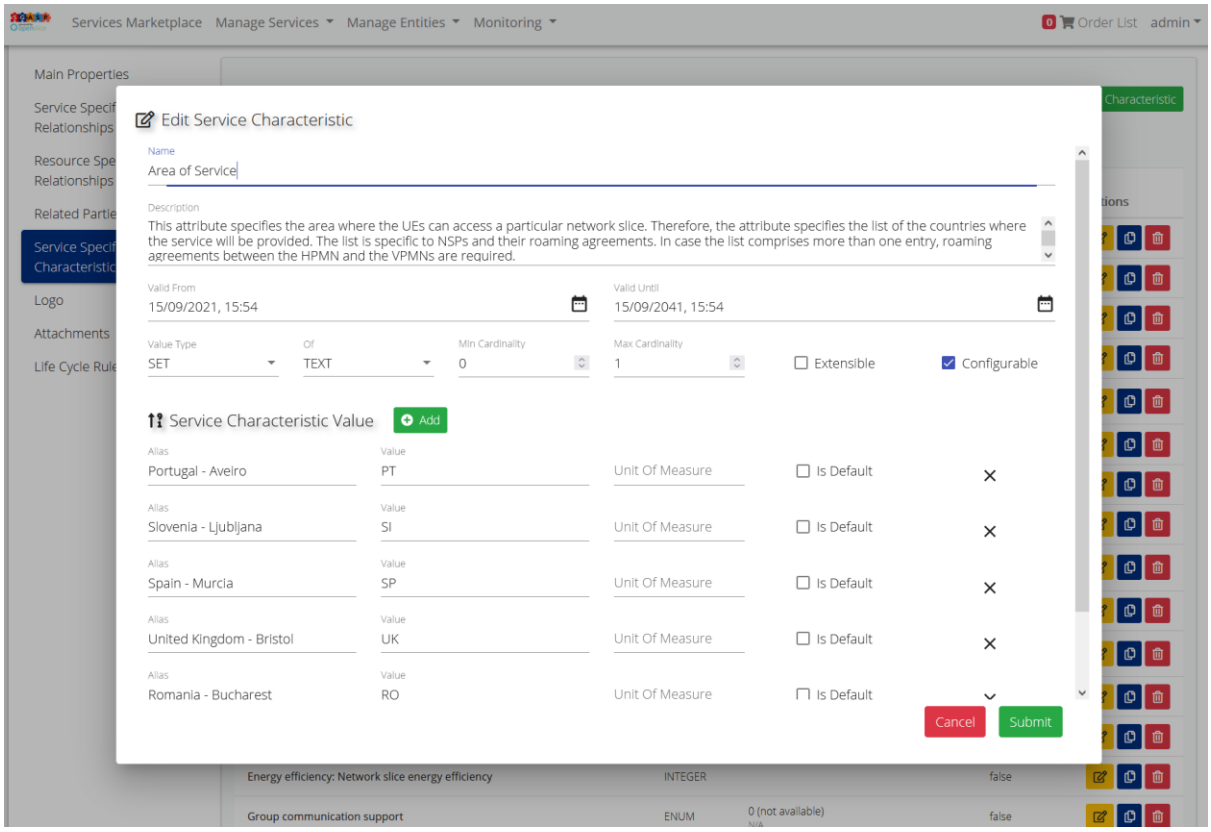


Figure 33. Edit Service Specification Characteristic UI

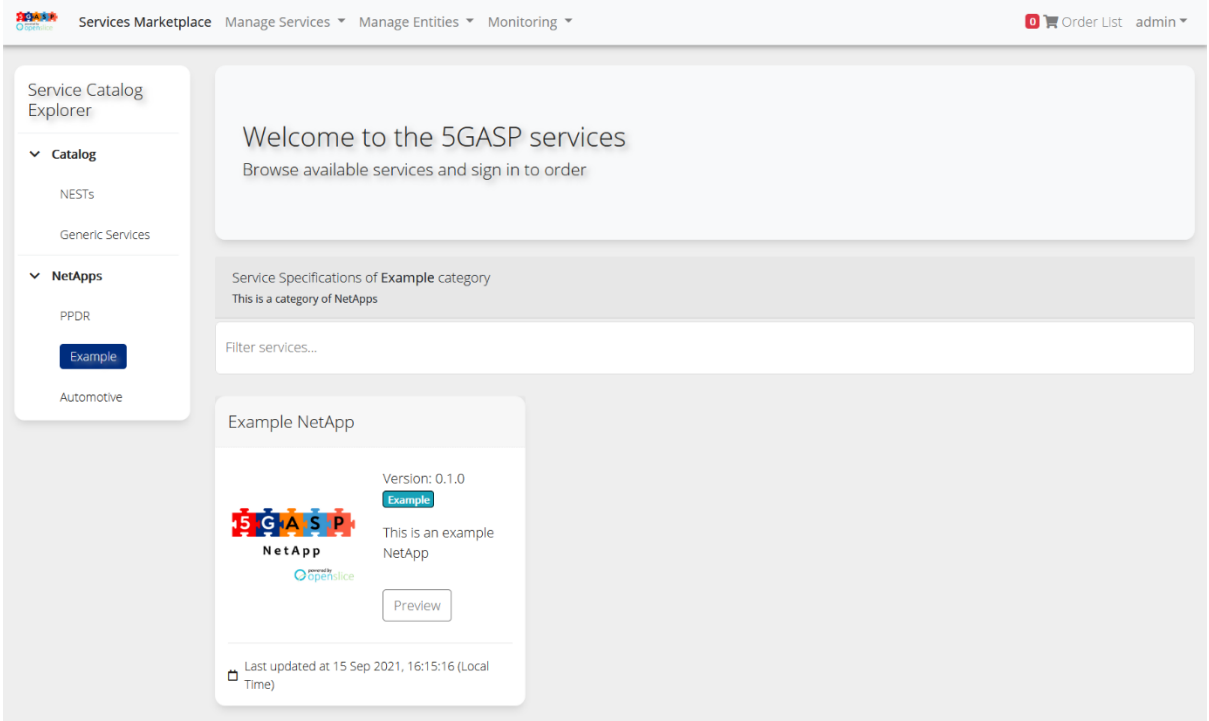


Figure 34. NetApp Catalogue browsing UI

While browsing the public catalogues, the portal user, i.e. NetApp Developer, is able to select a specific CFSS to further examine and conceivably place it in an order list, resembling that of an online marketplace, as depicted in Figure 35.

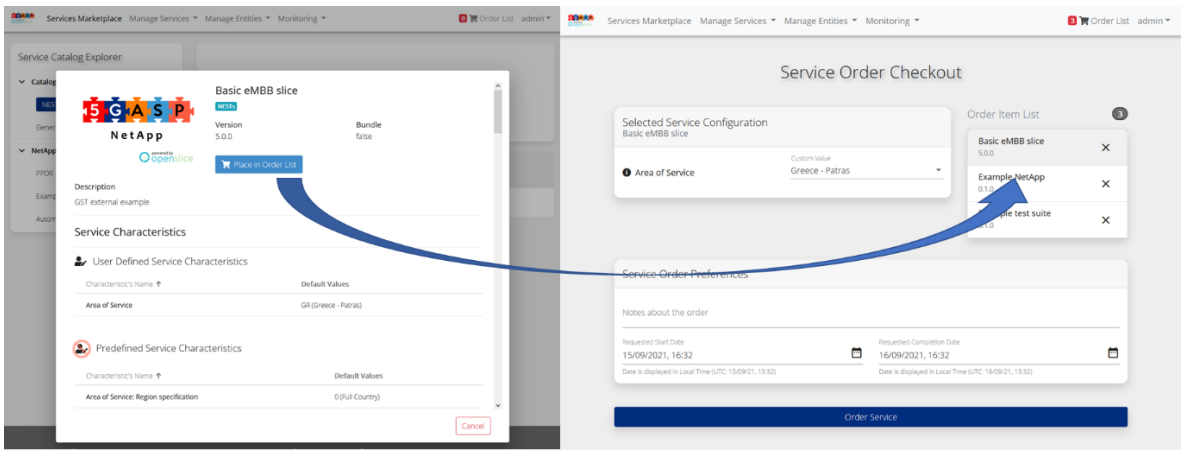


Figure 35. Service Order issuing UI

Consequently, as a Service Order is issued and captured by NODS, the fulfilment process is initiated as described in section 4. The overall fulfilment process is visualised through a specifically designed interface for that purpose, as illustrated in Figure 36.

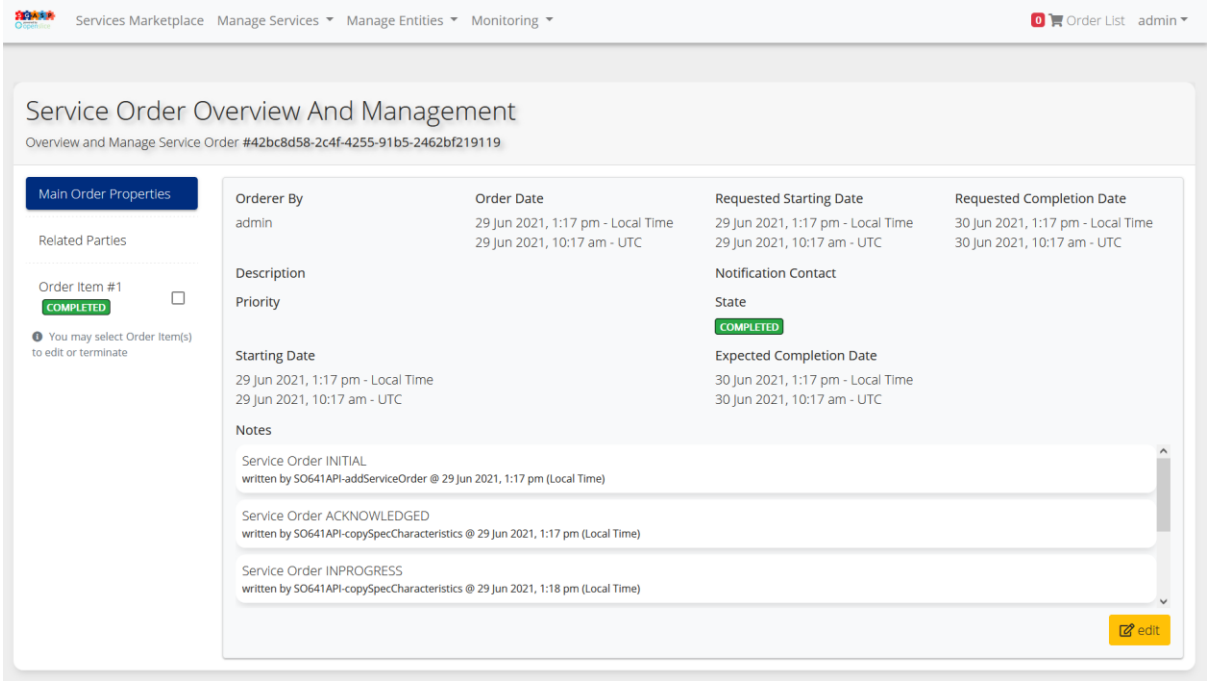


Figure 36. Service Order overview and management UI

It has been made evident, in Section 3.2.8, that dynamic orchestration patterns need to be supported for the purposes of 5GASP. This is achieved by incorporating LCM rules, that execute a specific logic throughout several orchestration phases. Thus, a unique UI is introduced enabling an expert user to sketch on-demand orchestration graphs by employing interlocking blocks. These graphs generate syntactically meaningful code that is injected in the orchestration pipeline. This approach aims to introduce an abstraction layer above the orchestration pipeline and, although it is addressed to an expert user, it does not require direct coding expertise. Such an example can be seen in Figure 37.

