H2020 5GASP Project

Grant No. 101016448

D3.2 5GASP experimentation services, middleware and multi-domain facilities continuous integration, 2nd release

Abstract

This document provides an update to the status of 5GASP experimental services, interfaces, facilities support to host the NetApps and the implementation details to create the multi-domain SDN/NFV fabric. This document is a first update to the 5GASP infrastructure and services implementation plan described in the deliverable D3.1. Alongside the status of various components of the 5GASP ecosystem, it will also cover if there are any changes to the proposed implementation plan and further planned enhancements in the 5GASP infrastructure.

Document properties

Document number	D3.2
Document title	5GASP experimentation services, middleware and multi-domain facilities continuous integration, 2nd release
Document responsible	N. Uniyal, X. Vasilakos
Document editor	N. Uniyal, X. Vasilakos
Editorial team	UNIVBRIS, VMware, Neobility
Target dissemination level	PU
Status of the document	Final Version
Version	2.0

Document history

Revision	Date	Issued by	Description
Initial	10/03/2022	UNIVBRIS	Initial draft
V0.2	11/04/2022	UNIVBRIS	Updated draft
V1.0	20/04/2022	UNIVBRIS	Final ToC
V1.1	20/06/2022	UNIVBRIS	Collated contributions from all partners except ORO. Conclusion missing.
V1.2	22/06/2022	UNIVBRIS	All contributions updated, conclusion missing
V2.0	29/06/2022	UNIVBRIS	Final draft reviewed and edited version

Disclaimer

Document Authors

Company	Name	Contribution
UNIVBRIS	Navdeep Uniyal, Xenofon Vasilakos, Constantinos Vrontos	Introduction, Abstract, Implementation to support hosting network slices, Future implementation plans, Support for vertical needs-Bristol facility, Review
ITAv	Rafael Direito Daniel Gomes Diogo Gomes	5GASP Network Orchestrator, Cross Domain Network Orchestration - Interface E4, Inter-facility connectivity - Interface E7, Design, Underlying technologies and security considerations, Implementation of facilities

		interconnection, Implementation to support hosting network slices, Future implementation plans
UoP	Kostis Trantzas Christos Tranoris Ioannis Chatzis	NODS architecture- Recap, NetApp onboarding and experiment management portal, 5GASP experimentation APIs service, 5GASP Service Orchestrator, MANO Client API service, Summary table with current status, Interaction with CI/CD Service - Interface E2, Communication with the NFVOs - Interface E1, Facility and testing management - Interface E5, 5GASP NODS Implementation Release 2, Implementation of E1 interface, Implementation to support hosting network slices - Patras facility, Support for vertical's needs - Patras facility
OdinS	Jorge Gallego-Madrid Ana Hermosilla	OdinS' facility information, Support for Vertical's needs
YoGoKo	Mohammad-Yakub ABUALHOUL	Conclusion
VMware	Vesselin Arnaudov	Review
Neobility	Andrei Radulescu	Review
ORO	Marius Iordache Catalin Brezeanu	Implemented Capabilities from 5GASP Facilities, Future Implementation Plans
ININ	Janez Sterle Rudolf Susnik Luka Korsic	ININ's facility information Review

Contents

List of Figures

List of Tables

List of Acronyms

Acronyms	Full Form
5GASP	5G Application & Services experimentation and certification Platform
API	Application Programming Interface
BPMN	Business Process Model and Notation
CFSS	Customer Facing Service Specification
CI/CD	Continuous Integration/ Continuous Development
CIMs	Cloud Infrastructure Managers
CNF	Containerized Network Functions
CSMF	Communications Service Management Function
E2E	End to End
eMBB	enhanced Mobile BroadBand

gNB	g Node B
GUI	Graphical User Interface
IKEv2	Internet Key Exchange version 2
IPSec	Internet Protocol Security
LCM	Lifecycle management
MAC	Media Access Control
MANO	Management and Orchestration
NAT	Network Address Translation
NBI	North Bound Interface
NEST	Network Slice Template
NetApp	Network Application
NetOr	Network Orchestrator
NFV	Network Function Virtualisation
NFVO	Network Function Virtualisation Orchestrator
NODS	NetApp Onboarding and Deployment Services
NS	Network Service
NSD	Network Service Descriptor
NSI	Network Slice Instance
NSMF	Network Slice Management Function
OSM	Open Source MANO
P2P	Peer to Peer
RAN	Radio Access Network
RU	Radio Unit
SMF	Session Management Function
SSL	Secure Sockets Layer
TMF	TeleManagement Forum
TRVD	Test Results Visualization Dashboard
UI	User Interface
UPF	User Plane Function
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VPN	Virtual Private Network
VPN	Virtual Private Network
WP	Work Package

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 behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior 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

This document is intended to cover the details on the implementation of the 5G Application & Services experimentation and certification Platform (5GASP) reference architecture explained in Work package (WP) 2 documented in D2.1 [1]. This document is an extension of the first version described in the deliverable D3.1 [2]. It will further cover the core building blocks of the 5GASP architecture.

Based on the technical goals presented in D3.1 [2], this document covers the up-to-date design and implementation details of the NetApp Onboarding and Deployment Services (NODS) portal. Along with it, this document will cover the update on the 5GASP interface design and implementation introduced in D3.1 [2]. The explicit reference and update on the status of internal services would provide an up-to-date view on the current progress of this task.

Following on the description in D3.1 [2], the reader can get further details on the interaction with the CI/CD toolchain. The user will also get a view on the interaction between/with the other facilities like NFVO, testing framework along with the 5GASP NetApp store.

Furthermore, this document will cover the interdomain connectivity between the 5GASP test sites. The 5GASP multi-domain NFV fabric is designed to allow secure deployment of Network Services across the testbeds. The updated security considerations will be described in section 5.2 of this document.

We describe the deployed Enhanced Mobile Broadband (eMBB) network slices at all the 5GASP testbeds from the NODS portal in section 6.2. Finally, the section 6.4 summarises a table to match the NetApp requirements to the testbed capabilities.

2 5GASP NetApp onboarding and deployment services (NODS) design and implementation

2.1 NODS architecture- Recap

This section comprises the overall NODS architecture as it was introduced in D3.1 [2], hence not exhaustively explained again, enhanced with the latest updates. The NODS solution is based on the open-source project OpenSlice [3]. Figure 2-1 illustrates the latest updates in **bold**, also highlighting the interaction of the established components with newly added ones. During the latest update, we did not identify the need to alter the actors' list, as it was introduced in the previously referenced document.

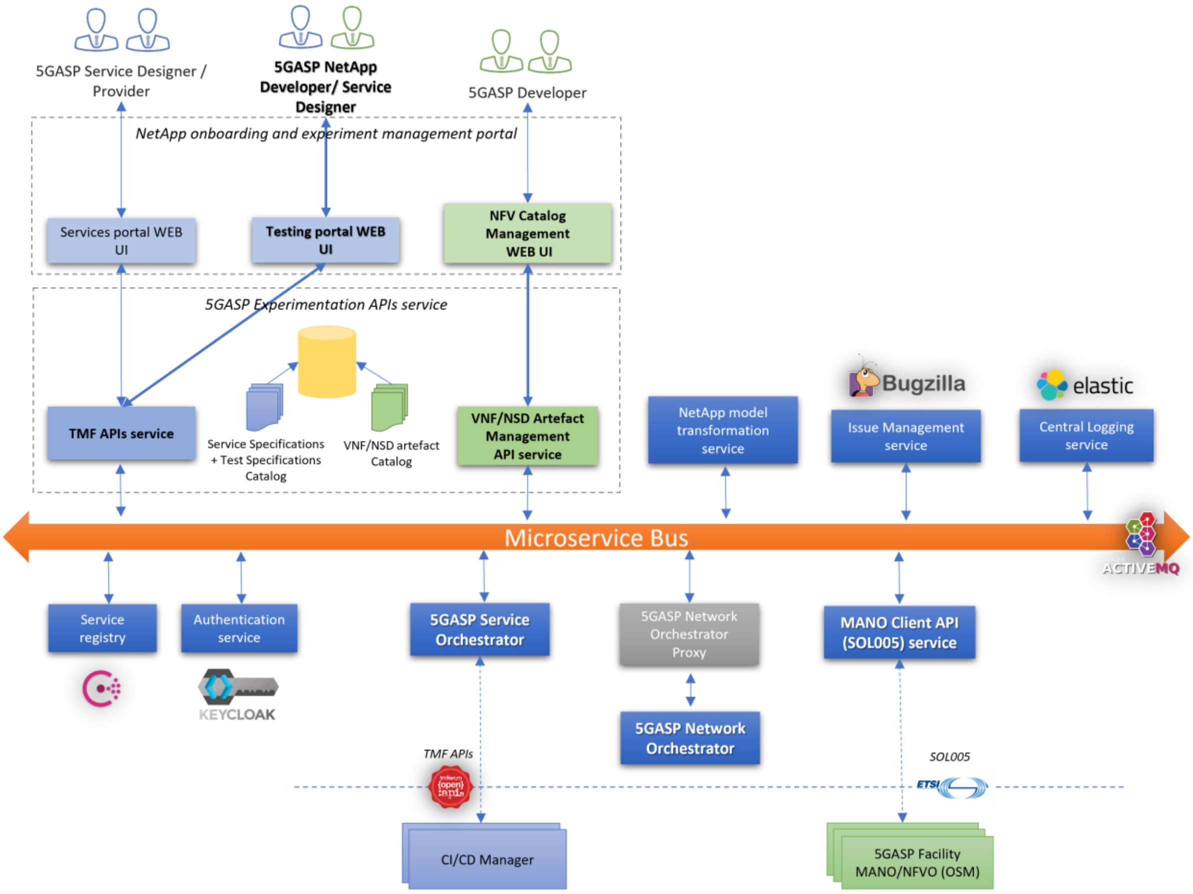


Figure 2-1: 5GASP NODS updated architecture

In summary, the newly introduced as well as the updated components are the following:

- The NetApp onboarding and experiment management portal,
- The 5GASP experimentation APIs service,
- The 5GASP Service Orchestrator,
- The 5GASP Network Orchestrator,
- The Management and Orchestration (MANO) client API service.

The following section, i.e. section 2.2, includes the updated components and services since D3.1 [2].

2.2 Internal services – updates

2.2.1 NetApp onboarding and experiment management portal

The NetApp onboarding and experiment management portal was initially designed to provide a single entry-point to relevant actors. As we have not identified new actors that compromised this assumption, the portal was augmented with the addition of another User Interface (UI), explicitly to facilitate the test design and importing. Hence, as it is reproduced in Figure 2-2, the portal consists of:

- A NFV catalogue management web UI,
- A Service portal web UI,
- A Testing portal web UI

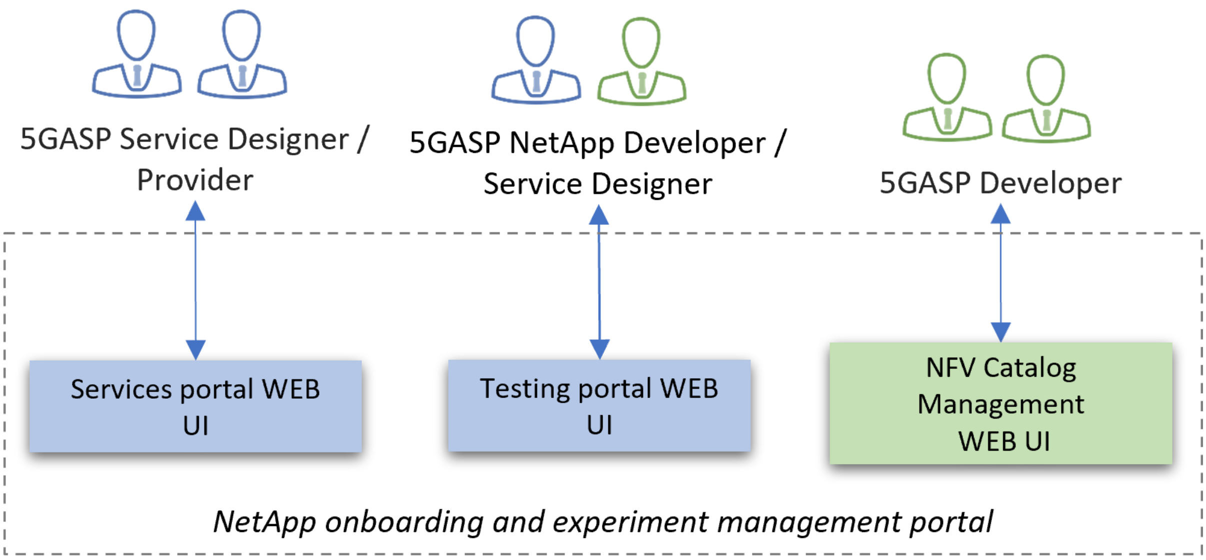


Figure 2-2: NetApp onboarding and experiment management portal as a single entry-point (updated)

The NFV catalogue management UI retains its primary usage as the portal that enables 5GASP Developer to onboard the NetApp packages, i.e. NFV artefacts. Specifically, all the underlying testbeds, that comprise the unified facility, are eventually registered into NODS, hence the 5GASP Developer may currently choose to onboard artefacts to the testbed of their preference. Also, distinctive messages are presented to the developer, following their actions, validating the onboarding of artefacts or prompting for possible errors, respectively. Additionally, the management interface that is offered to the 5GASP platform is extensively augmented to support the management needs of the several underlying administrative domains. The substantial augmentation was to support a bidirectional synchronization procedure that work both from NODS to underlying domains and vice versa. We have identified that data integrity issues emerged, as more and more artefacts were onboarded at the project's domains. That being said and acknowledging that the NODS administrator may differentiate from the respective testbed's administrator, we wanted to introduce a process that ensures the sanity of artefact catalogues. The envisaged scenario involved a facility administrator who leverages the facility's UI, deprived access or bypassing NODS, to make changes to the list of onboarded artefacts, hence rendering NODS catalogues outdated. This was tackled by implementing a polling mechanism that queries underlying facilities for their artefacts and eventually comparing them to the corresponding ones that are onboarded to the portal. If a misalignment is detected, the artefact's model is edited with an outdated status and prompts for an action, accordingly. Moreover, with the scope of facilitating the resource management in a more holistic way, a designated UI was created that assembles all the respective Virtual Infrastructure Managers (VIMs) of the registered testbeds.