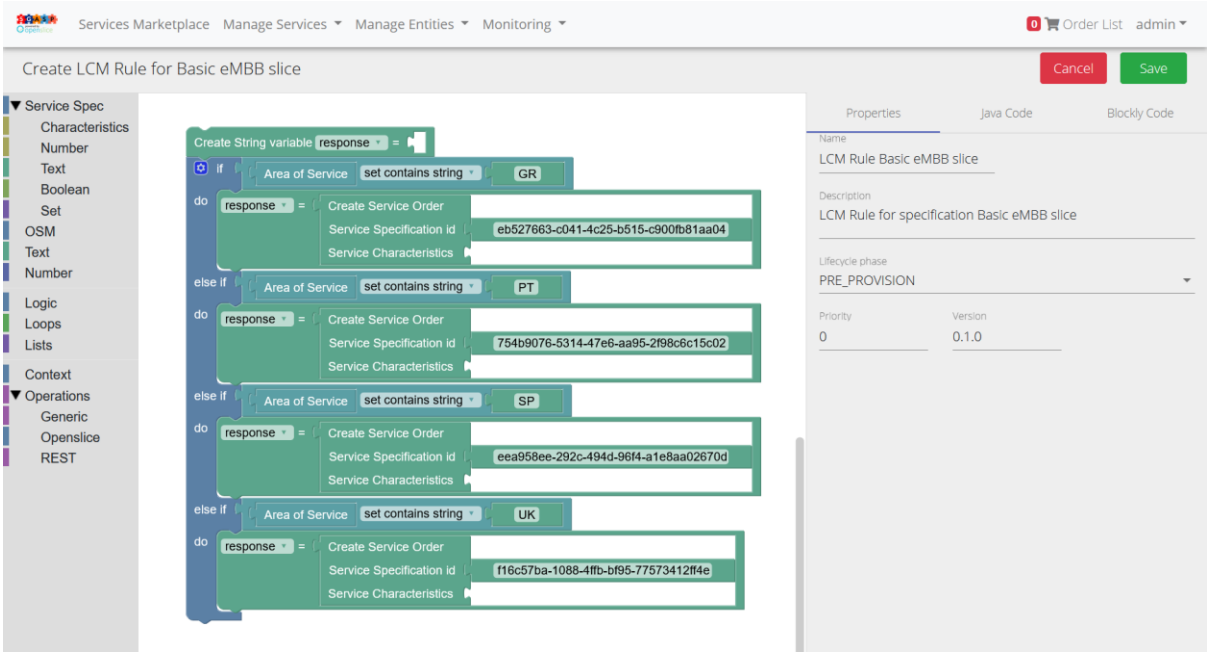


Figure 37. LCM rule designing UI

To end up, an overview of the overall system is provided through the health check service which illustrates the current status of the E2E infrastructure and its interconnectivity links (Figure 38). Additionally, in case of any malfunction detected, a corresponding alarm is raised. Therefore, the Platform Administrator has access to an alarm management interface (Figure 39), where alarms are accumulated and can be handled accordingly.

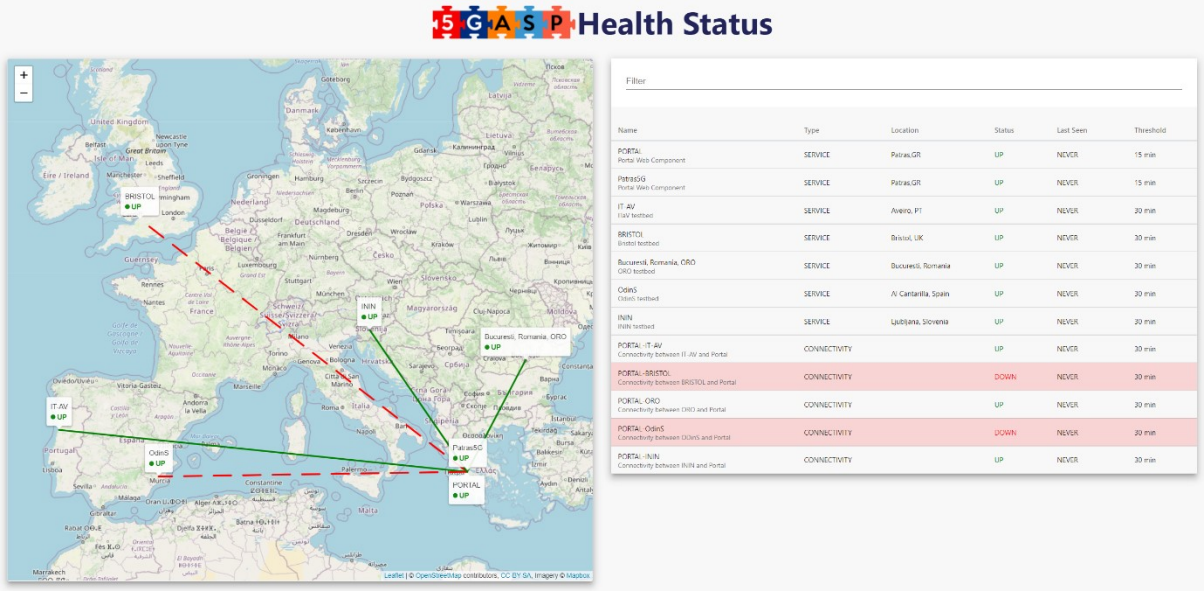


Figure 38. Health Check service UI

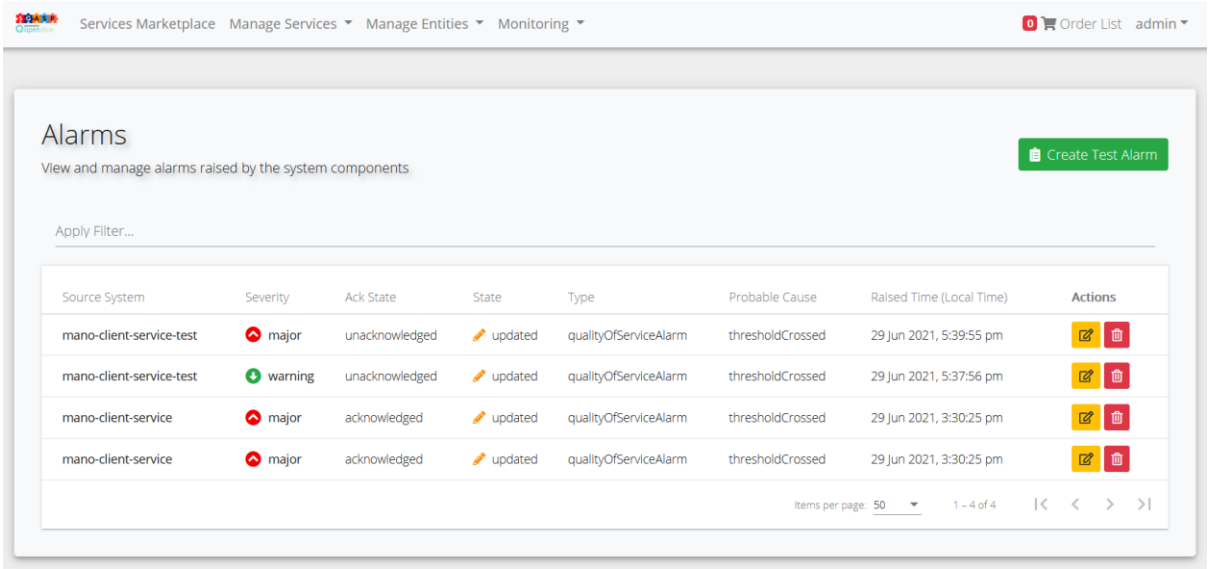


Figure 39. Alarm management UI

6 5GASP multi-domain NFV fabric

This section contains an introduction to the requirements for addressing the facility sites interconnection solution, which shall lay the groundwork for establishing the multi-domain NFV/SDN fabric. Based on these requirements, several technologies are discussed along with how they can be incorporated into an interconnection paradigm, simultaneously focusing on emerged security aspects among others.

6.1 Requirements

To achieve the multi-domain fabric, each domain should have some considerations. Each administrative domain should have an NFVO and a Virtual Infrastructure Manager (VIM) to manage the infrastructure resources and orchestrate new services. A requirement is that those domains' NFVOs should be publicly available, allowing external agents to interact with them. With that in mind, the NetOr can interact with the different administrative NFVOs, configuring and defining the VPN tunnels.

Additionally, each domain should allow each deployed service to have at least one public access point. That point, in the inter-domain scenario, is used as the domain's tunnel endpoint.

Specifically, the following requirements along with explanation, should be expected if we assume two sites to have support over a use-case requiring multi-domain connectivity:

- **L3 Interconnectivity between sites**: This can be done via Public Internet or a private L3 network (e.g. MPLS), where the former is more accessible to a wide variety of partners. Moreover, at least one of the sites needs to have an accessible IP address reachable from the other sites.
- **VPN connectivity**: Each site needs to have a mechanism such as the Gateway VNF described in Section 6.3, which assists in deploying a site-to-site tunnel. This VNF can have a technology such as Wireguard [32] or OpenVPN [33].
- **Virtual network segmentation**: Each site must have some technology to enable network virtualization to isolate use-cases, such as VXLAN [34]. For example, such a technology is part of OpenStack Neutron [35] by default.
- **Virtual resource isolation**: This means having dedicated vCPUs and RAM for network functions. This is ensured by having a dedicated cloud computing solution at each site for e.g., based on OpenStack, which ensures resource isolation for VNFs.
- **High availability**: This requires multiple solutions on both compute and network level. For compute, a combination of load balancer and multiple instances of VNFs can solve the problem, which is available both at OpenStack [36] and Kubernetes [37]. For a multi-domain use-case, the high availability for network requires redundant connections between sites. These could be either multiple private L3 networks such as MPLS, or a combination of such a network with public internet.
- **Network Quality of Service (QoS) requirements**: These include some strict latency and throughput guarantees. For throughput, this requires having secured upstream bandwidth from the Internet Service Provider to guarantee a minimum/maximum throughput for the use-case, however public internet does not ensure guaranteed

throughput. For this reason, private MPLS networks which have Traffic Engineering (TE) capabilities needs to be used. A good solution is to use Software Defined WAN (SD-WAN), which can track the bandwidth utilization of multiple connections: Internet or MPLS at the same time and can switch-over the tunnel connectivity from one mechanism to another based on the current bandwidth requirements. Such an SD-WAN solution can be co-located with the NetOr solution proposed to be used in 5GASP.

For strict latency requirements, the physical as well as the network distance between the two sites should be an input to determine the acceptable latency i.e., the sites need to be close to each other to support the maximum allowable latency threshold. Here as well, the public internet does not guarantee a maximum latency threshold, and a private MPLS network is more recommended, however SD-WAN solution can also optimize the connection in this case. For more fine-grained control, the Gateway VNFs at each end may add DiffServ QoS fields in the tunnel packets, such that they are prioritized over other packets on the internet or MPLS network. In this instance, MPLS TE solution can enable using shorter and less congested paths in the network to ensure required latency.

6.2 Technologies

Some options for the interconnection technologies used in the multi-domain scenario are:

- **OpenVPN** is a popular and highly secure protocol vastly used. It runs on either the TCP or UDP transport protocol of TCP/IP stack, guaranteeing the assured order delivery of information or focusing on faster speeds, respectively. It has many benefits, such as being open-source, versatile, extremely secure, and facilitates bypassing firewall issues when needed. Unfortunately, it may demand a complex setup due to its vast choice options and intricate mechanisms. It may not be the best option when speed is the priority, but it is one of the best when needing top-notch security.
- **IPSec/IKEv2** [38] sets the foundation for a secure VPN connection by establishing an authenticated and encrypted connection. It was developed by Microsoft and Cisco to be fast, stable, and secure. It succeeds on all these fronts, but where it shines is its stability. As part of the IPSec internet security toolbox, IKEv2 uses other IPSec tools to provide comprehensive VPN coverage. Its most notable benefits are the robust stability achieved with the usage of IPSec tools, flexible security, and speed provided by the VPN protocol. On the other hand, it has limited compatibility, since the protocol does not support some operations systems, such as Linux-based ones. It may be one of the best options when needing a stable VPN connection that persists when switching networks.
- **L2L IPSec Tunnel** is a standardized VPN technology (preferred by Operators) for connecting securely distant locations over unsecured internet using **IPSec/IKEv2** described previously, by running two IPSec VPN phases. Phase 1 for negotiation and configuration of the policy establishment towards the secure channel for the future communication, defining the authentication methods and security protocols engaged. Phase 2 for crypto traffic management using software configuration to perform data flow selection for security process and to define the flows policy and peers to the traffic flows.

- **GREoverIPSec** [39]. IPSec/IKEv2 suite can also be combined with Generic Routing Encapsulation (GRE) Tunnels for dynamic route advertisement in case if dynamic encryptions domains are required. Both unicast and multicast-aware routing protocols can be used over GRE tunnels (e.g. BGP, OSPF, etc.).
- **Wireguard** is the newest and fastest known tunneling protocol. It uses state-of-the-art cryptography that already presented in other VPN options, such as OpenVPN and IPSec/IKEv2. However, it's still considered experimental and has some vulnerabilities that need to be fixed. Its benefits include being free, open-source, modern, and extremely fast and lean. Its disadvantage is its incompleteness, still having room for improvements. Wireguard is possibly the best option when speed is paramount, such as streaming, online gaming, or downloading large files scenarios.
- **Secure Socket Tunnelling Protocol (SSTP)** [40] is a fairly secure and capable VPN protocol. Despite being a primarily Microsoft product, SSTP is available on other systems besides Windows. Its advantages are being a Microsoft product, which for Windows users this protocol is supported or even built-in, its security by supporting state of the art algorithms and the ease of bypassing firewalls. Its disadvantage is, also being a Microsoft product, that it is not an open-source product and that it lacks support for other operating systems, such as Linux-based ones. SSTP proves to be a good option for bypassing geo-restrictions and enhancing privacy while browsing the internet.

6.3 Interconnection example

In this section, we present an example of how a NetApp deployed over multiple 5GASP facilities may be interconnected. We assume a NetApp, which has a VNF hosted in the University of Bristol UK, 5GASP facility and another VNF hosted in Instituto de Telecomunicações Aveiro Portugal (ITAv) 5GASP facility. These facilities are connected to the public Internet as shown in Figure 40.

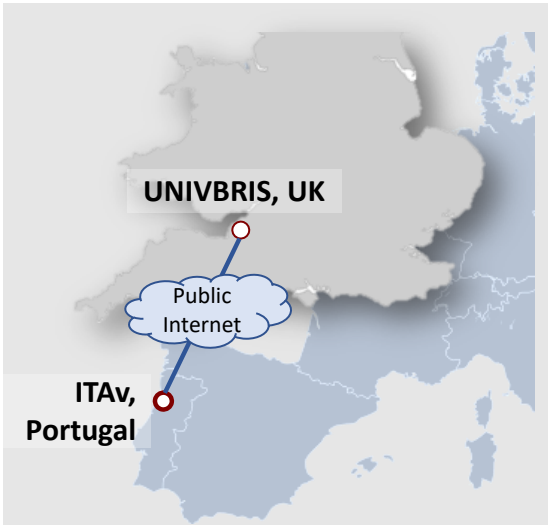


Figure 40. University of Bristol and IT Aveiro sites interconnected via public internet

For simplicity, we consider that both facilities have identical setup as shown in Figure 41:

- OSM as the NFV orchestrator to manage the NS/VNF lifecycle and operation via a VIM,
- OpenStack as the VIM to manage the compute resources,
- An internal network to interconnect the compute servers and provide further external connectivity to the public Internet.

When a NetApp developer deploys the NetApp, they may specify the location of the VNFs as the two 5GASP facilities. This will have some consequences on the deployment phase as the 5GASP global orchestrator needs to refine the NSDs in such way that when the VNFs are deployed in each 5GASP facility, they are interconnected via the public Internet. There are two techniques to have a multi-domain VNF connectivity.

First, a Gateway VNF (G-VNF) at each site is permanently deployed. G-VNF hosts the VPN software, where WireGuard is a suitable option as proposed in D2.1. To create a VPN tunnel between two G-VNF, both G-VNFs need to be connected to a public internet or a private MPLS network, where at least one of the G-VNFs needs to expose an accessible IP address and port pair, which can be used to initiate the tunnel. The G-VNF at each site is also connected to *Network*$_A$ and *Network*$_B$ (see Figure 41), where these networks are deployed either using the Neutron plugin in OpenStack, or some local SDN controller. These networks are either VLAN L2-based provider networks or VXLAN based self-service networks, while observing OpenStack terminology. The NetApp VNFs are connected to the G-VNF via *Network*$_A$ and *Network*$_B$. In VXLAN based self-service networks, Neutron project deploys VXLAN tunnels between the G-VNF and NetApp VNFs, if they are deployed on different compute servers. These networks must be specified at the NSD level before deployment. Once the VNFs are deployed, the NetOr, which essentially acts as a broker, performs the tunnel creation procedure between the G-VNFs which was briefly explained in D2.1. This includes fetching the tunnel information, including the public IP address of the G-VNFs and the tunnel public key. Using this information, the NetOr proceeds to exchange the connection information between the G-VNFs (i.e. the peers), to configure the tunnel between the G-VNFs. This allows the NetApp VNFs to be interconnected transparently at Layer 2.

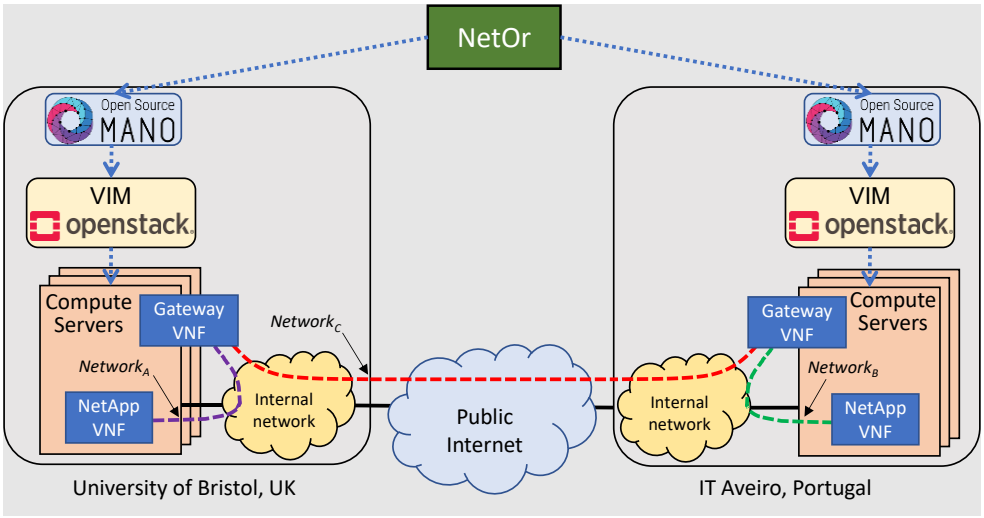


Figure 41. Multi-domain VNF connectivity facilitated by NetOr

Second, the G-VNF may be deployed dynamically as part of the NS. This requires them to be specified in the NSD and connected to the public internet at the deployment phase. Consequently, each site needs to expose an OpenStack network such as $Network_C$ connected to the public internet, which is connected to the G-VNF at deployment phase. After the G-VNFs are deployed, the procedure of establishing the tunnel is same as the step described previously using the NetOr.

6.4 Interconnection security aspects

As discussed, communication among infrastructure facilities and cloud-native functions and virtual workloads tuning inside will require connectivity to interfaces for data-plane, as well as for the CI/CD orchestration pipeline. As facilities are already exposing connectivity options such as OpenVPN, IPsec, Wireguard, and (Secure Shell Protocol) SSH tunnels, the common denominator would be a linux-based Wireguard connectivity for IP-level overlay. From a security discussion context, WireGuard is a secure network tunnel that operates at TCP layer 3 and is implemented as a kernel virtual network interface for Linux. It seeks to replace both IPsec and popular user-space or TLS-based alternatives like OpenVPN for most use cases by lowering complexity and increasing performance (particular for multi-core machines). In terms of security properties, this VPN solution uses single round-trip key exchange and all sessions are transparently created to the user; similarly to OpenSSH, it employed pre-shared static keys for mutual authentication and establishes perfect forward secrecy for all sessions. Further attributes of Wireguard include DoS mitigation mechanism and fast authenticated encryption of packets. Non-functional security properties of Wireguard include a smaller footprint (compared to OpenVPN) of its OSS base and has allowed for a successful security audit in 2020.

In light of this, data and control-plane security for the facility interconnect will be handled at a low (IP) level with a tool that is fast becoming an industry staple and provides a robust level of encryption, verifiability and performance. The implementation established in 5GASP will be the point-key share map to enable facilities to setup the overlay network between the nodes. Each node will know the public key and endpoint IP:port tuple of all remotes. As the facility number and setup is fairly static, configuration management in the central 5GASP services can be leveraged and will be used for maintaining this directory.

7 Required implementation capabilities from 5GASP facilities

This section provides the first implementation drafts of facilities towards the establishment of the multi-domain fabric and the provision of hosting network slices in each facility. Also, some envisioned plans are presented with aspects on how to tailor 5GASP facilities to vertical NetApps requirements. The latter are only introduced in the form of proposals, as they refer to an upcoming task (M12). This section's input will be reflected more maturely in future updates of this document.

7.1 Implementation to support the multi-domain NFV fabric

Three main requirements were determined towards the aim to lay the foundation for the multi-domain fabric among the facilities, namely:

- Implementation of E1 interface,
- Implementation of facilities interconnection,
- Implementation for testing enabling.

The scheme to support the above requirements, along with its implementation roadmap are presented in the following subsections.

7.1.1 Implementation of E1 Interface

All facilities have elected Option B, out of all described in section 4.2.1, as the most viable implementation scenario of E1 interface, at least for early project's phase. This scenario involves an integration of an ETSI-NFV compliant MANO stack by each facility and its direct exposure to NODS, which acts as an E2E Service Orchestrator. An interconnectivity map between NODS and its underlying facilities is illustrated in Figure 42.