The services portal UI withholds its position as the central entity that enables the design and hosting of the introduced triplet model. As two of the triplet's entities were adequately covered by D3.1, this report addresses major contributions introduced to the testing segment of the model. Indicatively, after a test is designed and packaged in the corresponding model at the designated UI, a services portal actor may import it and expect the translation of the test package into a Customer Facing Service Specification (CFSS) [4], namely the entity that comprises the triplet, through a seamless and automated process. These imported CFSSs that reflect the test cases bear a distinctive characteristic that differentiates them from the rest, i.e. a *testSpecRef*, that implies the link with a respective test specification.

The new addition of the updated overall portal, i.e. testing portal UI, addresses the need to design tests, bundle them in a standardized manner and render them available for execution. 5GASP's approach to define testing and validation of NetApps in terms of script testing was introduced in D5.1 [5], in the form of a test descriptor model. Whereas the descriptor's definition is not in the scope of the present document, its encapsulation within a standardized model that would describe the testing universally was initially addressed in D3.1. At this current version, we have supported the elected standardized model, namely TMF's Service Test Specification [6], with the corresponding implementation. Therefore, the recently introduced testing portal UI has included: i) a test designing a management interface, and ii) an executed test instance interface. The former provided a user-friendly UI that enables NetApp developer/Service designers to upload the aforementioned test descriptors and, sequentially, it automatically identifies deployment parameters of these descriptors by leveraging a specific placeholder. These placeholders imply that the value of the described parameters cannot be known beforehand and should be populated during deployment time, e.g. allocated IP addresses. Accordingly, an additional interface was implemented that enabled the overview of the executed test instances.

2.2.2 5GASP experimentation APIs service

The 5GASP experimentation APIs service received enhancements in both of its composed segments, i.e. VNF/NSD Artefact Management API and TMF family APIs. Figure 2-3 also illustrates the supported new features, namely TMF support for testing and the consumption of the latter by the testing portal.

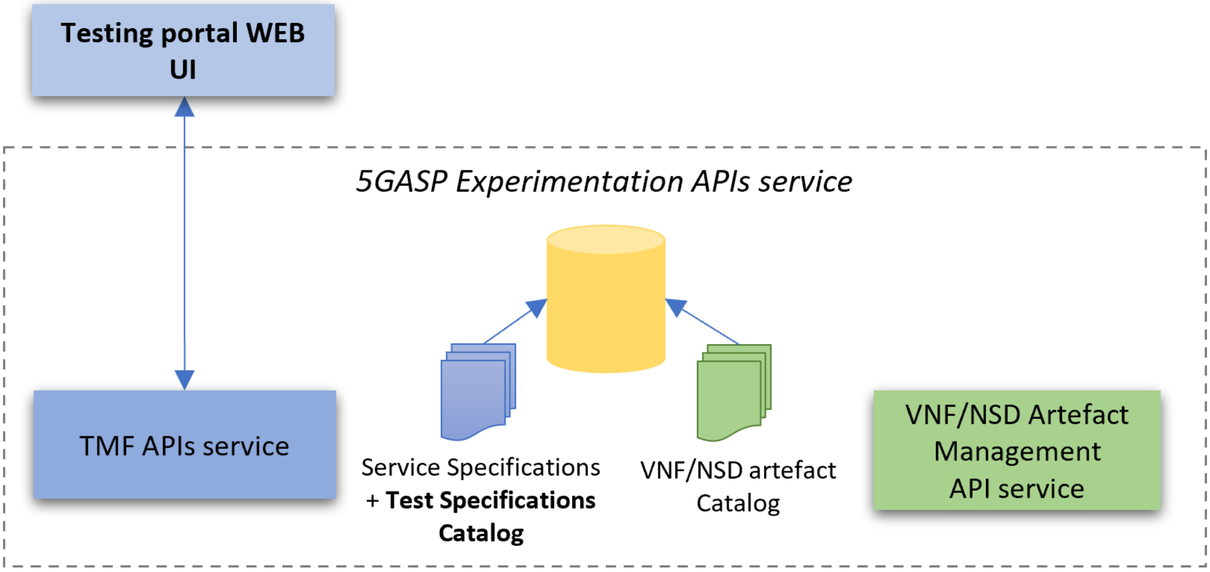


Figure 2-3: 5GASP experimentation APIs service components (updated)

The VNF/NSD management API service was augmented to meet the new requirements enforced by the NFV catalogue management, as described in the previous section. Specifically, it supports the improvements on the infrastructure listing and management (VIM and NFV artefacts' bidirectional synchronization) and provides the latest model that incorporates the outdated status of the artefacts.
On the other hand, TMF APIs service now supports TMF653 Service Test Management [6], simultaneously with the abundance of TM Forum's Open APIs that were presented in the previous document. Furthermore, the automated transformation process from Service Test Specification to Service Specification [4], as previously stated, is also designed and incorporated. Lastly, this newly introduced API is fully consumed by the latest addition of the portal, i.e. the testing portal UI.

2.2.3 5GASP Service Orchestrator

The 5GASP Service Orchestrator is the principal component of NODS architecture. It is tasked with order fulfilment tasks, deployment decisions to underlying facilities and the orchestration between components of the CI/CD automation circle. As already outlined, Service Orchestrator employs the open-source Flowable business process engine to outline its numerous orchestration schemes and cases. These schemes are expressed as Business Process Model and Notation (BPMN) diagrams, which are then consumed by the process engine to facilitate the described use case.
In the interval between WP3 deliverables, a major process has been added to the orchestration process that incapsulates the CI/CD automation circle, as illustrated in Figure 2-4.

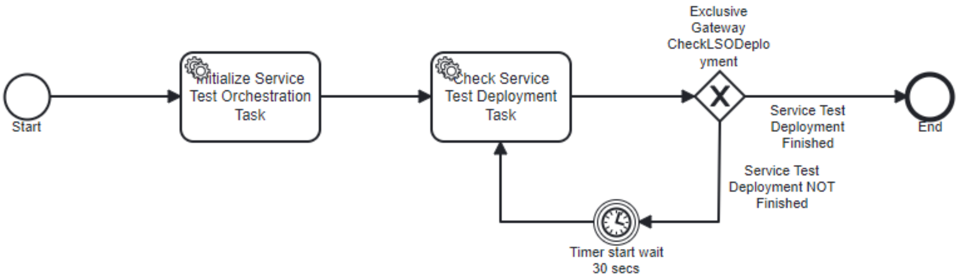


Figure 2-4: Testing orchestration pipeline

The above task is added to the pool of supported orchestration schemes that can be executed upon the receipt of a new deployment order, along with NFVO deployment, manual intervention, and external service provider deployment tasks, among others. This addition further enhanced the fundamental orchestration schemes enabling testing, thus providing a sturdy foundation to support a nominal end-to-end (E2E) automation circle. Though, it was already evident from the beginning of this venture that dynamic orchestration patterns should also be supported. These patterns could be injected into specific landmark phases within the Network Slice Instance lifecycle as defined by 3GPP [7]. Hence, several particular phases were introduced, i.e., pre-provision, after-activation, supervision, and after-deactivation. It briefly became prominent that another phase should be introduced that affects the creation of a service/slice itself, accepting requirements and policies that must be met before the instantiation process of a dependent service could emerge. The provided orchestration scheme implied that all related services of an order are executed in a parallel manner. Thus, specific envisaged use cases could not be captured, e.g. the NetApp should be instantiated, only after the hosting network slice is operating, or the successful deployment of a NetApp should trigger the respective testing pipeline. For that reason, a new phase was introduced, i.e. creation, as seen in Figure 2-5.

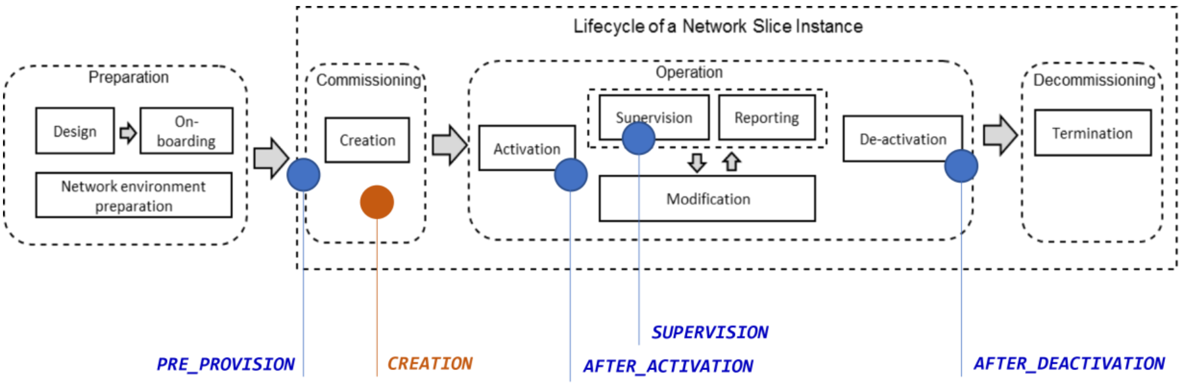


Figure 2-5: Updated landmark phases in relation to Network Slice Instance lifecycle (image from [7])

The proposed dynamic orchestration upon the above phases is materialized through the design of lifecycle management (LCM) rules. These rules are designed in a UI using Blockly [8], which generates code for interlocking blocks. Then, the output code is injected into the orchestration pipeline. Figure 2-6 presents such a rule for the creation phase. As an example, we can examine the creation of a complex service called Bundle C. Bundle C comprises of Bundle A, Bundle B, Service 4, Service 3. Bundle A comprises of Service 1, while Bundle B of Service 2, respectively. Prior to the addition of the creation rule, as soon as Bundle C order deployment order was captured, the overall bundle, i.e., Bundle A, Bundle B, Service 4, and Service 3, was simultaneously nominated for deployment. Although, the illustrated rule proposes that only Bundle A and Service 3 will be deployed during the initial phase. Therefore, only after Bundle A successfully deploys, the orchestration of Bundle B can be initiated and eventually the same principal applies to Service 4, after Bundle B is deployed. The above example is related to the dependencies that need to be met in the deployment of the triple, as well (see Section 4, triplet's successive execution).

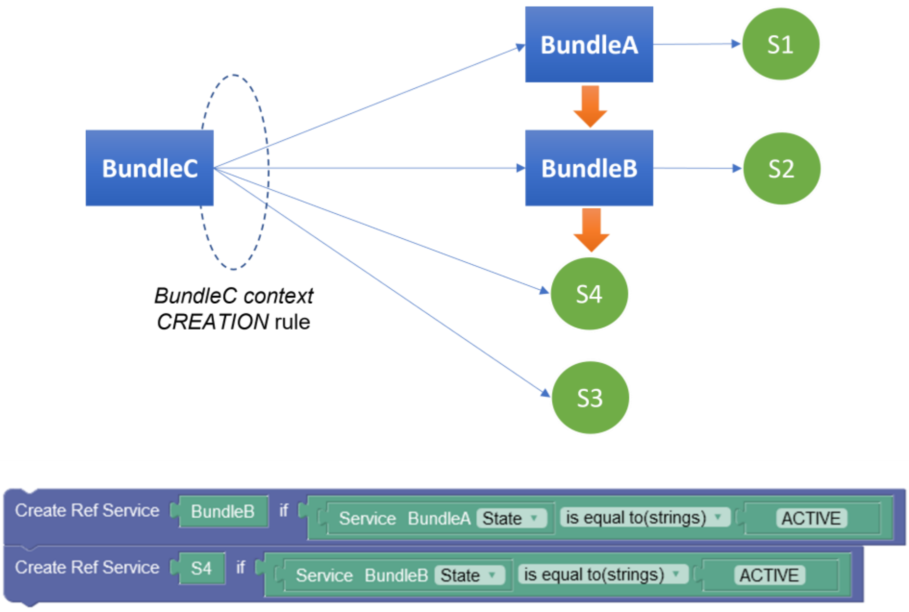


Figure 2-6: Creation LCM rule example.

To sum up, the fulfilment of the newly introduced landmark phase within the supported orchestration schemes is made apparent in Figure 2-7.

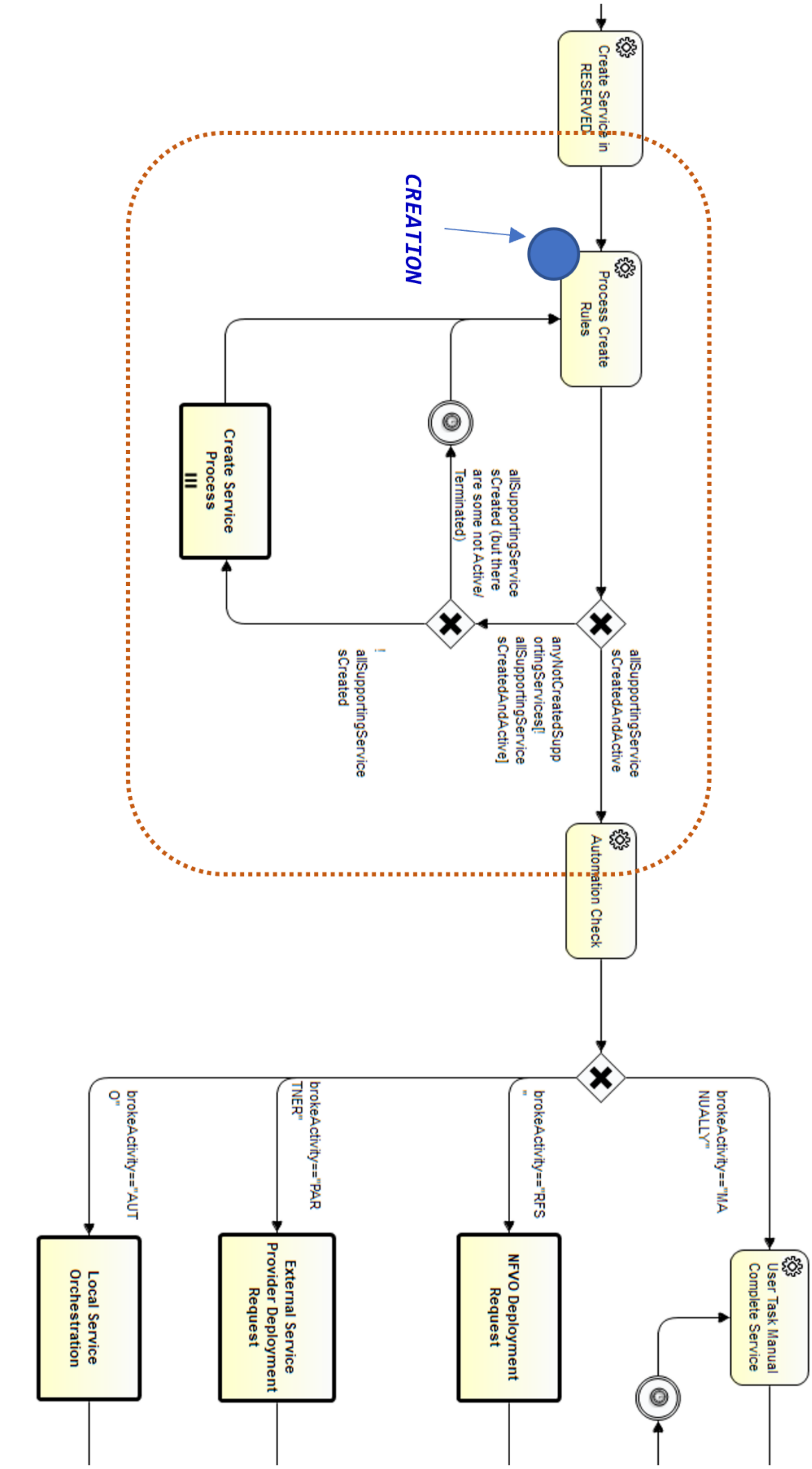


Figure 2-7 Creation phase landmark orchestration scheme

2.2.4 5GASP Network Orchestrator

The Network Orchestrator (NetOr) is a complex system composed of multiple micro-services, responsible for providing the interdomain scenario in 5GASP. By combining a micro-service-oriented architecture with an event-driven approach, this system provides scalability, flexibility, modularity, and efficiency requisites. This 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. This service will act as the Communications Service Management Function (CSMF) and communicate with lower-level Network Slice Management Functions (NSMFs). NetOr is then responsible for creating and configuring a Virtual Private Network (VPN) tunnel between the independent administrative domains, creating a secure communication channel.

Additionally, to achieve the multi-domain scenarios, not only do the network resources need to be aligned with the defined mechanisms 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. Thus, NetOr is the service serving as that centralized service orchestrating agent, gathering, and redistributing the inter-domain information.

Taking into consideration Deliverable D3.1 [2], a considerable number of improvements have been made since. For instance, the resilience of NetOr has been enhanced as it has a better capability of recovering from failures that may occur, for example, a crash of an existent peer on a communication channel will be handled by NetOr assuring that the connection between the remaining peers remains intact. Besides this, the required configuration to use NetOr's functionalities has also been refined allowing the instantiation and management of Tunnel Peers in a smoother and more automated manner. Finally, new features have been added to this service that may be useful in the maintenance and management of all tunnels, which are the following:

- **Manage Routes Configuration:** Removing and Adding routes can now be performed through NetOr, taking as parameters the Network and the Gateway that will be used on the route.
- **Get Network Interfaces:** Information about the Network Interfaces on the VNF can be retrieved such as the IP or the MAC (Media Access Control) Address.
- **Delete a Peer:** Removing a Peer from an existing tunnel connection may be necessary in some situations.
- **Update Peer Information:** Peers' information such as the Endpoint IP, the allowed IPs to connect to the Peer in cause can be updated.

2.2.5 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. The information exchanged relies on ETSI SOL005 interface [9] and thus, several corresponding plugins charged with the translation of resource requirements into lifecycle actions in the MANO domain are provided. At present, plugins that support OSM (version 8, 9, 10), as well as a generic SOL005 version are implemented. During the elapsed interval between the previous version and the present document, OSM version 11 [10] was released and is supported by NODS. Furthermore, several minor alterations were introduced within the aforementioned plugins, which enabled the support of the newly presented features of Section 2.2.1 and 2.2.2, i.e. bidirectional synchronization, enhanced artefact's model.

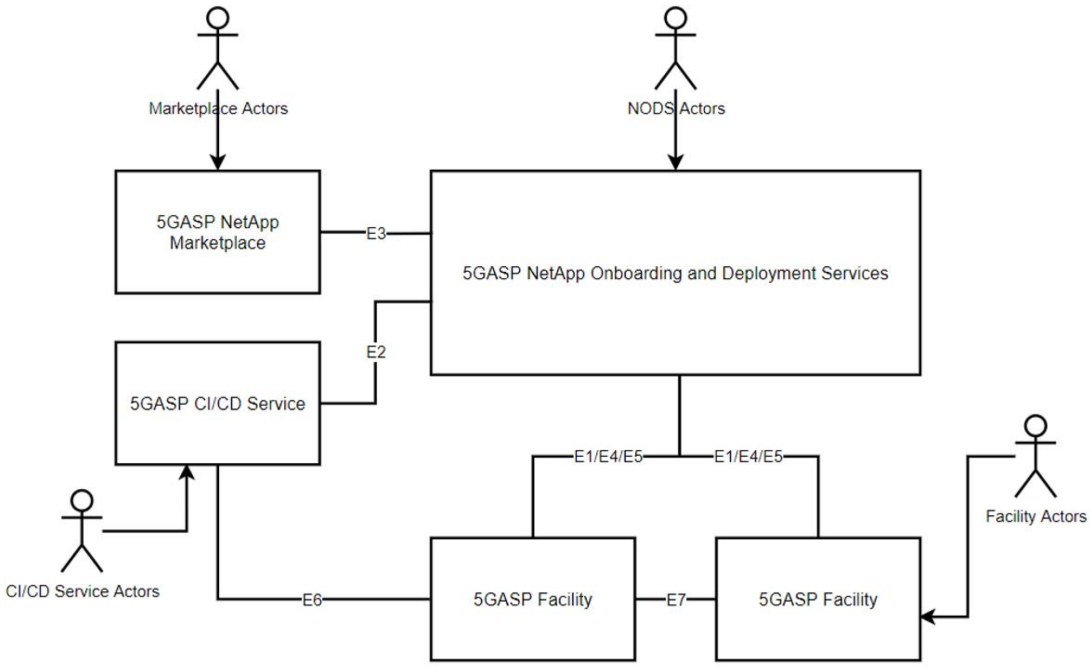
2.3 Summary table with current status

Following Table 2-1 summarises the status of the services developed for the 5GASP platform.

Service Name	**Design**	**Development**	**Test**	**Integration**
NetApp onboarding and experiment management portal	Completed	Completed	Completed	Completed
5GASP experimentation APIs service	Completed	Completed	Completed	Completed
Service registry	Completed	Completed	Completed	Completed
Authentication service	Completed	Completed	Completed	Completed
Issue management service	Completed	Completed	Completed	Completed
Central logging service	Completed	Completed	Completed	Completed
5GASP Service Orchestrator	Completed	Completed	Completed	Completed
5GASP Network Orchestrator	Completed	Completed	Completed	Completed (partially)
MANO Client APU service	Completed	Completed	Completed	Completed
Microservices bus	Completed	Completed	Completed	Completed

Table 2-1: Status of 5GASP services

3 5GASP ecosystem and the interfaces



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 3-1: Reference figure

3.1 Interaction with CI/CD - Interface E2

The interaction between NODS and the CI/CD Service, i.e. CI/CD Manager, is highlighted in the previous version of this document, in terms of its extensive design, architecture and the data models that are exchanged, thus it is not reproduced. Next, the interaction between the CI/CD Manager and 5GASP facilities' CI/CD entity, namely the *CI/CD Agent*, is also substantially covered in D5.1 [5]. Shortly, The CI/CD Manager is the central entity that coordinates all the testing and validation tasks, creating and distributing the respective tasks to the CI/CD Agents deployed in the testbeds. To boost the interoperability of the CI/CD Manager, it implements the TMF 653 specification [6], which is the base of all interactions with the NODS.
The above design patterns and concept were implemented, integrated and extensively tested in the 5GASP ecosystem by Q1 of 2022. The CI/CD pipeline is triggered by the NODS with the submission of a TMF 653 Service Test payload to the CI/CD Manager, as presented in Figure 3-2. Specifically, the Service Test payload, submitted by the NODS, comprises all the information needed to commence a new testing and validation task. Some information is directly specified in the initial payload, while other is referenced through URLs, from where the information can be gathered. For instance, the initial payload is composed of a collection of

test characteristics that solely by themselves are not sufficient to perform the validation task. These characteristics are defined by the NODS and comprise information such as the IPs of a NetApp's VNFs. Without the onboarded Testing Descriptor, it is impossible to make any sense of the submitted characteristics. Thus, the initial payload also includes an URL from which the Testing Descriptor can be obtained. The CI/CD Manager uses this information to obtain the Testing Descriptor, which will then be rendered using the Service Test characteristics. Only after this process, the CI/CD Manager will be able to generate a validation pipeline configuration, which guides the entire testing process.

It is also worth mentioning that the Service Test Payload may also include URLs to collect developer-defined tests, should the NetApp developer onboard its own tests. If so, the CI/CD Manager will assemble these tests and make them available to the CI/CD Agents so that they can eventually execute them.

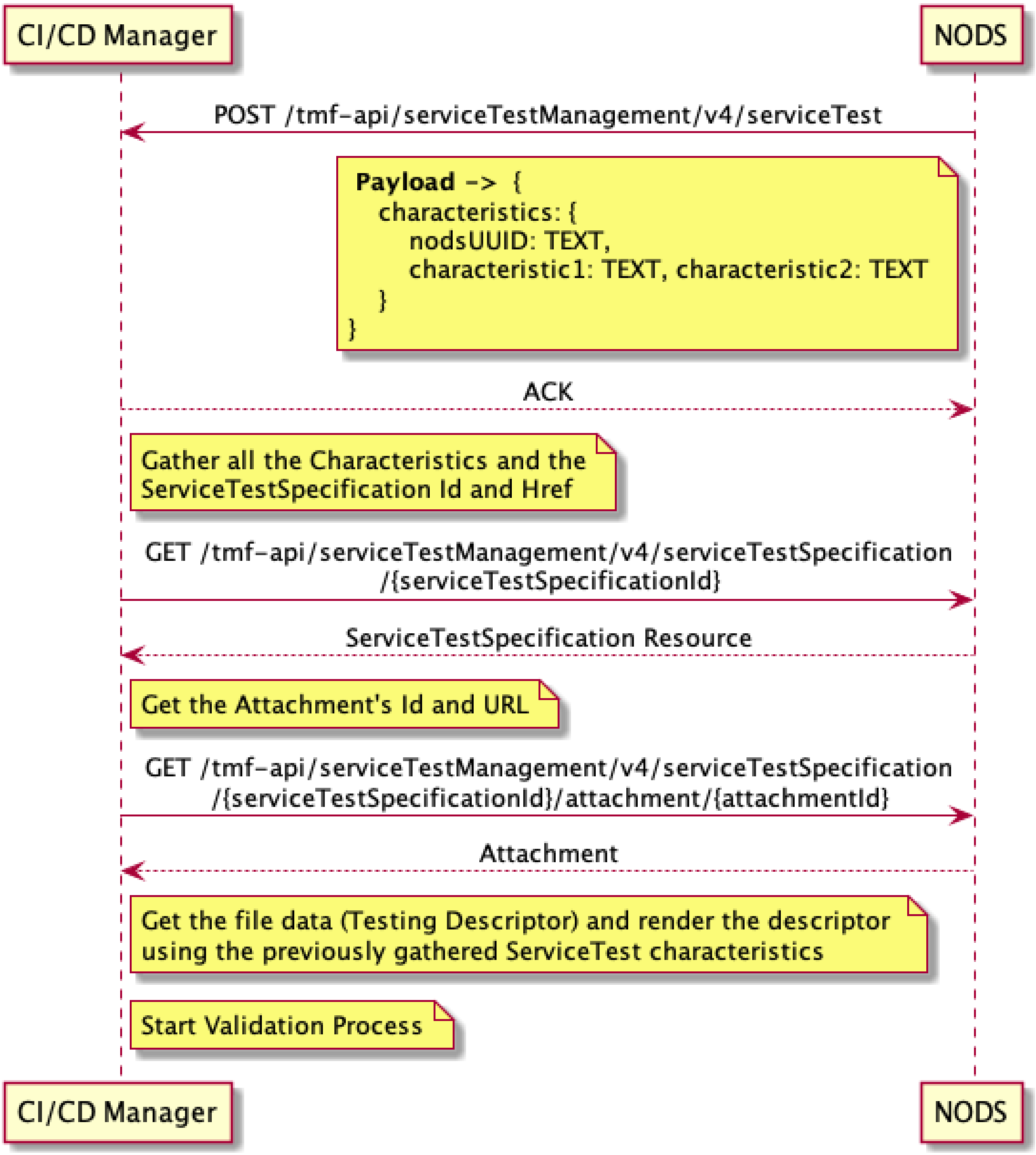


Figure 3-2: NODS - CI/CD Manager interaction

3.2 Interaction with Facilities - Interfaces E1/E4/E5/E7

3.2.1 Communication with the NFVOs - Interface E1

In the previous version of this document, we identified four available options regarding the interconnectivity between NODS and the underlying facilities. These options are as follows:

- Option A: Single OSS portal on each facility site
- Option B: NODS as the E2E Service Orchestrator
- Option C: Non ETSI-NFV APIs
- Option D: ETSI-NFV or 3GPP compliant APIs over other management services

Further details on the above options can be found in D3.1 [2], thus are omitted. Taking into consideration the technological stacks deployed in each facility, and the available interfaces exposed, we have elected Option B as the most convenient. The majority of the facilities employed OSM as its selected NFVO, thus exposing a SOL005 compliant Northbound Interface (NBI). As already noted in Section 2.2.5, NODS incorporates a relative SOL005 compliant client, thus rendering the direct exposure of facilities' OSMs as the nominated scenario, as Figure 3-3 illustrates.