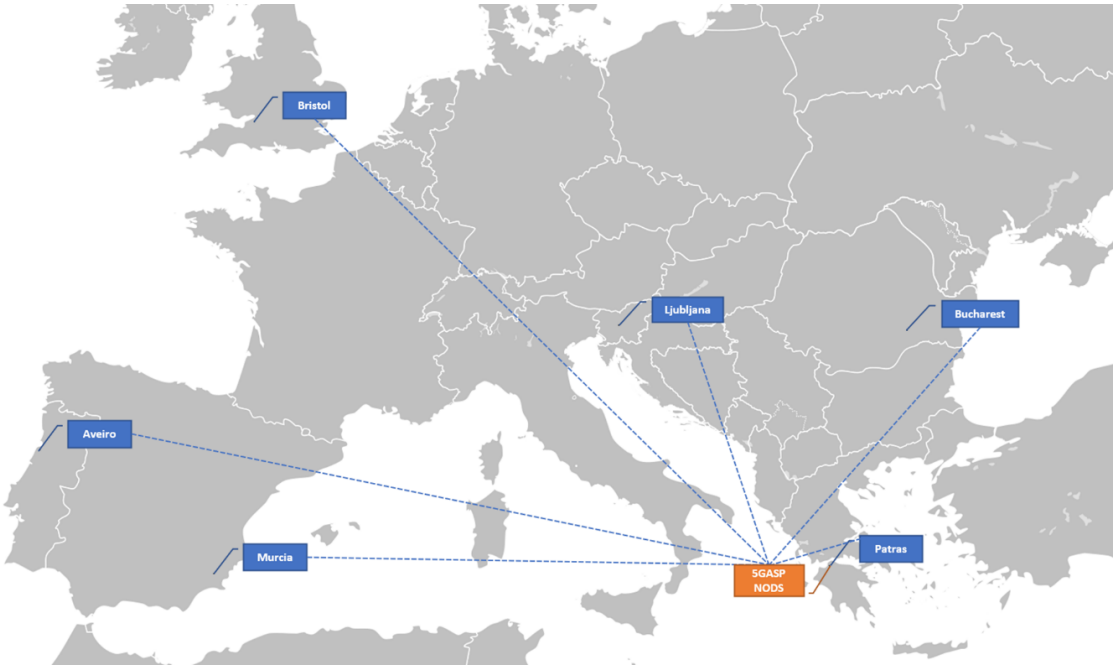


Figure 42. NODS - Facilities interconnectivity map

The following table summarises the exposed service election per facility along with its selected interconnection option with NODS.

Facility	E1 Interface option (See section 4.2.1)	Exposed service (OSS, NFVO, CSMF)	Interconnection option
Aveiro	Option B	NFVO (OSM) Version: 9.1.3 Tenant: 5GASP_ITAv	VPN
Patras	Option B	NFVO (OSM) Version: 10 Tenant: 5GASP_UoP	Internal Network
Bristol	Option B	NFVO (OSM) [1] Version: 10 Tenant: 5GASP_UNIVBRIS	VPN
Bucharest	Option B	NFVO (OSM) [2] Version: 8 Tenant: 5GASP_ORO	VPN (L2L IPSec)
Murcia	Option B	NFVO (OSM) Version: 8 Tenant: 5GASP_Gaia5G	Firewall rule
Ljubljana	Option B	NFVO (OSM) Version: 10 Tenant: 5GASP_ININ	VPN

Table 7. E1 Interface implementation per facility

7.1.2 Implementation of facilities interconnection

As described in the previous sections, 5GASP offers maximum flexibility of interconnection through the dynamic provisioning of tunnels between facilities.

In order for this dynamicity to be achieved, facilities must nonetheless implement some aspects, namely: provisioning of information in NetOr about each facility through a web interface (NFVO credentials of each facility and network addressing), local provisioning of firewall rules (to enable access to the NFVO from NetOr) and/or VPN credentials (through which NetOr can acquire access to the NFVO).

Due to internal security rules, we acknowledge that not all facilities will be able to interconnect with their counterparts. Some facilities have very strict rules about inbound connections defined administratively by superior responsible departments. In cases of two such facilities are required to interconnect, we will resort to transport domain provided by another facility.

Interconnection test will be carried out in the following months to validate NetOr capabilities. We have defined a 2-phase plan:

[1] Bristol facility is expected to finalize the exposure procedure by the end of M10. The version and tenant values are expectations.

[2] Bucharest facility is expected to finalize the exposure procedure by the middle of M10. The version and tenant values are expectations.

- Phase 1: Connecting facilities without connectivity restrictions (by Q4 2021)
- Phase 2: Connecting facilities with inbound connectivity restrictions to 1st phase facilities (by Q1 2022)

In the following table we summarize this plan.

Facility	Aveiro	Patras	Bristol	Bucharest	Murcia	Ljubljana
Aveiro		Phase 1	Phase 2	Phase 1	Phase 1	Phase 1
Patras	Phase 1		Phase 2	Phase 1	Phase 1	Phase 1
Bristol	Phase 2	Phase 2		Phase 2	Phase 2	Phase 2
Bucharest	Phase 1	Phase 1	Phase 2		Phase 1	Phase 1
Murcia	Phase 1	Phase 1	Phase 2	Phase 1		Phase 1
Ljubljana	Phase 1	Phase 1	Phase 2	Phase 1	Phase 1	

Table 8. Facilities interconnection plan

7.1.3 Implementation for testing enabling

The CI/CD Agent is one of the most crucial components of the CI/CD Service since it is the entity that performs the tests on the NetApps and gathers their outputs.

On the initial conceptualization and implementation of the CI/CD Service, each testbed will be composed of one Agent, which is already deployed before the validation of a NetApp starts.

To deploy the CI/CD Agent, a Virtual Machine (VM) Image will be provided to all partners, which can be instantiated using Openstack or another virtualization platform of their choosing.

On the instantiation of a CI/CD Agent, a cloud-init file will have to be provided. The cloud-init file will be responsible for creating a configuration file - config.ini - used as the source of several configurations on the CI/CD Agent. On boot, a systemd service will read config.ini. Given the information provided, it will register the CI/CD Agent on the CI/CD Manager and create a communication channel with the LTR of the facility.

The cloud-init configuration can be observed in Figure 43.

```
#cloud-config
password: password
chpasswd: { expire: False }
ssh_pwauth: True
write_files:
- content: |
    [JENKINS]
    DefaultUser = admin
    DefaultPassword = admin
    PasswordOutputFile = /var/lib/jenkins/jenkins_new_pw
    [CI_CD_MANAGER]
    Url = http://10.0.13.23:8000
    [LOG]
    Filepath =  /var/lib/jenkins/logs/startup.log
  path: /var/lib/jenkins/config.ini
  permissions: '0644'
```

Figure 43. CI/CD Agent - Cloud-Init Configuration File

Considering the shortage of months between the kick-off of the corresponding tasks, which implement the testing environment in each facility, and the delivery date of this document, the following table provides the implementation dates of the test node as estimations.

Facility	Installation readiness of test node
Aveiro	Q4 2021
Patras	Q4 2021
Bristol	Q4 2021
Bucharest	Q4 2021
Murcia	Q4 2021
Ljubljana	Q4 2021

Table 9. Installation readiness plan of test node per facility

7.2 Implementation to support hosting network slices

As described in section 2.1, part of the NetApp deployment request is the definition of requirements of the underlying Network Slice. To support this, as a first implementation phase, each facility of 5GASP will implement a generic eMBB slice, which will be used to host NetApps. At a later stage of 5GASP and while NetApp requirements for a Network Slice are more mature, additional slice types will be offered by facilities. This section provides the actions and current status per facility to support the hosting of network slice(s). This is implemented in terms of NSDs and VNFs that configure the Radio, Transport and Core Network of each facility. The section presents the first implementation plan and approach and estimated delivery date on how to support a minimum unified network slice template

(simple EMBB slice) and optionally reference any possible future extension towards more slice templates, if available.

7.2.1 Aveiro facility

The ITAV facility for 5GASP will rely on the existing 5G infrastructure as an access network. This access network operates side-to-side with a cloud infrastructure for providing support to the deployment of verticals' VNFs.

The 5G infrastructure, 5GAIner, is powered by Huawei and encompasses 5G New Radio (NR) radio cells, both outdoor (BBU 5900 + AAU 5649) and indoor (BBU 5900 + RHUB 5963 + pRRU 5961), together with a 5G SA Core enhanced with Multi-access Edge Computing capabilities. The infrastructure is geographically distributed between four different sites: (i) two indoor deployments within the university campus; (ii) one off-campus outdoor; and (iii) one off-campus indoor, edge-based deployment. Figure 44 presents the deployed 5GAINer architecture, identifying the different resources and their interconnection. On the left side, the four different site locations are represented. Locations with more than one antenna are configured to establish automatic neighbouring relationships in order to provide seamless handovers. On the right side, the core of the network is represented, as well as the networking providing communication support for the NFV Infrastructure (NFVI) where the 5G functions are instantiated. The NFVI is monitored by means of the eSight platform as well as the Mobile Automation Engine (MAE) solution. This last element also assumes the orchestration and life-cycle management mantle.

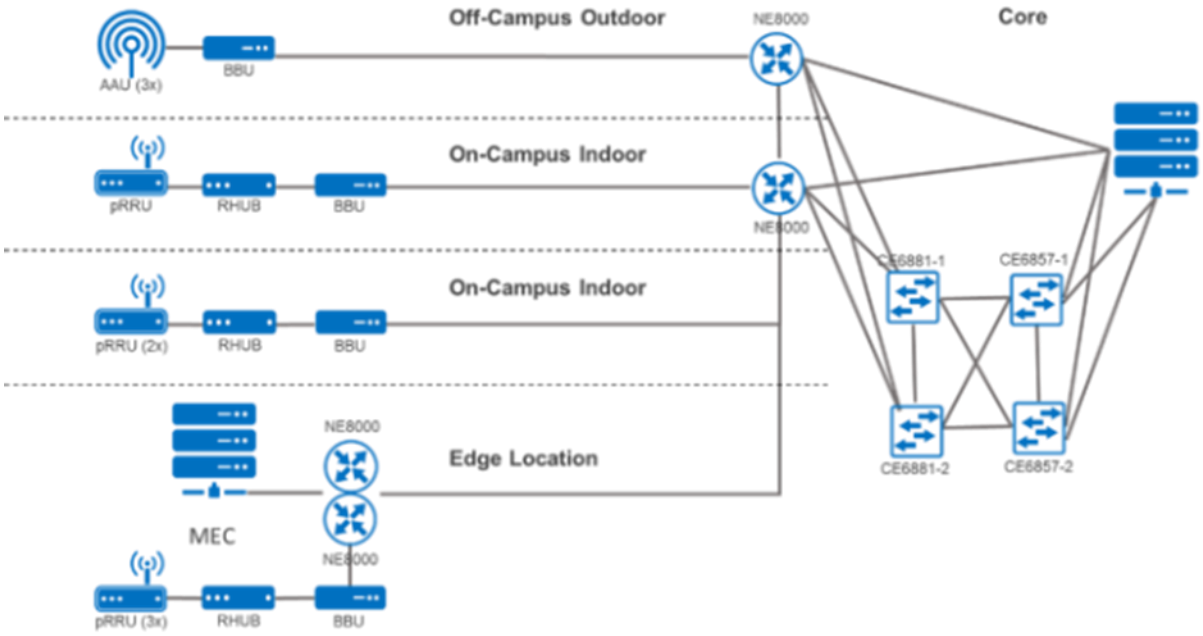


Figure 44. 5Gainer distributed 5G network

At the moment of writing of this deliverable a single eMBB slice is configured at the core. However, enhanced slicing support is expected as the new software releases are rolled out.

In the Core Network an OpenStack deployment always running the most recent version will provide computing and storage services to the 5G network.

7.2.2 Patras facility

The Patras facility will make available 5G Radios based on Amarisoft, while the core will be either the 5G Core of Amarisoft or Open5GS [41] (which is open source). Moreover, Open5GS is service based and can be deployed both as VMs or in a Kubernetes cluster.

For 5GASP, both an Openstack tenant for hosting VNFs as well as a Kubernetes cluster for hosting CNFs will be available. The following figures explain our approach to deliver the slice in terms of NSDs and VNFs that configure the Radio, Transport and Core Network.

Figure 45, displays the approach when delivering a slice in the Openstack VIM tenant, while Figure 46 displays the approach when the core is deployed in the Kubernetes cluster. In both cases there are two NSDs. One NSD contains VNFs/PNFs that configure our 5G Radios, while other NSDs contains VNFs (in the first case) that deploy the Open5GS in VMs; in the second case there is a NSD and VNFs with underlying Helm chart implemented that delivers the core in the Kubernetes clusters. Radios are configured to the requested QoS Class Identifier (QCI) of the slice. International Mobile Subscriber Identity (IMSIs) of our Subscriber Identification Module (SIM) cards are deployed on DAY 0, while it is possible to add more users while the core is in operation.

The plan is to make the orchestrated slice available for 5GASP in Q4 of 2021, ready to be ordered by 5GASP NODS and used by NetApps.

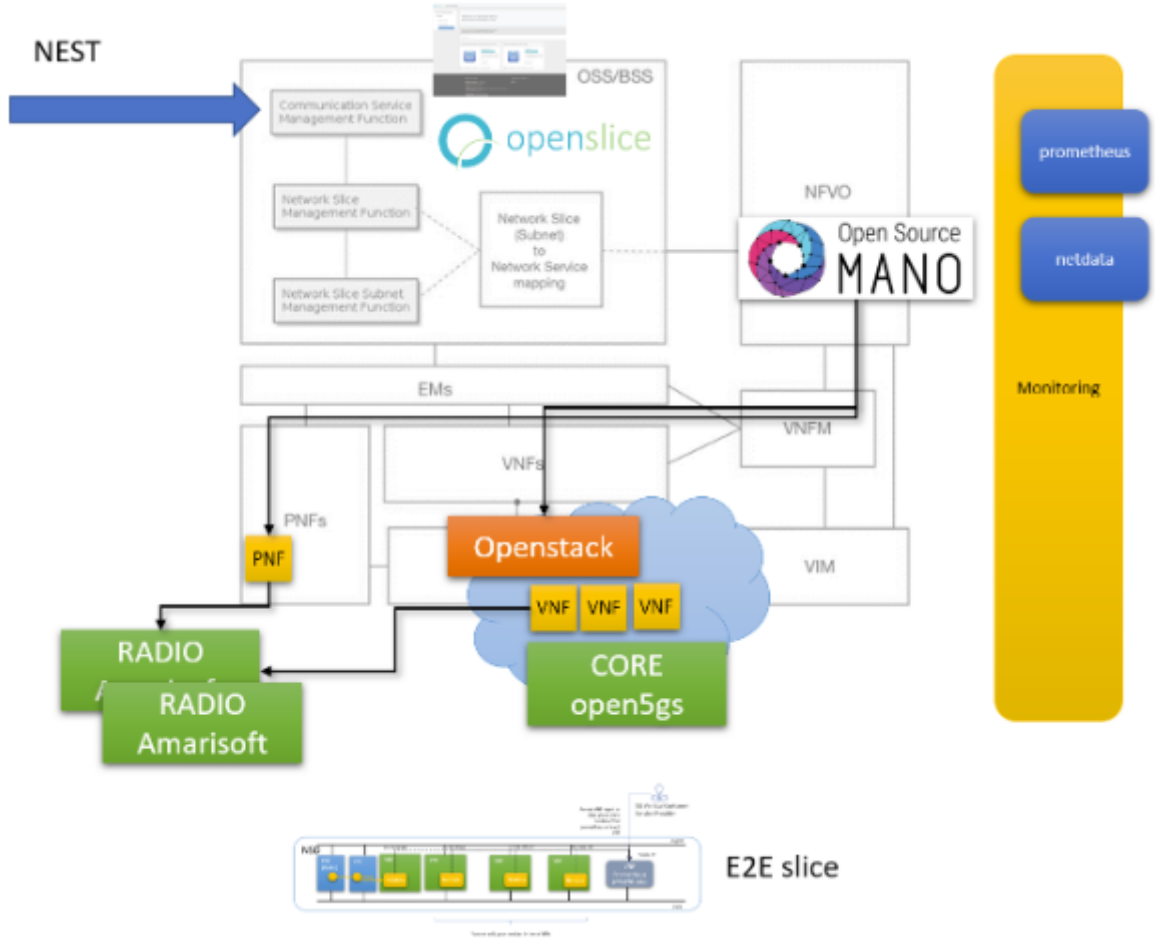


Figure 45. Slice delivery (Openstack VIM tenant)

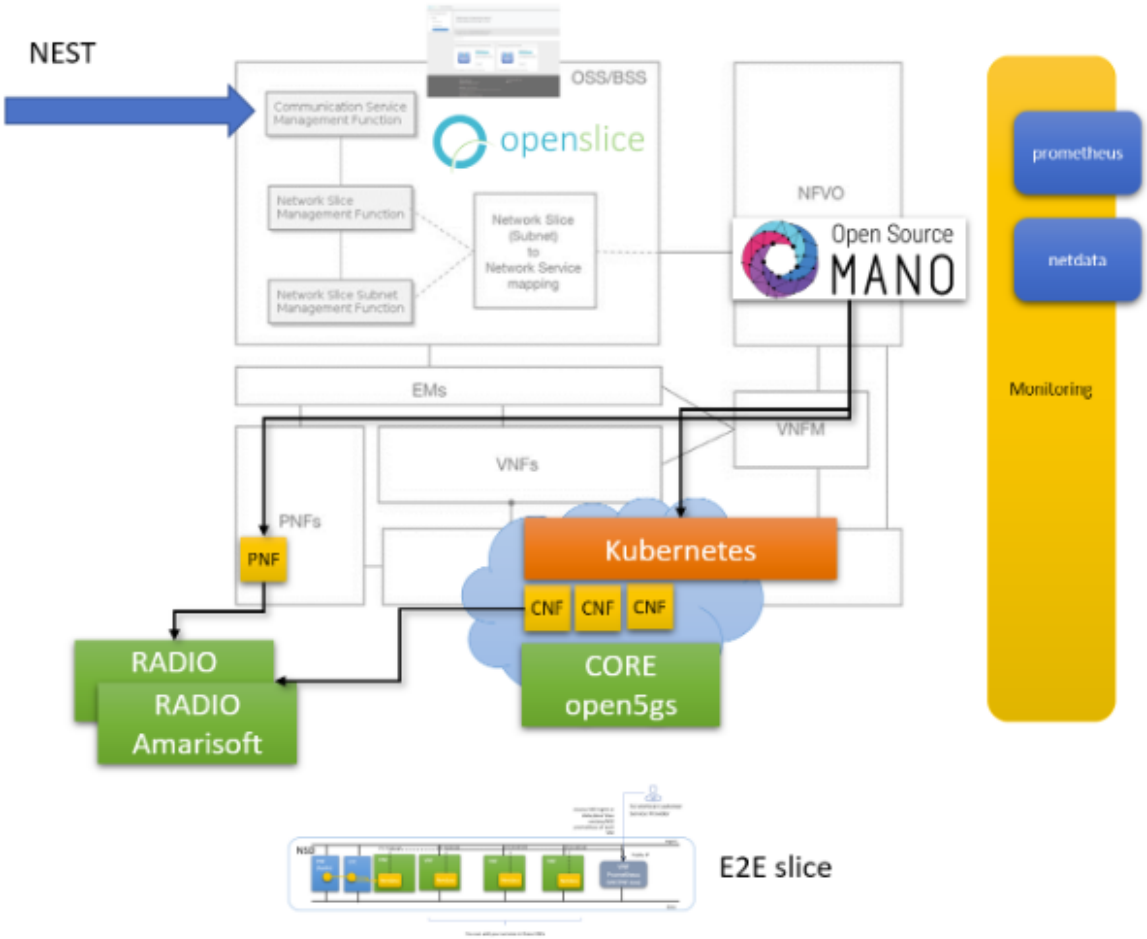


Figure 46. Slice delivery (Kubernetes cluster)

7.2.3 Bristol facility

The following phases are designed to prepare the UNIVBRIS testbed infrastructure, VIMs, and orchestrator to create 5G eMBB slice in deploying the required 5G Standalone Architecture (SA) core functions, multiple User Plane Functions (UPFs) instantiated at the edge nodes and setting up the gNodeBs (gNBs). Each phase includes details related to the design and requirements as well as timeline.

These phases are planned to be finished by end of 2021.

7.2.3.1 Phase 1 – Infrastructure setup and orchestration integration

This phase includes the hardware allocations, Transport Network setup, VIM implementation for two edges, orchestration integration and final configuration. The milestone is to have OpenStack virtualization platform with OSM orchestration on top of it to host different virtual network functions including 5G SA core and disaggregated UPFs function as well as different VMs to host a Kubernetes cluster.

Figure 47 presents the Transport Network entities, compute resources and PNFs for each location. As shown, there are two edge nodes with Multi-Access Edge Computing (MEC) facilities located at “We The Curious” (WTC) and MShed Museum (Mshed) venues. In this

configuration, the outdoor 5G coverage has an overlapping cell coverage area between these two hosting sites to create seamless handover of the services between the gNBs to support demonstration of various use cases.

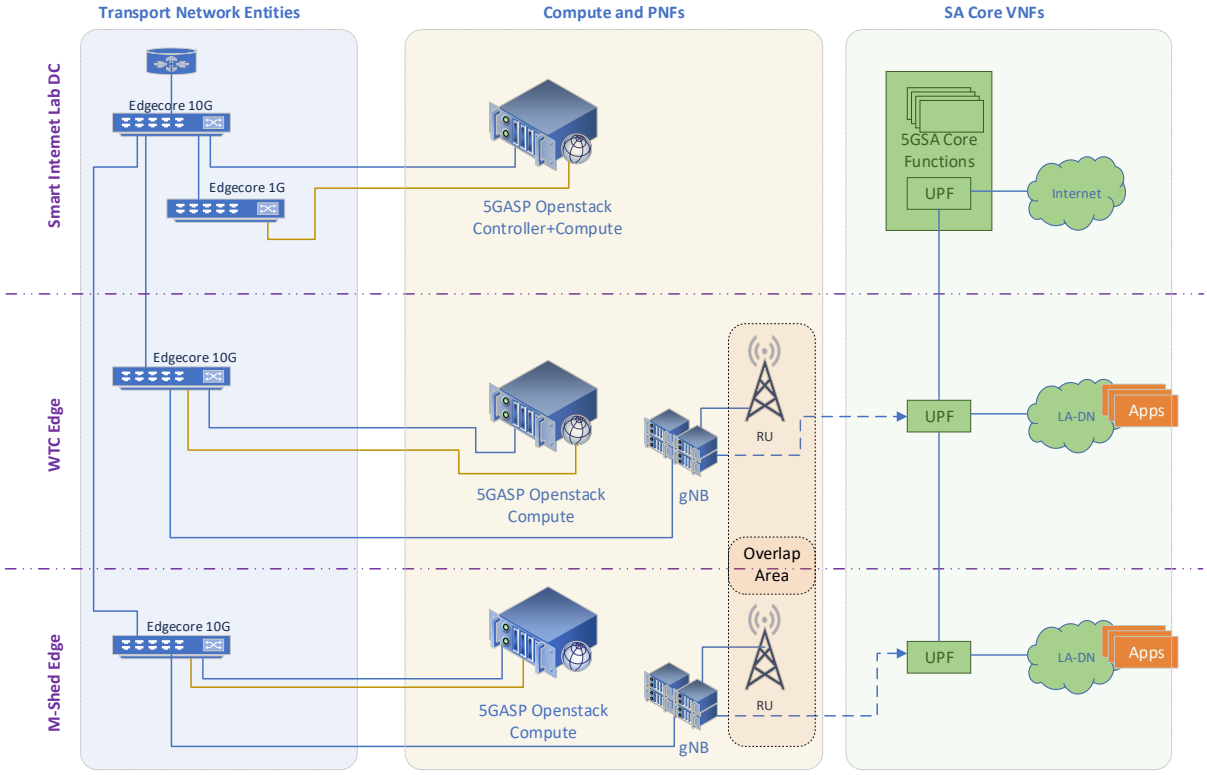


Figure 47. Physical location of entities at UNIVBRIS

This implementation also includes basic network services, such as Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS) and VPN to allow different partners to access this setup in UNIVBRIS testbed for remote configuration and use case evaluation.