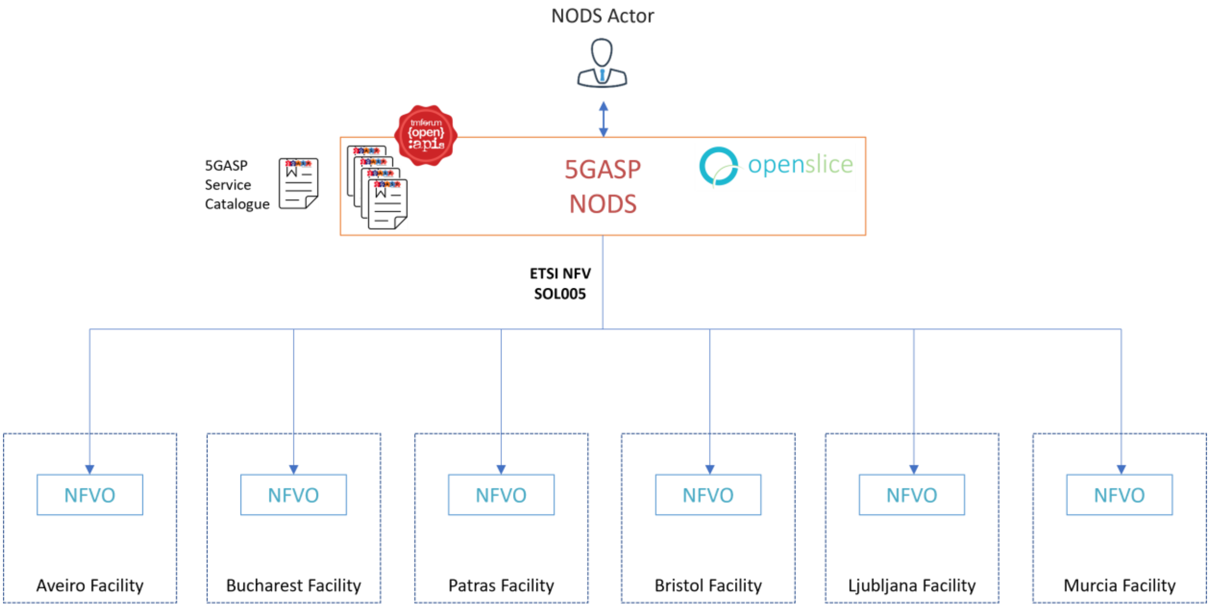


Figure 3-3: NODS - facilities interaction

3.2.2 Cross Domain Network Orchestration - Interface E4

D3.1 [2] introduced Interface E4, as the Interface for Cross Domain Network Orchestration. This interface is mainly used by ITAv's Network Orchestrator to deploy testbed-level wireguard servers, that will enable an interdomain scenario in 5GASP, thus its utmost importance. Until this deliverable's submission date, several experiments have been performed, which allowed for additional requirements definition, targeting the achievement of a zero-touch orchestration of the inter-testbed VPN tunnels.
To support this scenario, each testbed must reserve a static private IP, that shall be assigned to the Wireguard [11] peer located on premises. Besides this, the testbeds must also provide

a public IP to be used to publicly advertise their Wireguard server, taking into consideration that the traffic incoming and outgoing the Wireguard server cannot be NAT'd.
With these mechanisms in place, it will be possible to dynamically orchestrate and operate the inter-testbed VPN tunnels, without the intervention of the testbed's owners.
Thus, some testbed-specific milestones are defined:

- Allocation of a static private IP, to be assigned to the testbed's Wireguard peer;
- Allocation of a static public IP, to enable communication between peers located on different testbeds;
- Ensure the Wireguard's incoming and outgoing traffic is not being NAT'd;
- Addition of static routes, to the default testbed's network gateway. This will enable the transparent routing of packets destined for the other testbeds.

3.2.3 Facility and testing management - Interface E5

At present, 5GASP has completed the CI/CD automation incorporating script tests, both pre-defined and developer-defined. Extensive details on the introduced testing architecture can be found in D5.3 [12]. Thus far, the executed testing, i.e., script testing, solely employs interface E2 as scripts are uploaded in the CI/CD Manager and then forwarded to the respective CI/CD Agents through interface E6 [5]. During the next testing iteration, we expect to incorporate the supplementary testing option, i.e., tool testing. Consequently, the described interface is reserved for the aforementioned case and shall be validated for its functionality in the following months.

3.2.4 Inter-facility connectivity - Interface E7

Following the orchestration and deployment of the Wireguard [11] VPN tunnels (Interface E4), Interface E7 will be used to provide inter-testbed communication. So far, we have achieved interdomain communication between ITAv's, UoP's and OdinS' testbeds, thus enabling the deployment of interdomain NetApps over these domains. Until 2022 Q3, we expect to have interdomain communication between all 5GASP's testbeds. This will be achieved with the orchestration of a Wireguard VPN mesh network.

3.3 Interaction with NetAppStore - Interface E3

This interface is charged with publishing new NetApps to the NetAppStore, as soon as a CI/CD automation circle that awards a certification [13] is successfully completed. Both the enabling API and the resource model, i.e., TMF's Product [14], is defined and developed.
Once the NetApp Marketplace is organized and deployed, the E3 interface will be used to display NetApps and connect the Marketplace to other onboarding and deployment services. We have now achieved a stable transfer of information about NetApps from Marketplace to our interface, from where, in the future, they can be used in onboarding and deployment services.
Data is fed to the E3 interface, using the REST API, in the form of JSON. On the E3 side, they look like this:

```
{
  '@type': string
```

```
description: string
id: number
isBundle: boolean
lastUpdate: string
lifecycleStatus: string
name: string
uuid: string
version: string
}
```

3.4 Summary of the interfaces development

The following Table 3-1 summarises the status of the interfaces as shown in Figure 3-1

Interface Name	Description	Design	Development	Test	Integration
E2	Completed	Completed	Completed	Completed	Completed
E3	Completed	Completed	Completed	In progress	Planned
E1/E4/E7	Completed	Completed	Completed	Completed	Completed
E5	Completed	Planned	Planned	Planned	Planned

Table 3-1: Status of the interfaces

4 5GASP NODS Implementation Release 2

This section is dedicated to the implementation aspects of the additions described in the previous sections. It incorporates the updates on the prototype platform delivered for MS3.1, together with the newly introduced elements, such as testing support.

To begin with, the capability to manage the underlying facilities, offered to NODS Platform Administrator, was augmented with the addition of synchronization option, as seen in Figure 4-1. The figure illustrates the overall 5GASP facility, with all its testbeds incorporated, while the rest of the management capabilities remain unchanged.

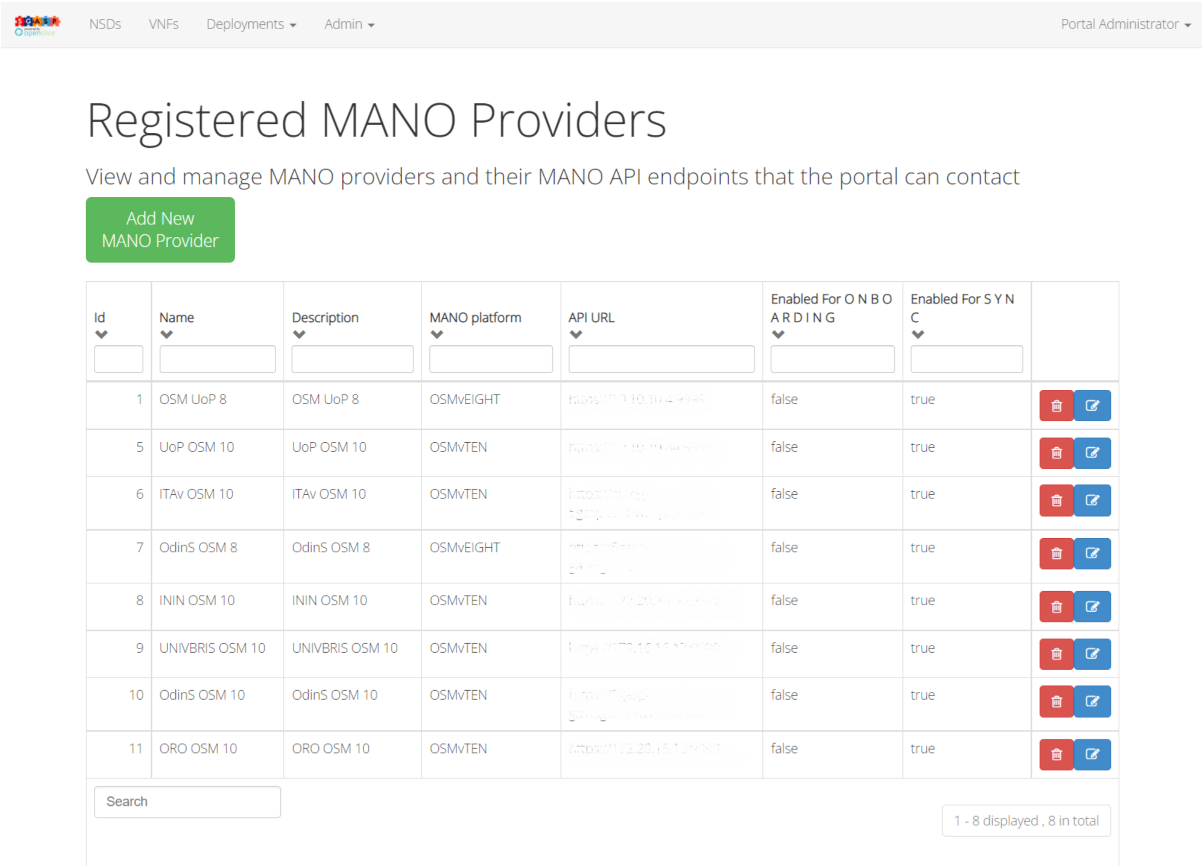


Figure 4-1: Facility management UI (updated)

The dedicated interface that was employed to reflect the onboarded VNF/NSD to relative actors is also updated to support the synchronization model. As seen in Figure 4-2, a new onboarding status is added, i.e. *OSM_MISSING,* that represents an artefact that was indeed onboarded or synchronized but currently is not present in OSM's artefact list, thus it cannot be expected to be available for deployment.

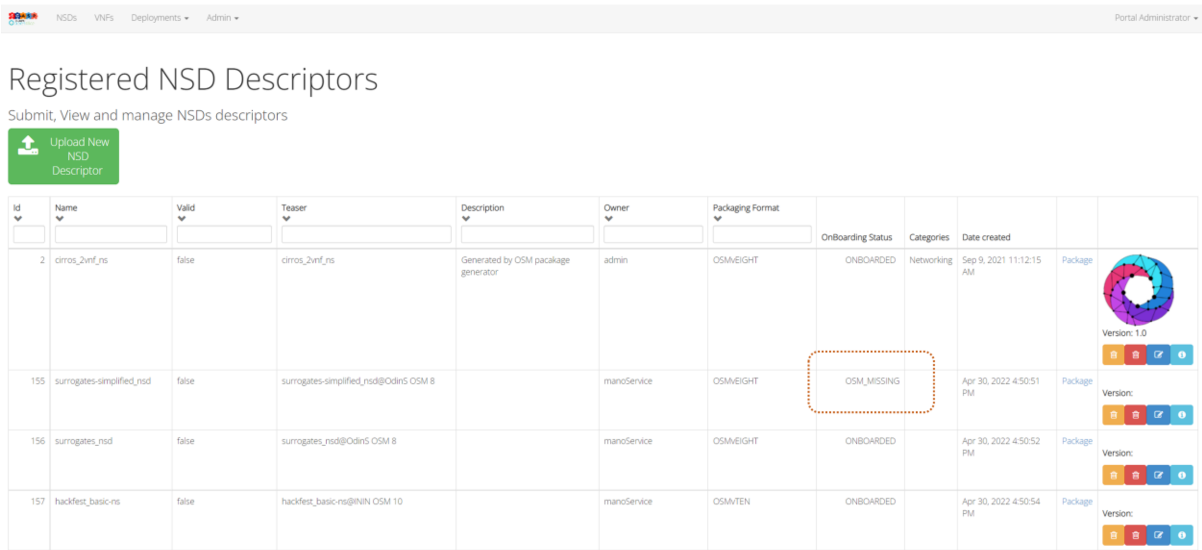


Figure 4-2: Onboarded VNF/NSD artefacts listing UI (updated)

Regarding the NFV catalogue management UI, the last alteration can be indicated with the addition of facilities' infrastructure. As previously mentioned, the UI created assembles the CIMs of the registered facilities and can be seen in Figure 4-3.