Figure 48 presents the network overlay design in UNIVBRIS testbed where blue, brown and green subnets are dedicated to the 5GASP project as management, control and data plane networks respectively. The red and yellow subnets are the testbed VIM and orchestration management networks respectively. The service router will apply routes between different networks as well as the VPN users and the Internet.

To reach the milestone of this phase, three to four weeks is estimated for this activity.

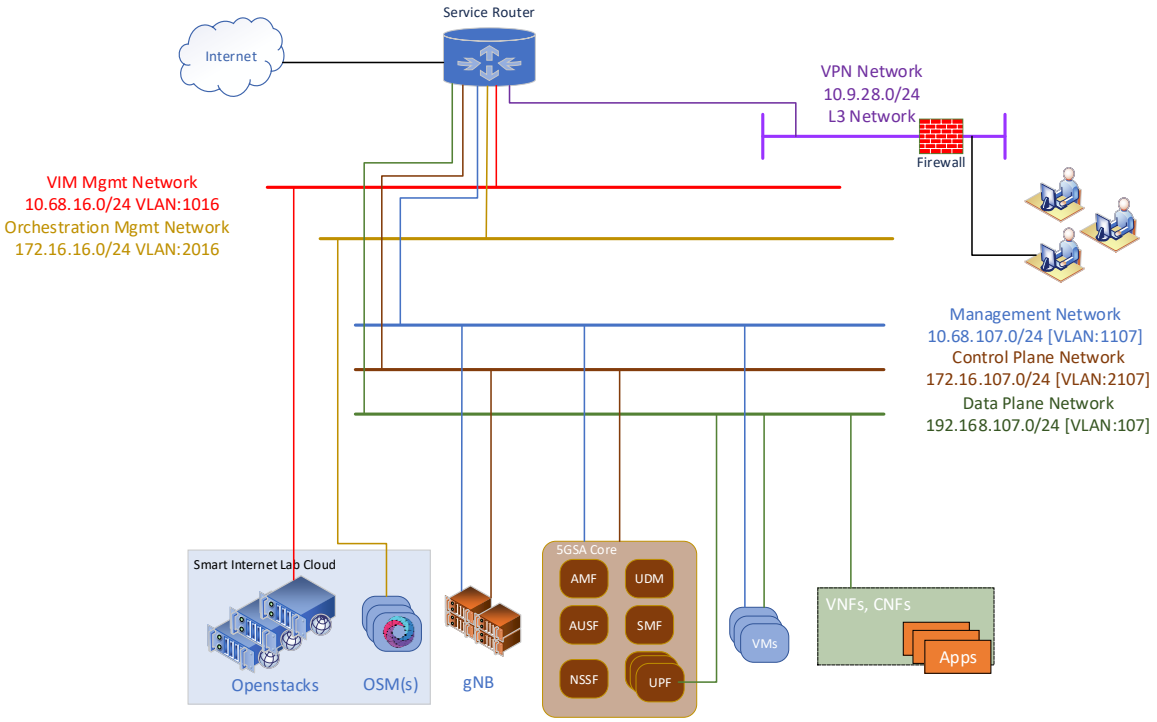


Figure 48. UNIVBRIS testbed network overlay design for 5GASP

7.2.3.2 Phase 2 – Setting up 5G core VMs and network services with initial configuration

In this phase all the required VMs will be created with initial configurations. This includes the VM that hosts most of the essential 5G SA core elements such as Access and Mobility Management Function (AMF), Session Management Function (SMF), Unified Data Management (UDM), Unified Data Repository (UDR), Policy Control Function (PCF), Authentication Server Function (AUSF), Network Repository Function (NRF), Policy and Charging Rules Function (PCRF) and Network Slice Selection Function (NSSF). UNIVBRIS will use Open5GS as a standalone core software.

The general 5G SA mobile network parameters such as Public Land Mobile Network (PLMN), Slice/Service Type (SST), default Slice Differentiator (SD), Data Network Name (DNN), subscriber database and gNB related parameters such as Tracking Area Code (TAC), etc., will be pre-configured to deliver a simple eMBB slice.

For achieving multiple edge sites with local data breakout, giving us the ability to route the traffic from each gNBs in each edge to appropriate MEC solution in other edge site, multiple UPF implementations have been considered. These UPFs will be attached to the same SMF via N4 link and interconnected using N9 Link. The routing table of each UPF will be preconfigured to route traffic to appropriate MEC based on the TAC and serving cell for each User Equipment (UE).

This activities in this phase are estimated to take as long as three to four weeks.

7.2.3.3 Phase 3 – VNFD and NSD definition for the 5G core and UPF VNFs with initial configuration

UNIVBRIS will use OSM as an orchestration solution to manage and orchestrate the compute and network resource allocated to this project. The milestone is to design and define required descriptors to deploy an eMBB slice by instantiating required VNFs in different VIMs and setting up the network connections at the end of the day.

In this phase, all the required VNFDs and NSDs will be defined based on the design requirements. An NSD will be shared with the UoPatras to be used by the 5GASP Global Service Orchestrator to instantiate the 5G SA core and UPF VNFs. These descriptors will include details related to the compute resources which are required by the 5G core VNF(s) and disaggregated UPFs, the network setup and the appropriate VIM information to place the functions in appropriate locations.

The required images which were created and preconfigured in previous phase will be used in this phase to deploy proper network services.

7.2.3.4 Phase 4 – Configuring gNB and integration with 5G core

UNIVBRIS will use Nokia gNBs and RUs to provide Radio Access Network (RAN) for the 5GASP project. Here, the RAN will be configured manually as the gNBs support multiple profiles for different PLMN IDs, where one of these will be the 5GASP designed PLMN ID. As an example, Figure 49 represents the configuration related to the PLMN in Nokia gNB.

*	MCC in PLMN MRBTS-2020107/NRBTS-2020107	mcc	1	1
*	MNC in PLMN MRBTS-2020107/NRBTS-2020107	mnc	5	5
*	MNC length of PLMN ID in primary PLMN of gNB MRBTS-2020107/NRBTS-2020107	mncLength	2	2
	NG User Plane IP configuration MRBTS-2020107/NRBTS-2020107	ngUplane		
	Structure 1			
	IPv4 address 1 distinguished name MRBTS-2020107/NRBTS-2020107	ipV4AddressDN1	MRBTS-2020107/TNLSVC-1/1	MRBTS-2020107/TNLSVC-1/TNL...
	IPv6 address 1 distinguished name MRBTS-2020107/NRBTS-2020107	ipV6AddressDN1		
	New Radio PLMN distinguished name MRBTS-2020107/NRBTS-2020107	nrPlmnDN		
	S1 User Plane IP configuration MRBTS-2020107/NRBTS-2020107	s1Uplane		
	Structure 1			
	IPv4 address 1 distinguished name MRBTS-2020107/NRBTS-2020107	ipV4AddressDN1	MRBTS-2020107/TNLSVC-1/1	MRBTS-2020107/TNLSVC-1/TNL...
	IPv6 address 1 distinguished name MRBTS-2020107/NRBTS-2020107	ipV6AddressDN1		
	New Radio PLMN distinguished name MRBTS-2020107/NRBTS-2020107	nrPlmnDN		
	Xn User Plane configuration MRBTS-2020107/NRBTS-2020107	xnUplane		
	Structure 1			
	Primary IPv4 address reference MRBTS-2020107/NRBTS-2020107	ipV4AddressDN1	MRBTS-2020107/TNLSVC-1/1	MRBTS-2020107/TNLSVC-1/TNL...
	IPv6 address 1 distinguished name MRBTS-2020107/NRBTS-2020107	ipV6AddressDN1		
	New Radio PLMN distinguished name MRBTS-2020107/NRBTS-2020107	nrPlmnDN		
*	Primary PLMN ID of adjacent E-UTRAN cell in ECGI	ecgiPlmn		

Figure 49. Nokia gNB Element Manager sample

Two gNBs are considered for this project, one per each edge. The N3 Link between the gNB and the edge's UPF will pass through a 10G edge switch.

The gNBs will also have access to the control plane network to interconnect with AMF via N2 using a high-performance transport network. It is feasible to implement the core functions at one of the edge nodes, or the central data centre, which is in Smart Internet Lab about 2km distance from the edge nodes.

This phase requires about one week for configuration and integration activities to be done.

7.2.3.5 Phase 5 – testing and evaluating eMBB slice

In the last phase, after having conducted all the previous phases successfully, it is possible to instantiate and deploy the 5G SA Core as a network service using OSM. This will create the appropriate VMs to host 5G functions as well as two UPFs, one per each edge node and setup the network connectivity for management, control, and data plane.

Because the gNBs have been configured in the previous phase, the gNBs will register themselves with the core and the eMBB slice will be finally deployed and ready for service delivery. The UEs will then be able to camp on to the 5G network and establish data session with access to appropriate networks based on the serving cell and TAC.

At this stage, number of KPIs, like end-to-end latency [ms], throughput [Mbps] and jitter [ms] will be measured and documented for further references, this also includes the functionality of handing over UEs from one cell to another cell and accessing appropriate LA-DN resources in each MEC. The KPI measurement will be recorded using general tools like iPerf [42] and ping based on different locations of the UEs.

This testing and evaluation phase will take about two to three weeks.

7.2.4 Bucharest facility

The Bucharest facility is located in Orange Romania 5GLAB and will make available several main 3GPP Rel.16 5G components for RAN, Core, Transport and Security, by using two network flavours: (1) 5G OpenTools based on 5G OpenAirInterface and (2) 5G Private Mobile Network and 5G Rel.16 RAN (Nokia). Based on this configuration setup, as the 1st implementation phase, the eMBB network slice will be implemented based on the facility capabilities, while also being able to deploy at least another different network slice. The Bucharest facility will also host the NetApps in the virtualised infrastructure as Openstack based VMs or Kubernetes CNFs.

As for 5GASP in the ORO facility, there will be a dedicated 5GASP tenant in Openstack for VMs and/or a Kubernetes for CNFs providing the resources isolation.

The high-level approach to deliver the slice in both scenarios (VMs and Kubernetes) is based on Figure 50, highlighting the main components.

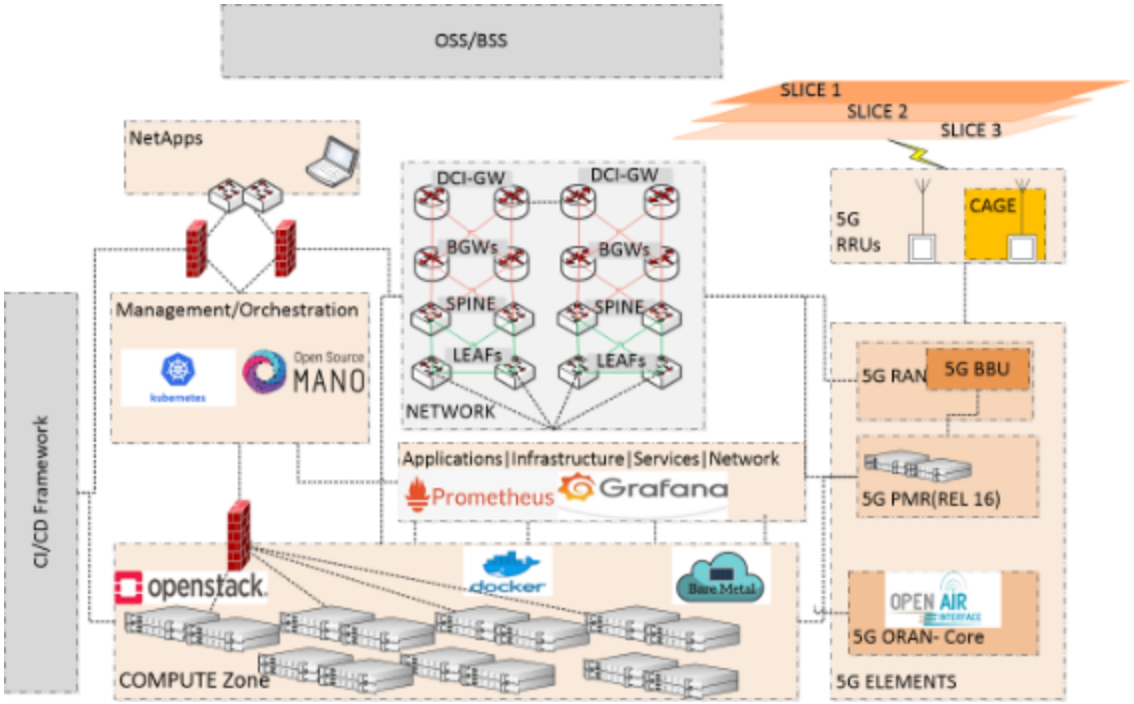


Figure 50. Bucharest facility 5G slice delivery

The delivery of 1st slice will be offered through the 5G based infrastructure for RAN and core, in a turnkey or customizable solution in a later stage, based on the 5G SA Core Network evolution in Bucharest facility. Parameters are set manually at this stage and Orange is investigating the end-to-end dynamic configuration and slice setup.

Orange Romania plan is to deploy the first network slice by the end of 2021.

7.2.5 Murcia facility

The testbed located in Murcia will offer a 5G infrastructure based on a turnkey solution provided by Amarisoft, both for the RAN and for the core. In addition, the site has also availability to accommodate open source 5G cores. An Openstack tenant will be available for 5GASP requirements to host NetApps in the form of VNFs.

To deliver the initial slice, the 5G radio and the 5G core configuration parameters will be tuned by using the Amarisoft management software. This slice will be then extended to the Openstack VIM tenant to deliver the E2E slice. Although this approach is essentially static, Murcia site team is currently designing and working in a dynamic solution that will manage and orchestrate the E2E slicing from the 5G infrastructure to the virtualisation platform, through the Transport Network. At the same time, the team is also working on a monitoring platform that will accumulate metrics and compute statistics of the whole infrastructure.

The plan is to offer the initial slice allocation for 5GASP at the end of 2021.

7.2.6 Ljubljana facility

The facility in Ljubljana will provide 5G mobile network infrastructure in SA mode using in-house built container-based images for gNB and Core Network components (based on a commercial software) with the option to run additional 3rd party core components at later stages of the project. Data centre capabilities will be provided on top of OpenStack-based

cloud with a dedicated tenant being available for 5GASP project. Additionally, dedicated Kubernetes cluster is available to allow deployment of CNF-based components.

The 5G network slice will be provisioned and configured through VNFs which will host the gNB and Core components. Certain configuration parameters regarding the 5G slice will be exposed as DAY 0 configuration which will allow simplified setting of parameters such as Mobile Country Code/Mobile Network Code (MCC/MNC), Channel bandwidth, etc. The configuration options are then passed to the appropriate components so the full 5G network service is established from RAN to the Core where also VNF/CNFs from NetApps will be deployed.

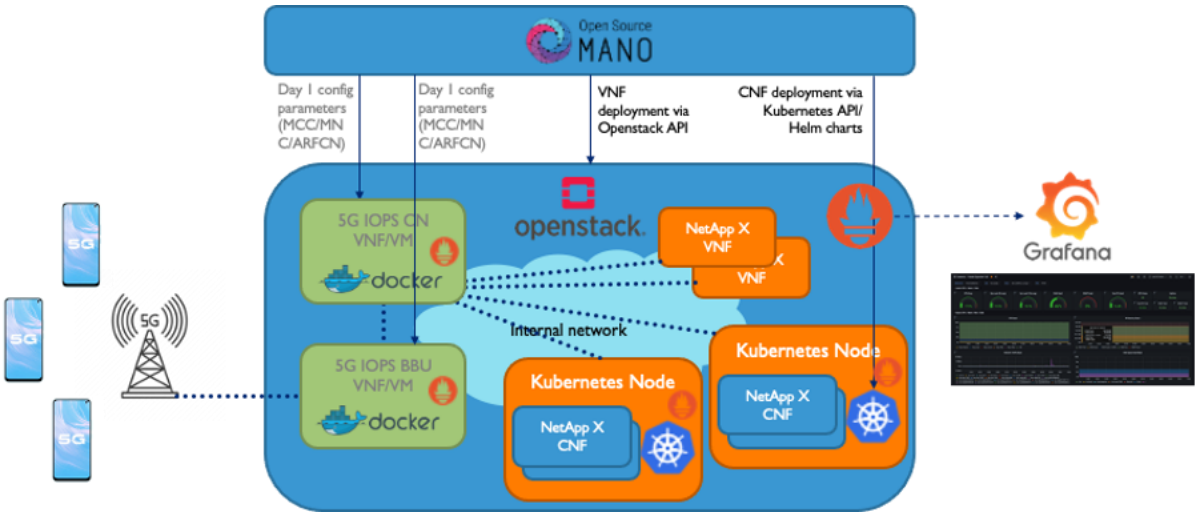


Figure 51. ININ's 5G network slice delivery

Cloud monitoring will be realized via Prometheus [43] supported solution such as node exporter or Netdata [44] and will be available as graphical dashboards provided by Grafana [45]. For network monitoring, ININ's commercial qMON monitoring solution [46] will be used. It allows placing agents and reference servers as VNFs/CNFs and also running agent software as an Application on Android 11+ device. The latter will provide the capability to test true end-to-end scenarios, i.e. from the end user device and NetApp components running in backend. Near real-time analysis will be provided by Grafana dashboards, detailed post-analytics is also available to 5GASP project participants and can be provided based on commercial software by Tableau [47].

Initial slice allocations and NetApps deployment is expected to be available at the end of 2021.

7.3 Proposals to support vertical's needs

This section presents the current envisioned plans by some facilities to support the specific needs of the verticals. In future deployments, components from NetApps need to be integrated, or specific hardware (e.g. sensors, 5G UEs) need to be installed in the facility for example were the 5G Radio access is available. It is expected to incorporate more details, depending on the NetApps needs, in future revisions of this deliverable.

7.3.1 Automotive (Murcia site)

Murcia site plans to offer support for the automotive vertical by means of 802.11p technology and IPv6 connectivity. It will be available with dual role (RSU/BSU capable PCEngines APU3d2 devices) units which can be deployed alongside Espinardo campus for vehicular experimentation, on demand. It will be also integrated in the multi-access infrastructure of the Gaia5G facility, but this integration is still being designed.

Figure 52. Multi-access infrastructure in Murcia campus

7.3.2 Public Protection and Disaster Relief (Ljubljana site)

Ljubljana site will offer support for Public Protection and Disaster Relief (PPDR) vertical on 5G and cloud infrastructure. They primarily target to support PPDR disaster scenarios were due to the natural or manmade accidents several elements of mobile infrastructure could be down or non-operational. In this regard, Ljubljana site will exploit several 5G technologies such as 5G NR for the PPDR user equipment access and cloud-based capabilities provided on top of OpenStack-based Infrastructure as a Service (IaaS). Additionally, dedicated Kubernetes cluster will be available to allow deployment of PPDR related network and services CNF components.

The facility will be based on ININ's PPDRone solution which comprises a Software Defined Radio (SDR) and Common Public Radio Interface (CPRI) based radio and mobile core system with flexible configuration options powered by NFV, cloud backend infrastructure, services, test and validation toolkit.

PPDR cloud monitoring will be realized via Prometheus supported solution and Grafana graphical dashboards. For end-to-end (E2E) PPDR network and services monitoring, ININ's commercial qMON monitoring solution will be exploited, which allows placing agents and reference servers as VNFs/CNFs and also running agent software as an application on Android 11+ device in order to provide true end-to-end test and monitoring scenarios, i.e. from the PPDR end user terminal to the PPDR NetApp components running in backend. Near real-time analysis will be available, provided by Grafana dashboards, detailed post-analytics will be also available via Tableau business analytics software.

The site will primarily support development and deployment of a NetApp for 5G Isolated Operation for Public Safety (5G IOPS) which will assure PPDR users support for local (on disaster site) 5G coverage even in the most extreme disaster situations (e.g. earthquake). 5G IOPS can be used for day-to-day PPDR operations to assure high availability of the PPDR services hosted by the mobile operators. In addition, NetApp is also applicable to provide nomadic network deployments where PPDR users can deploy 5G network services in ad-hoc fashion (e.g. vehicle mounted) on one or several 5G IOPS enabled gNbs.

In addition, Ljubljana site will be also integrated with Patras's facility enabling cross-border PPDR operations in multidomain 5G environments.