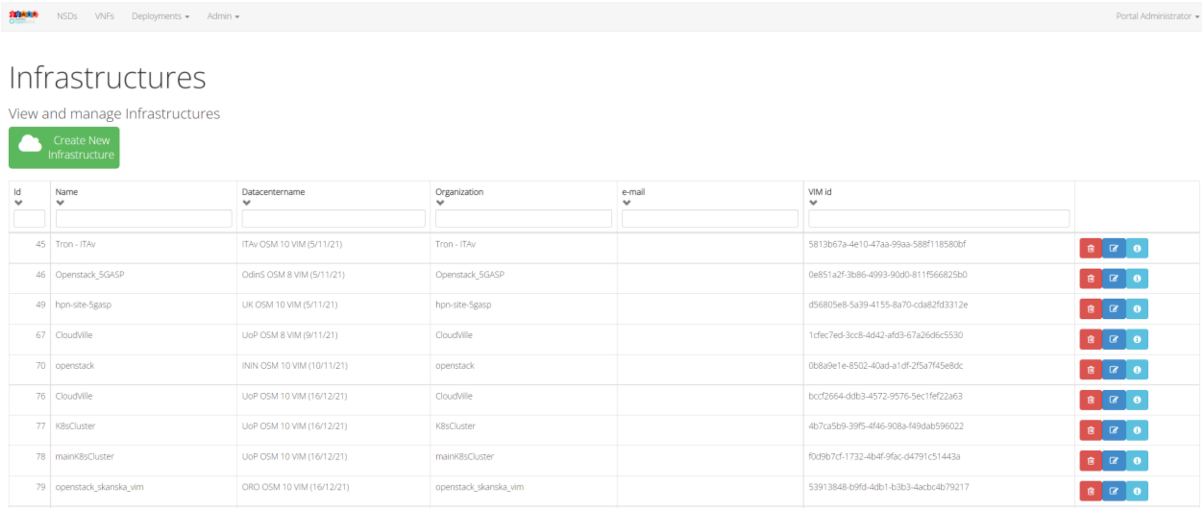


Figure 4-3: Managed infrastructure UI

Per Section 2.2.1, the overall NODS portal integrated a testing portal UI to address the need of test designing, YAML descriptors' uploading and test execution overview. Therefore, a designated UI is available for designing the tests, along with the capability to support the onboarding of the relative test descriptors and the automatic extraction of the parameters that should be filled during execution time within these descriptors (Figure 4-4). As expected, further complementing the test design overview, a list is offered with all the available designed tests (Figure 4-5). Lastly, an additional overview is offered for the running instances of the tests, as they are executed as a part of an ordered CI/CD process (Figure 4-6).

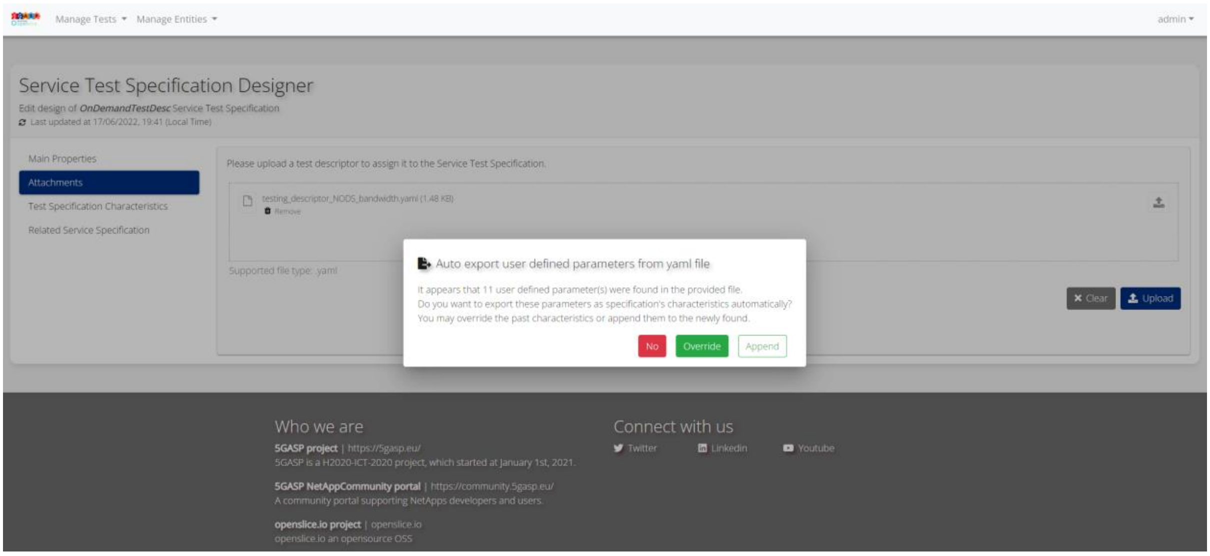


Figure 4-4: Test specification design UI

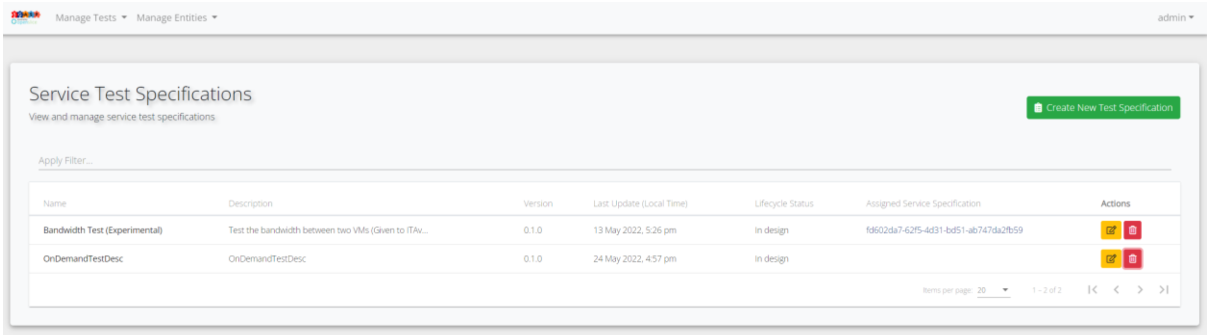


Figure 4-5: Test specification listing UI

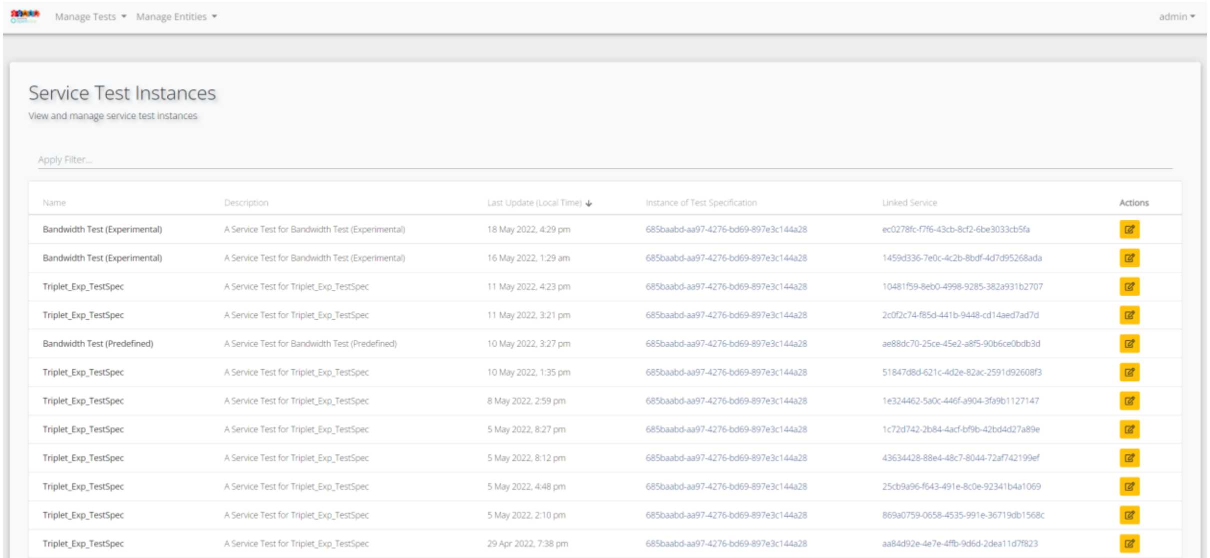


Figure 4-6: Test instances listing UI

Regarding the services portal UI, the hosting network slice deployments are available for ordering in all facilities and exposed through the respective catalogues, as Figure 4-7 illustrates.

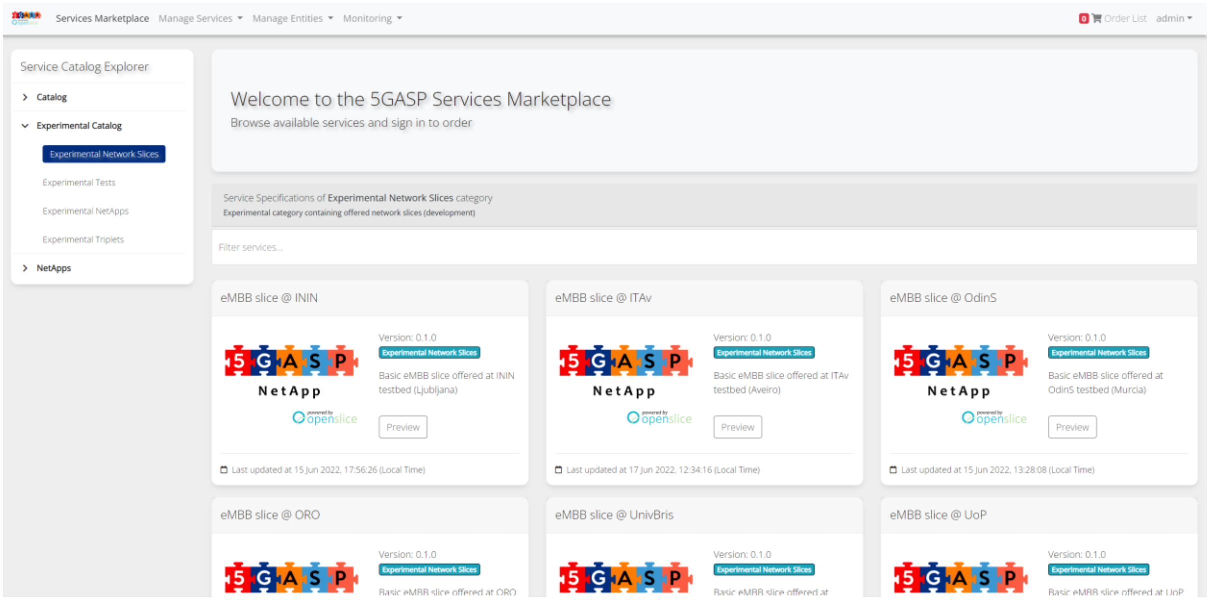


Figure 4-7: eMBB slice offering in facilities

Moreover, an experimental triplet deployment is showcased in Figure 4-8, that employs an eMBB slice and a prototype NetApp deployment in the UoP facility, followed by the execution of a prototype bandwidth test among the deployed NetApp's components. The triplet's elements are executed in a successive order, i.e. hosting slice deployment leads to exposure of UPF's data path, followed by NetApp deployment that generates the deployment IPs of its constituent VNFs, which are eventually handed over to CI/CD Manager to conduct the specified test. The aforementioned deployment order, implemented by the rule engine (see Section 2.2.3), is illustrated in Figure 4-9. Once the test case is successfully conducted a URL is made available through the UI (Figure 4-10) that leads to the test results report (Figure 4-11).

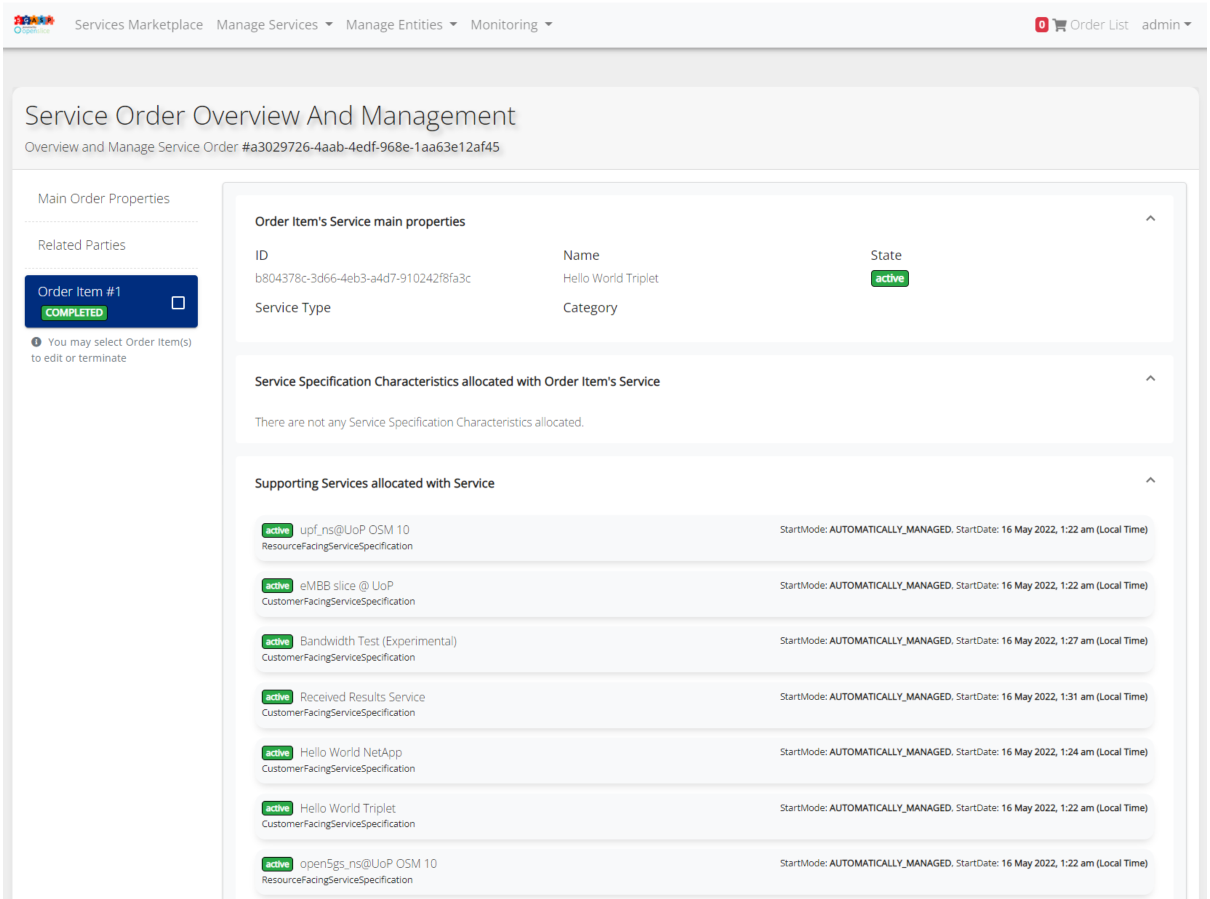


Figure 4-8: Experimental triplet deployment

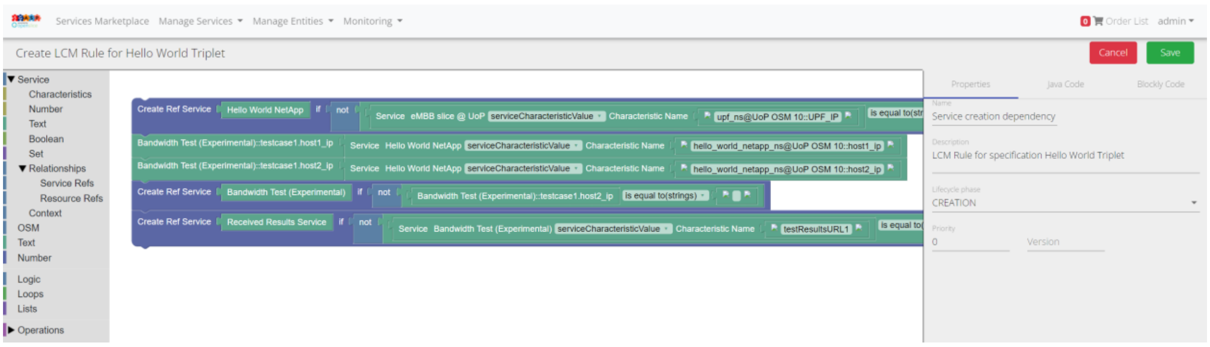


Figure 4-9: Deployment order rule design

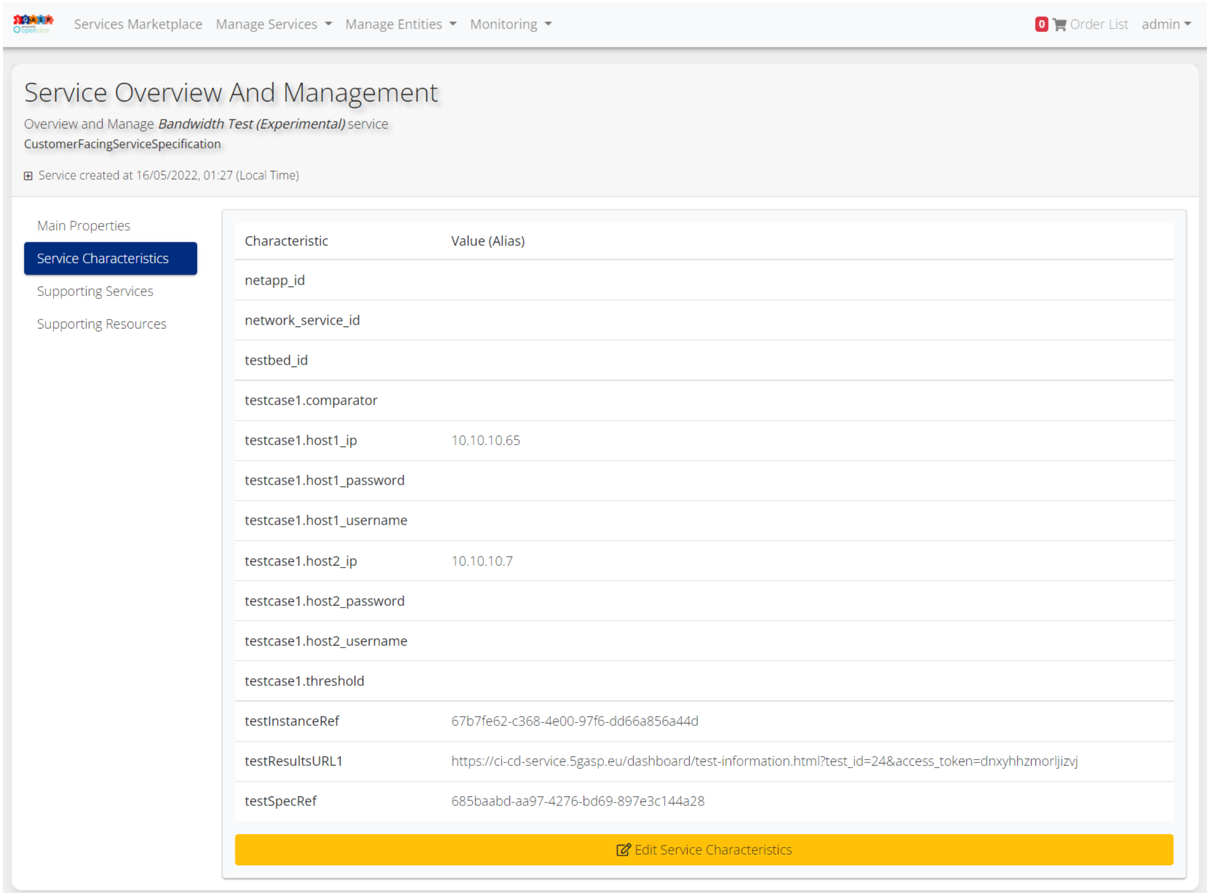
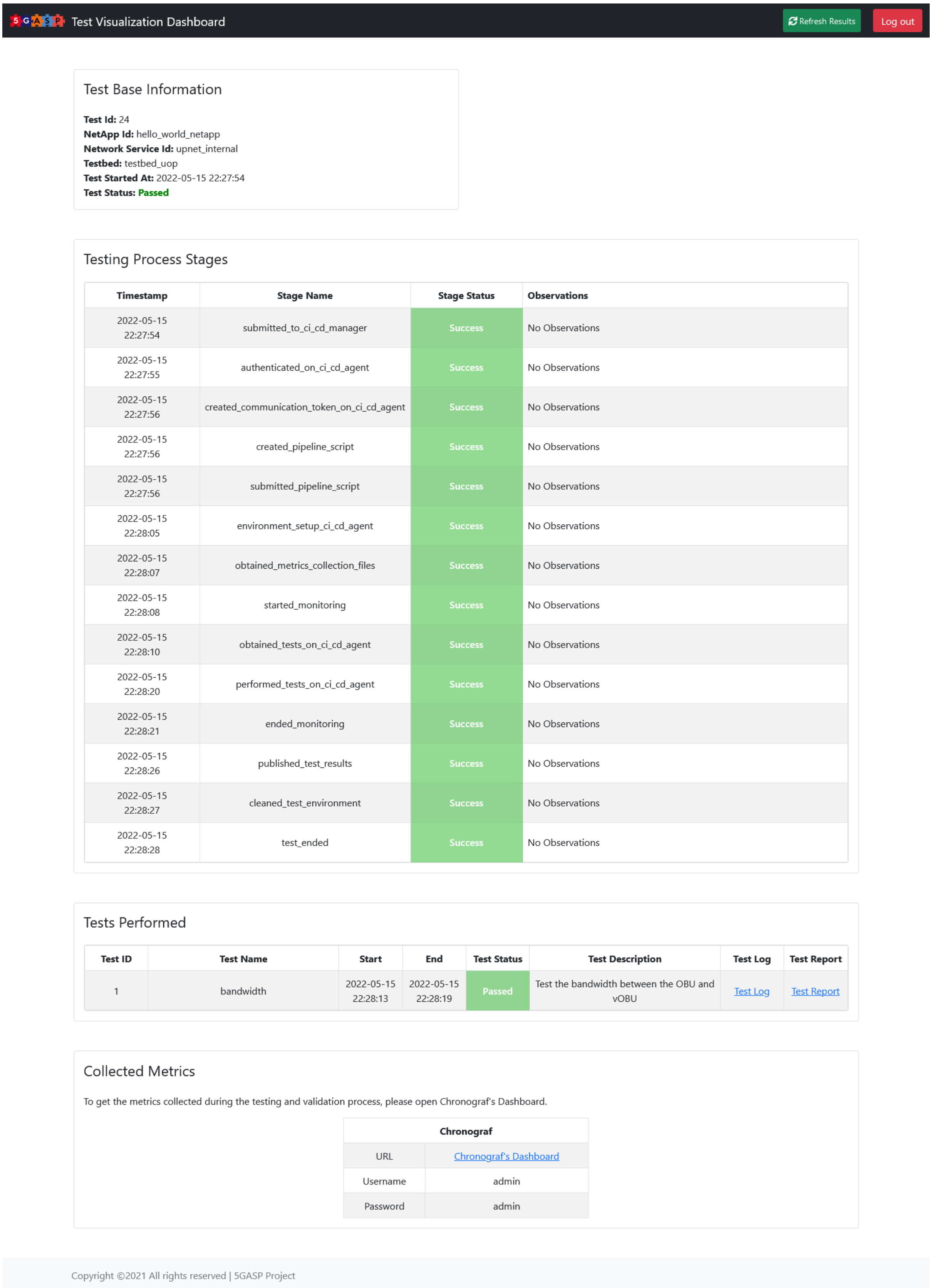


Figure 4-10: Test results URL exposure



5GASP Test Visualization Dashboard

Refresh Results

Log out

Test Base Information

Test Id: 24
NetApp Id: hello_world_netapp
Network Service Id: upnet_internal
Testbed: testbed_uop
Test Started At: 2022-05-15 22:27:54
Test Status: Passed

Testing Process Stages

Timestamp	Stage Name	Stage Status	Observations
2022-05-15 22:27:54	submitted_to_ci_cd_manager	Success	No Observations
2022-05-15 22:27:55	authenticated_on_ci_cd_agent	Success	No Observations
2022-05-15 22:27:56	created_communication_token_on_ci_cd_agent	Success	No Observations
2022-05-15 22:27:56	created_pipeline_script	Success	No Observations
2022-05-15 22:27:56	submitted_pipeline_script	Success	No Observations
2022-05-15 22:28:05	environment_setup_ci_cd_agent	Success	No Observations
2022-05-15 22:28:07	obtained_metrics_collection_files	Success	No Observations
2022-05-15 22:28:08	started_monitoring	Success	No Observations
2022-05-15 22:28:10	obtained_tests_on_ci_cd_agent	Success	No Observations
2022-05-15 22:28:20	performed_tests_on_ci_cd_agent	Success	No Observations
2022-05-15 22:28:21	ended_monitoring	Success	No Observations
2022-05-15 22:28:26	published_test_results	Success	No Observations
2022-05-15 22:28:27	cleaned_test_environment	Success	No Observations
2022-05-15 22:28:28	test_ended	Success	No Observations

Tests Performed

Test ID	Test Name	Start	End	Test Status	Test Description	Test Log	Test Report
1	bandwidth	2022-05-15 22:28:13	2022-05-15 22:28:19	Passed	Test the bandwidth between the OBU and vOBU	Test Log	Test Report

Collected Metrics

To get the metrics collected during the testing and validation process, please open Chronograf's Dashboard.

Chronograf	
URL	Chronograf's Dashboard
Username	admin
Password	admin

Figure 4-11: Test results report

5 5GASP multi-domain NFV fabric

5.1 Design

In section 2.2.4, the concept and goals of the Network Orchestrator have been introduced. However, some recent details about the Design and mechanisms of this Service were added/updated and are addressed in this subsection.
Concerning the 5GASP environment, NetOr will be responsible for deploying and orchestrating a mesh network between all testbeds. In other words, a Network Topology in which all testbeds' peers are connected directly and can efficiently route data to and from each other. The main goal behind this implementation decision is to provide a unified view of all 5GASP testbeds. Moreover, to achieve this scenario, P2P (Peer-to-Peer) VPN connections between all peers are used since, opposed to other topologies like Client-Server, the load is distributed for each peer, avoiding that a single entity handles everything.
Figure 5-1 shows the representation of the multidomain scenario.