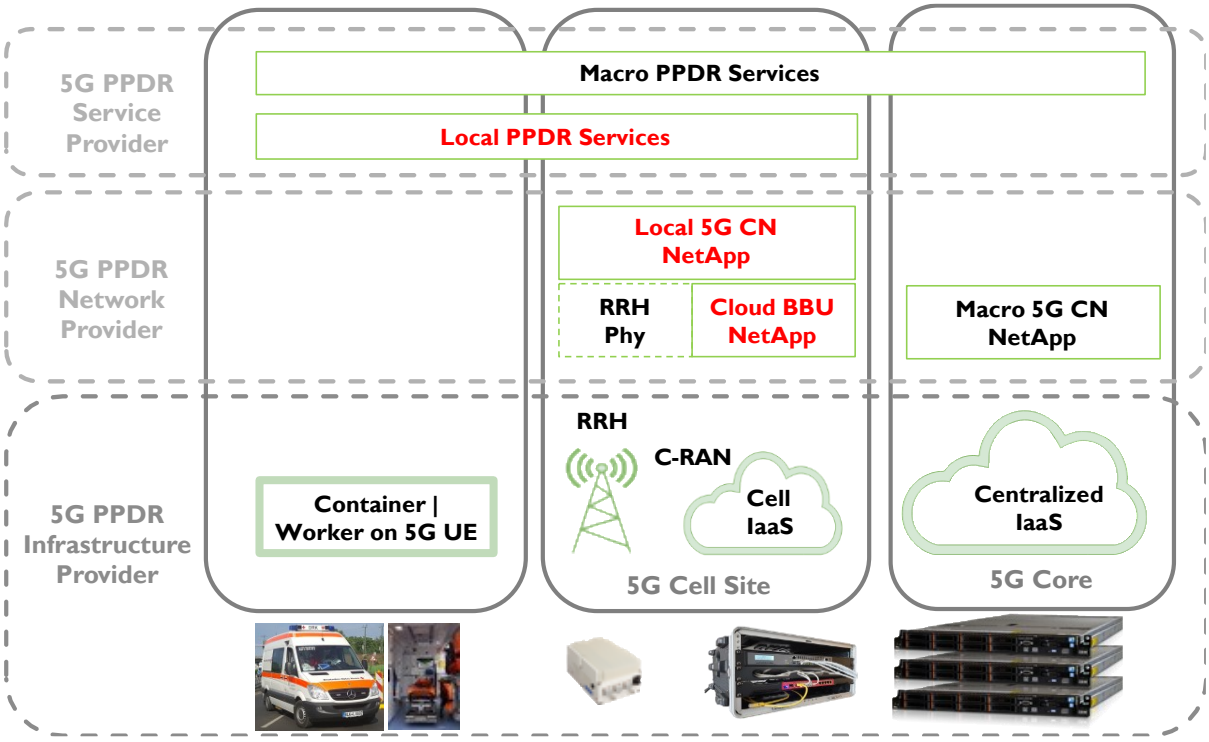


Figure 53: 5G IOPS NetApp deployment architecture in the 5GASP environment

7.4 Aggregated implementation plan

This section combines the implementation plans of previous subsections to a summarised table per facility site. Therefore, the medium-term schedule of WP3 is introduced and fragmented into a series of distinct tasks along with their expected timeframe, as follows:

- **Task #1:** Exposure of NFVO (Interface E1)
- **Task #2:** Installation of CI/CD Agent VM
- **Task #3:** Provision of simple eMBB slice
- **Task #4:** Interconnection between facilities (Interfaces E4, E7)
- **Task #5:** 1st iteration of NetApp(s) testbed access and onboarding (associated NetApp provider)
- **Task #6:** Provision of custom slice related to the onboarded NetApp(s) requirements

	Aveiro	Patras	Bristol	Bucharest	Murcia	Ljubljana
Task #1	Q4 2021	Q4 2021	Q4 2021	Q4 2021	Q4 2021	Q4 2021
Task #2	Q4 2021	Q4 2021	Q4 2021	Q4 2021	Q4 2021	Q4 2021
Task #3	Q1 2022	Q4 2021	Q4 2021	Q4 2021	Q4 2021	Q4 2021
Task #4	Q4 2021 – Q1 2022	Q4 2021 – Q1 2022	Q4 2021 – Q1 2022	Q4 2021 – Q1 2022	Q4 2021 – Q1 2022	Q4 2021 – Q1 2022
Task #5	Q1 2022 (OdinS)	Q4 2021 (Lamda Networks)	Q1 2022 (Lamda Networks, Bristol)	Q1 2022 (Neo)	Q4 2021 (OdinS)	Q4 2021 (ININ)
Task #6	Q2 2022	Q1 – Q2 2022	Q2 2022	Q2 2022	Q1 – Q2 2022	Q2 2022

Table 10. Aggregated implementation plan per facility site

8 Conclusion

As one of the core concepts of the 5GASP project is to provide a unified interactive model, the document focuses on the harmonisation of the design and data models introduced by widely accepted telecommunication standard bodies. Therefore, this design offers a familiar ecosystem for developers to work within the project and ensure interoperability and reproducibility among various systems, simultaneously supporting a more comprehensive range of verticals. Furthermore, this document presents the basic requirements to design the common platform to accelerate the development, testing and certification of NetApps for specific verticals use-cases listed in the reference architecture in WP2 and detailed in D2.1.

The document tackles the details of the internal architecture components as a service-based model, linked to the current release (Q2 2021 release) of the 5GASP portal, followed by a brief investigation of the peripheral components of the portrayed platform, i.e. the novel automated CI/CD toolchain, the multi-domain NFV fabric, the public registry of SMEs and their registered products (marketplace). The content presentation included identifying and defining the communication interfaces between these components, which are extensively described.

The basic requirements of the central 5GASP portal, i.e. 5GASP NODS, have been detailed while highlighting the deployment model supported by a triplet entity comprised of NetApp artefacts, network slice templates and test descriptors. Moreover, the identification of actors within NODS architecture described each player's role, e.g. NetApp/NF developer, Service Designer/Provider or Platform Administrator roles.

The project partners' testbeds are being made available to SMEs of different verticals looking for NetApp deployment, testing and validation. The requirements, technologies and interconnection options of an inter-domain Open5G-NFV ecosystem which supports the required reproducible environments were detailed. Moreover, other issues related to security and trust aspects associated with running 3rd party software in experimental network environments are further estimated.

5GASP ecosystem interaction with different services over seven well-defined interfaces in the architecture was additionally listed. Four options have been identified for interconnection at the services level between 5GASP NODS and the underlying facilities. The listed options considered the different technological stacks deployed in each facility. All the six listed facilities have elected the direct exposure of an ETSI-NFV compliant MANO stack and NODS employment as an E2E Service Orchestrator (Option B), as an initial interaction choice. The initial decision was based on the fact that it is the most viable implementation scenario to begin with.

The document further reports the expected implementation capabilities required from each testbed. On that notion, the latest update was provided regarding the readiness, as well as the current status of each facility to support the multi-domain fabric and the hosting of network slice(s). Furthermore, several potential verticals' needs were identified along with envisioned plans by some facilities to meet these needs.

For future work, partners will gather additional requirements and annotations laid out from users' perspective that will further aid the implementation and deployment details of the 5GASP experimentation infrastructure for NetApps. An updated version (D3.2), which shall

evaluate and validate the preliminary proposals of this document based on the learned lessons from the 5GASP framework users should be expected in M18, followed by the final release in M36 (D3.3).

References

[1] 5GASP, “D2.1 Architecture, Model Entities Specification and Design,” 2021. [Online]. Available: https://5gasp.eu/assets/documents/deliverables/D2.1%20Architecture,%20Model%20Entities%20Specification%20and%20Design.pdf. [Accessed September 2021].

[2] TM Forum, "TMF633 - Service Catalog Management API REST Specification".

[3] TM Forum, “TMF641 - Service Ordering Management API REST Specification”.

[4] IETF, “YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF),” 2010.

[5] OASIS, “Instance Model for TOSCA Version 1.0,” 2017.

[6] ETSI NFV ISG, “GR NFV-IFA 029, Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS",” 2019.

[7] GSMA, "Generic Network Slice Template v5.0," 2021. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads/NG.116-v5.0-7.pdf. [Accessed September 2021].

[8] 5G EVE, [Online]. Available: https://www.5g-eve.eu/. [Accessed September 2021].

[9] 5GTANGO, [Online]. Available: https://www.5gtango.eu/. [Accessed September 2021].

[10] TM Forum, “TMF653 - Service Test Management API REST Specification”.

[11] OpenSlice, [Online]. Available: http://openslice.io. [Accessed September 2021].

[12] TM Forum, "TMF 909A - Network as a Service (NaaS) API Component Suite Profile," 2019.

[13] HashiCorp, “Consul,” [Online]. Available: https://www.consul.io/. [Accessed September 2021].

[14] IETF, “The OAuth 2.0 Authorization Framework,” 2012.

[15] Keycloak, [Online]. Available: https://www.keycloak.org/. [Accessed September 2021].

[16] ETSI, “GS NFV-SOL 006; Protocols and Data Models; NFV descriptors based on YANG Specification,” 2020.

[17] ETSI, “GS NFV-SOL 001; Protocols and Data Models; NFV descriptors based on TOSCA Specification,” 2020.

[18] Bugzilla. [Online]. Available: https://www.bugzilla.org/. [Accessed September 2021].

[19] Elastic Stack, [Online]. Available: https://www.elastic.co/what-is/elk-stack. [Accessed September 2021].

[20] Flowable, [Online]. Available: https://flowable.com/products/flowable-orchestrate/. [Accessed September 2021].

[21] OMG, “Business Process Model and Notation (BPMN),” 2011.

[22] 3GPP, "TR 28.801; Telecommunication management; Study on management and orchestration of network slicing for next generation network (Release 15)," 2018.

[23] Google, “Blockly,” [Online]. Available: https://developers.google.com/blockly. [Accessed September 2021].

[24] ETSI, “Open Source MANO (OSM),” [Online]. Available: https://osm.etsi.org/. [Accessed September 2021].

[25] ETSI, "GS NFV-SOL 005; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point," 2020.

[26] Linux Foundation, “Open Network Automation Platform (ONAP),” [Online]. Available: https://www.onap.org/. [Accessed September 2021].

[27] Apache, “ActiveMQ,” [Online]. Available: https://activemq.apache.org/. [Accessed September 2021].

[28] Apache, “Camel,” [Online]. Available: https://camel.apache.org/. [Accessed September 2021].

[29] TM Forum, "TMF638 - Service Inventory API REST Specification".

[30] 3GPP, “TS 28.530, Management and orchestration; Concepts, use cases and requirements (Release 16),” 2020.

[31] TM Forum, “TMF620 - Product Catalog Management API REST Specification”.

[32] WireGuard, “WireGuard: Next Generation Kernel Network Tunnel,” [Online]. Available: https://www.wireguard.com/papers/wireguard.pdf. [Accessed September 2021].

[33] OpenVPN, [Online]. Available: https://openvpn.net/. [Accessed September 2021].

[34] IETF, “Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks,” 2014.

[35] Openstack, “Neutron,” [Online]. Available: https://www.openstack.org/software/releases/ocata/components/neutron. [Accessed September 2021].

[36] Openstack, [Online]. Available: https://www.openstack.org/. [Accessed September 2021].

[37] Kubernetes, [Online]. Available: https://kubernetes.io/. [Accessed September 2021].

[38] IETF, “Internet Key Exchange Protocol Version 2 (IKEv2),” 2014.

[39] IETF, “Generic Routing Encapsulation (GRE),” 2000.

[40] IETF, “A Simple SCCP Tunneling Protocol (SSTP),” 1999.

[41] Open5GS, [Online]. Available: https://open5gs.org/. [Accessed September 2021].

[42] iPerf, [Online]. Available: https://iperf.fr/. [Accessed September 2021].

[43] Prometheus, [Online]. Available: https://prometheus.io/. [Accessed September 2021].

[44] Netdata, [Online]. Available: https://www.netdata.cloud/. [Accessed September 2021].

[45] Grafana Labs, “Grafana,” [Online]. Available: https://grafana.com/. [Accessed September 2021].

[46] ININ, “qMON - quality monitoring suite,” [Online]. Available: http://www.iinstitute.eu/#qmon. [Accessed September 2021].

[47] Tableau, [Online]. Available: https://www.tableau.com. [Accessed September 2021].