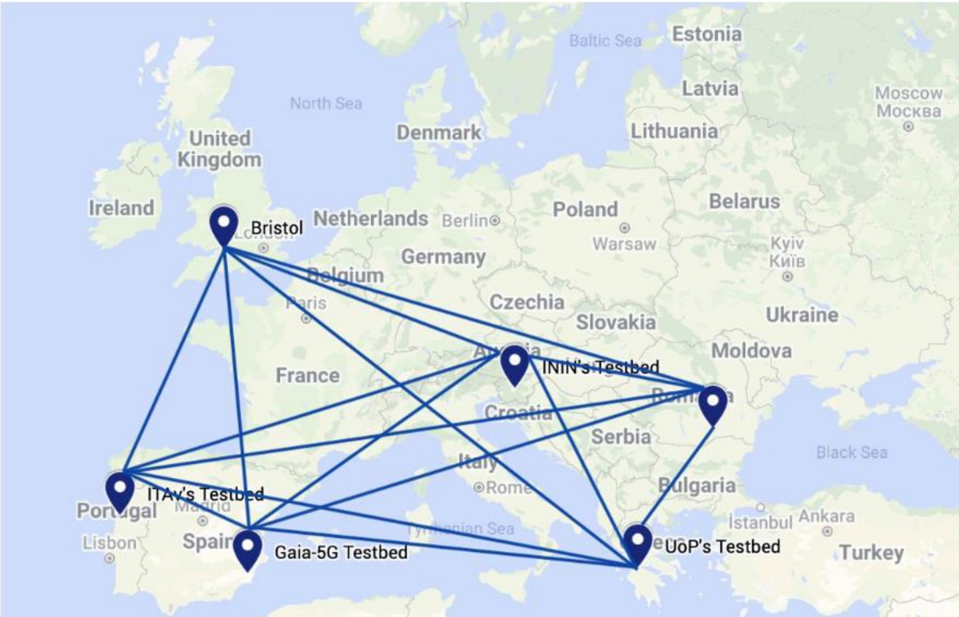


Figure 5-1: Representation of the multidomain scenario

To provide the above-mentioned scenario, NetOr creates an overlay of Wireguard [11] VPN Tunnels. Further in this document, Wireguard will be described as well as the choice of using this technology. Figure 5-2 shows the interaction between NetOr, each domain's NFVO and the involved Wireguard Peers. The process starts with the Tunnel Configuration action that is triggered by NetOr once the instantiation of a new inter-domain NS is requested. At this point, NetOr will handle the creation of a VNF through each domain's OSM, so that the tunnel may be established later. Consequently, each VNF will send information to NetOr, such as endpoints and public keys, and receive similar data from each peering VNF. After the tunnel configuration, it is possible to perform Day 2 Operations such as performance characteristics modification, Bandwidth for example, or the addition or removal of tunnel peers.

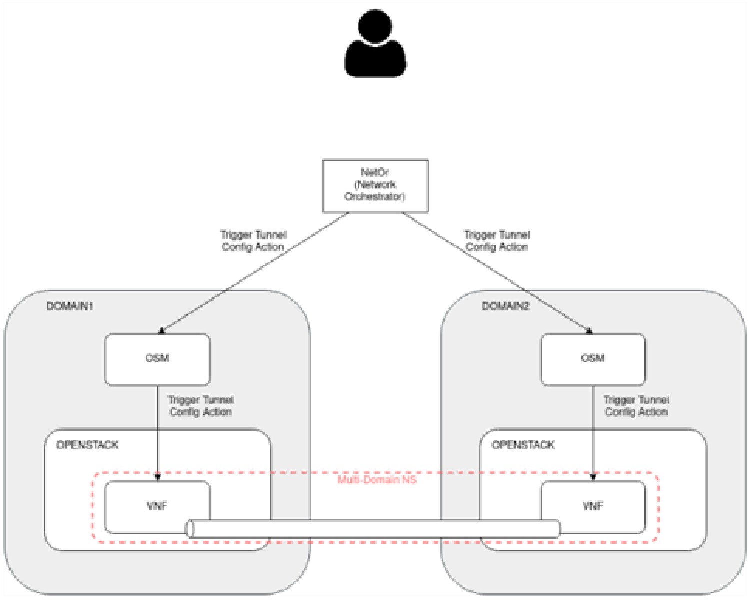


Figure 5-2: Proposed Interconnection Architecture

This process can be seen in Figure 5-3 from a more detailed point of view. Thus, as presented, 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 the instantiation, continuous polling is done over the status and information of its components. When 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, as mentioned previously. Once all the Service components are "running" and their tunnel information has been fetched, NetOr proceeds to exchange that information between peers, effectively providing the necessary data to peers and configuring the tunnel in the process. Only NetOr knows all tunnel peers since it instantiated them independently on different domains.

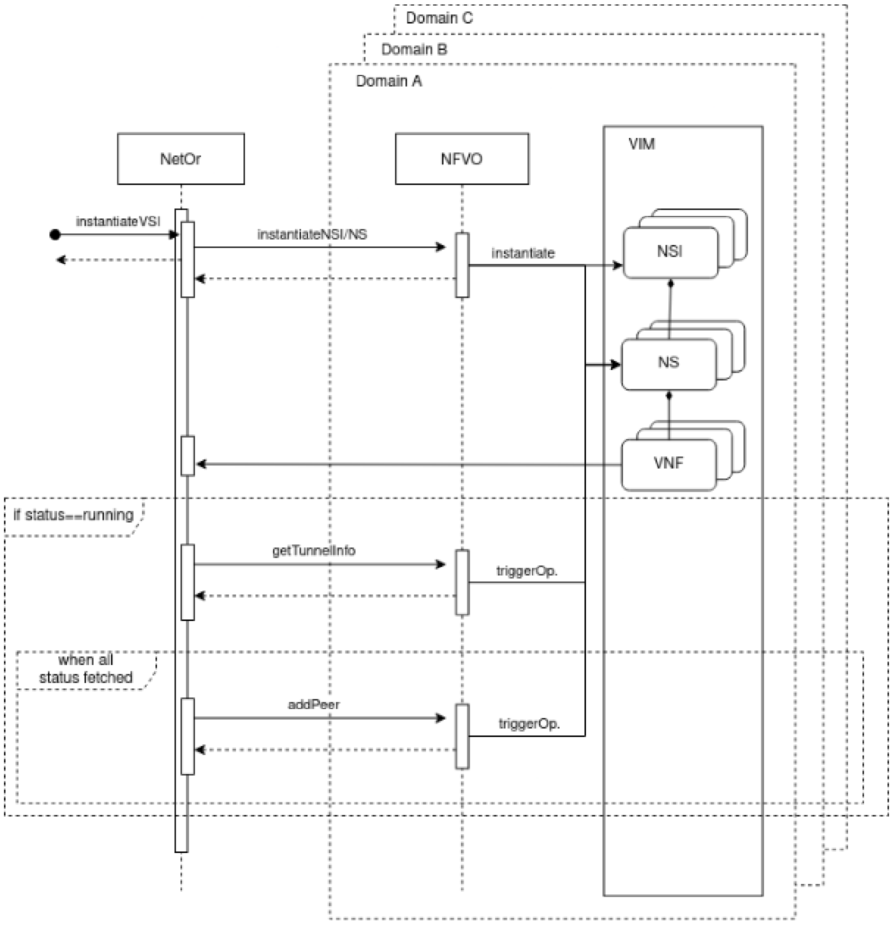


Figure 5-3: Data flow between NetOr and respective domains

5.2 Underlying technologies and security considerations

Concerning the Network Orchestrator and the inter-domain scenario, as mentioned in the previous section, Wireguard [11] is the technology used to provide the VPN Tunnels between the testbeds. Wireguard is the newest and fastest known tunnelling open-source protocol. It uses state-of-the-art cryptography that is already presented in other VPN options, such as OpenVPN and IPsec/IKEv2. Additionally, as it is a lightweight software, the attack surface is minimal: Wireguard has been implemented with very few lines of code and in a straightforward way, which makes it easier for Security Researchers to audit it and consequently find security vulnerabilities. Besides this, its ease of deployment and agility, combined with all the previously mentioned characteristics, allows this tool to have a panoply of advantages that are simply not provided by other VPN tools.
Despite all the positive factors just mentioned, some aspects need to be addressed regarding Security. In section 5.1, we mentioned that a mesh Network of Wireguard Peers was being used to offer the inter-domain scenario on 5GASP, and consequently, offer a unified view of all testbeds involved. However, the existence of inter-testbed VPN tunnels significantly increases the attack surface of a testbed, since if a single testbed is breached, all remaining testbeds may be affected. As a suggestion to mitigate this, Traffic Filtering and Intrusion Detection and Prevention Systems may be applied to the Wireguard Peers. Through this method, it is possible,

for example, to exclude all unnecessary traffic and consequently, hinder an attacker's actions, mitigating the exposure of the testbeds.

6 Implemented Capabilities from 5GASP facilities

6.1 Implementation to support the multi-domain NFV fabric

This section will describe the status and in brief the design of the multi-domain connectivity between the testbeds in 5GASP ecosystem. It will also update the readiness status of the testbeds to enable NetApp testing.

6.1.1 Implementation of E1 interface

As described in Section 3.2.1, all 5GASP facilities have elected Option B, i.e. direct exposure of OSM, as the way to communicate with NODS. The latter requires an undisrupted communication channel as for the synchronization scheduler to operate properly, keeping the artefact list updated between the portal and the underlying facilities. In addition, any triplet orchestration order by the respective actor must be considered as an asynchronous task to minimize the need of manual intervention. All the above, show the need to assure stable and uninterrupted VPN connectivity, which was achieved by fully automated recovery scripts that eliminate downtime. The following Table 6-1 incorporates the updated interface implementation per facility.

Facility	E1 Interface option	Exposed service	Interconnection option
Aveiro	Option B	NFVO (OSM) Version: 10	Firewall rule
Patras	Option B	NFVO (OSM) Version: 10	VPN (SSL)
Bristol	Option B	NFVO (OSM) Version: 10	VPN (SSL)
Bucharest	Option B	NFVO (OSM) Version: 10	VPN (SSL)
Murcia	Option B	NFVO (OSM) Version: 10	Firewall rule
Ljubljana	Option B	NFVO (OSM) Version: 10	VPN (SSL)

Table 6-1: E1 Interface implementation per facility (updated)

6.1.2 Implementation of facilities interconnection

So far, as mentioned in Section 3.2.4, an inter-domain scenario has been achieved between three facilities: ITAv, UoP and OdinS. However, at the moment, the scenario of a Zero-Touch Deployment of the Wireguard peers introduced several challenges since several organizations require the manual public exposure of their Wireguard peer. By 2022 Q3, it is expected to have the zero-touch deployment solution implemented, by following the milestones presented in Section 3.2.4.

When these milestones are accomplished, the next step is to deploy a mesh network between all testbeds, which will then be evaluated according to:

- The bandwidth provided by the tunnels
- The percentage of lost packets
- The communication latency
- The jitter

6.1.3 Implementation to enable testing

The testing framework is described in the document D5.3 [12]. The main components of the CI/CD Service are the CI/CD Manager, CI/CD Agents, Local Test Repositories, and the Test Results Visualization Dashboard (TRVD). CI/CD manager is the central component of our testing framework, being responsible for orchestrating the validation processes. The CI/CD agents and LTR are deployed at each test site, connected to the CI/CD manager, which reports the test results on to the TRVD. Table 6-2 summarises the implementation status of various components at each test site.

Test-site	CI/CD Agent Deployed ?	Has The Connectivity Between the Agent and the Manager Been Achieved?	LTR Deployed?
Averio	Yes	Yes	Yes
Bucharest	Yes	Yes	Yes
Bristol	Yes	Yes	Yes
Ljubljana	Yes	Yes	Yes
Murcia	Yes	Yes	Yes
Patras	Yes	Yes	Yes

Table 6-2: Test components deployment status

6.2 Implementation to support hosting network slices

D3.1 [2] introduced 5GASP's onboarding and deployment methodologies, defining a unified onboarding model, through a NetApp bundle that comprises all the artifacts needed to onboard and deploy a NetApp in a testbed and validate it. This bundle includes a NEST, that shall result in the deployment of the Network Slice where the NetApp itself will reside. Although, until this date, the deployment of said Network Slice has not been achieved. Instead, we chose, as an initial approach, to start with the manual deployment of static eMBB Slices in all testbeds. Table 6-3 presents the deployment status of these Network Slices.

Testbed	Status
ITAv	Achieved in May 2022
OdinS	Achieved in May 2022
UNIVBRIS	Achieved in March 2022
ININ	Achieved in March 2022

UoP	Achieved in December 2021
ORO	Achieved in March 2022

Table 6-3: Status of the Static eMBB Network Slice Deployment

6.2.1 Patras (UoP) facility

The Patras facility eMBB slice delivery is based on Amarisoft [15] 5G Radios and Open5GS [16] core solution. The latter was selected as it is open-source, service-based and can be deployed in a Kubernetes [17] cluster, as well as in VMs. Eventually, the approach to deploy the core in a Kubernetes cluster was elected as it follows the contemporary design methodology of cloud-native applications.

As Figure 6-1. illustrates, the core comprises Containerized Network Functions (CNFs) which point to the respective Helm [18] Charts to supply Open5GS as a Kubernetes service. Specifically, the CNFs are encapsulated in two network services: i) a network service that provides the overall 5G Core functionality, except User Plane Function (UPF), and ii) a network service in charge of deploying multiple discrete instances of UPF, based on specific placement preferences. The UPF instances self-register at the Session Management Function (SMF) of the deployed 5G Core, simultaneously exposing their data path, which can be leveraged by a NetApp to stream traffic. Both services accept instantiation parameters concerning specific deployment IP addresses.

Moreover, a discrete Radio Management Server is employed that offers a RESTful API to manage the underlying Radio Access nodes, e.g. registration to a 5G Core, start/stop/restart action, profiling, etc. At present, slice reconfiguration involves the registration of new gNodeBs to the deployed 5G Core. The management server is a static VM, while its transition in a containerized environment and eventually its packaging as an orchestrated NFV artefact is a work in progress.

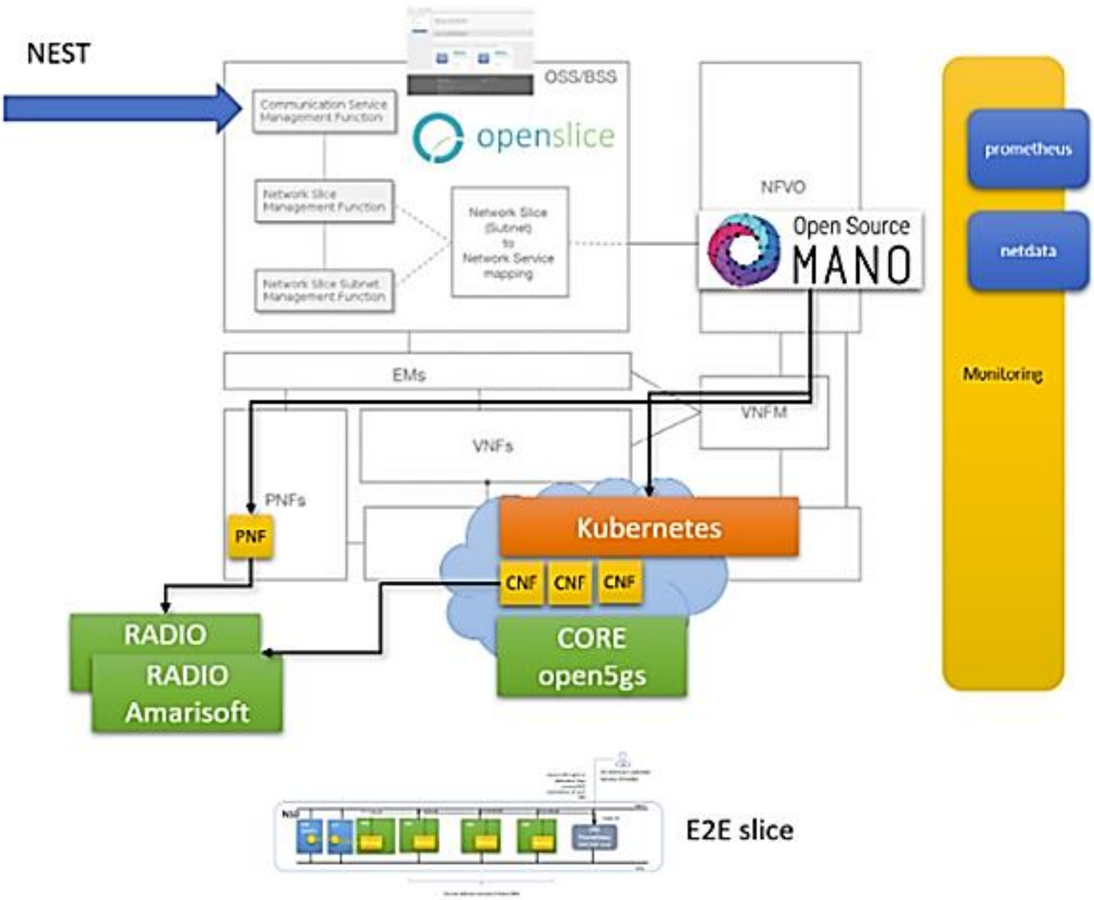


Figure 6-1: UoP Network Slice delivery (Kubernetes Cluster).

Following, Figure 6-2 and Figure 6-3 present the orchestrated deployment of the previously described eMBB Network Slice as captured through NODS and OSM's UI, respectively.

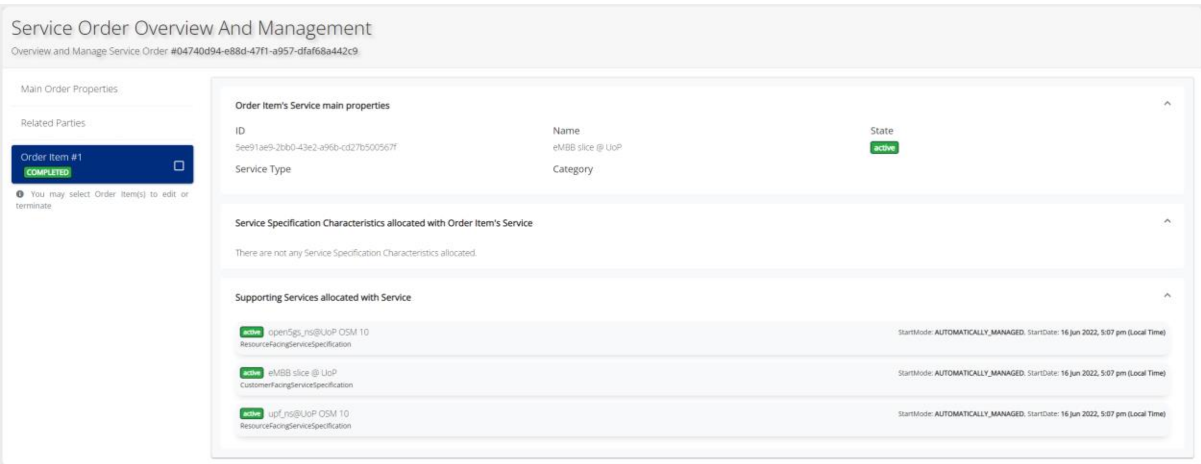


Figure 6-2: UoP Network Slice through NODS

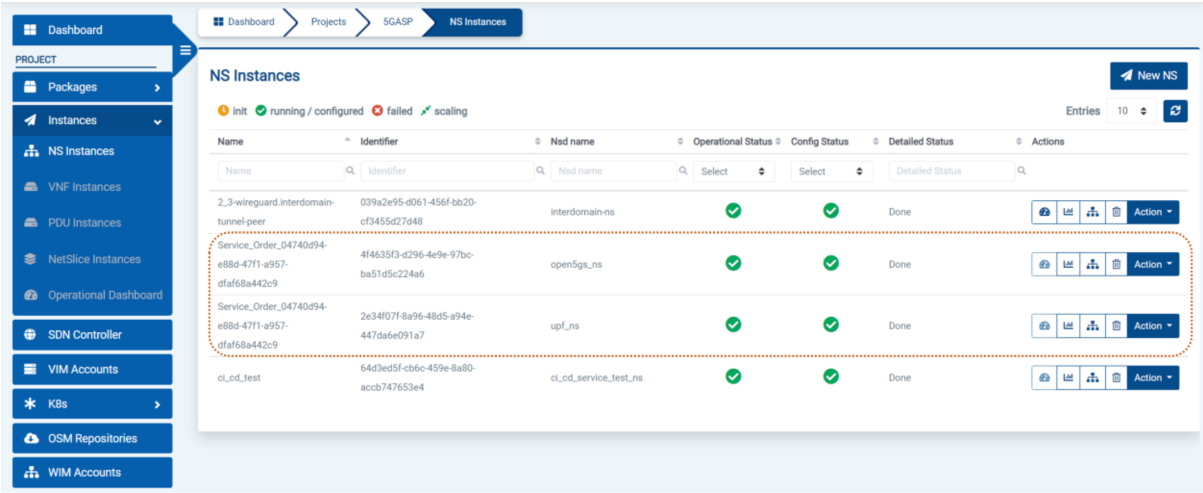


Figure 6-3: UoP Network Slice through OSM

6.2.2 ITAv's facility

ITAv's eMBB slice is delivered through a Huawei 5G Core and a Huawei 5G RAN. Since Huawei's 5G Core's management is done either via a Web GUI or SSH commands executed directly in each component of the Core, it was impossible to provide an out-of-the-box mechanism to enable the programmatically orchestration of eMBB slices. Thus, ITAv was forced to develop a RESTful API in to programmatically manage the 5G System.

This API abstracts the (I) execution of the SSH Commands directly in the components (using Paramiko to do so) and (ii) the configuration of the slices via Huawei's Mobile Broadband Automation Engine's (MAE) web interface (using Selenium to perform said configurations). The documentation of this API is presented in Figure 6-4.

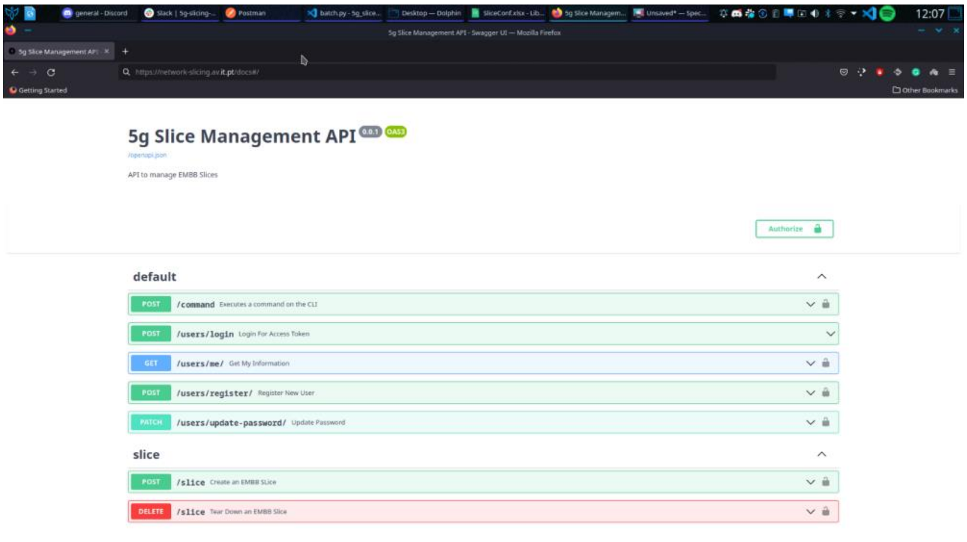


Figure 6-4: ITAv's 5G Slicing API Documentation

To orchestrate a new eMBB slice, the NODS interacts with this API, posting a payload similar to the one displayed on Figure 6-5.

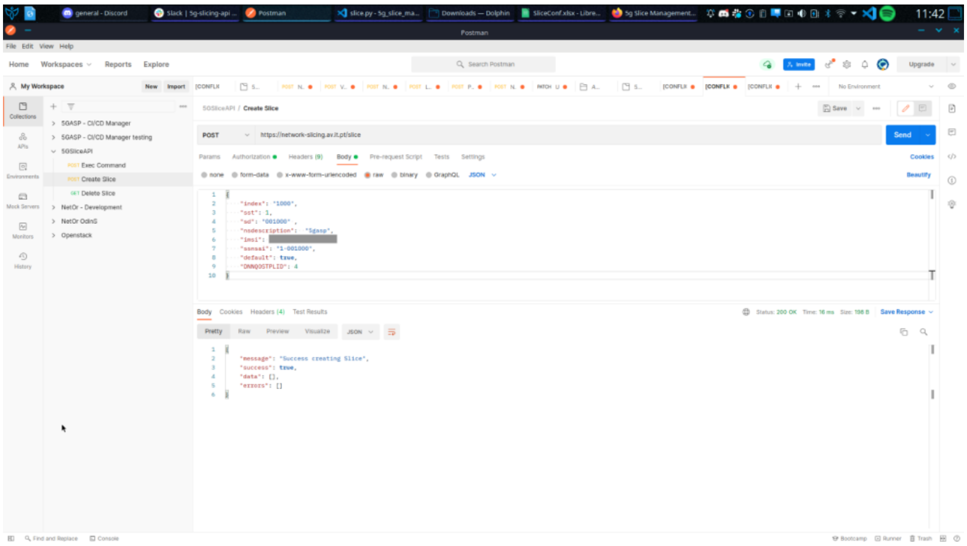


Figure 6-5: Network Slice Creation Payload

Following, Figure 6-6 and Figure 6-7 present the orchestrated deployment of the previously described eMBB Network Slice as captured through NODS and Huawei's MAE GUI, respectively.

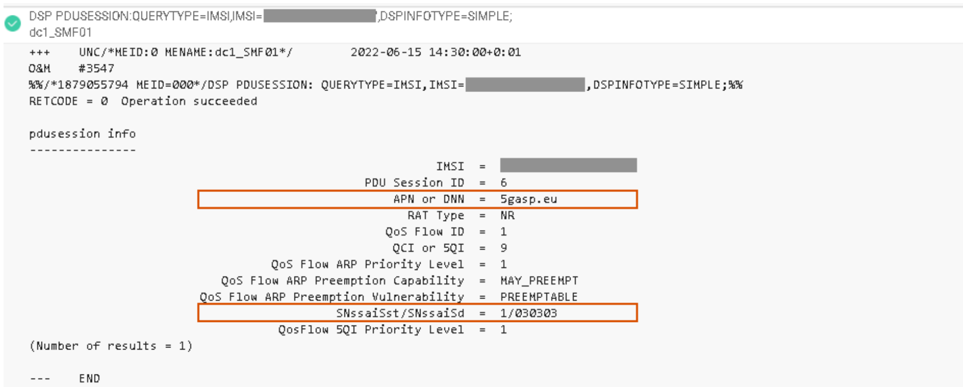


Figure 6-6: ITAv's Network Slice Information

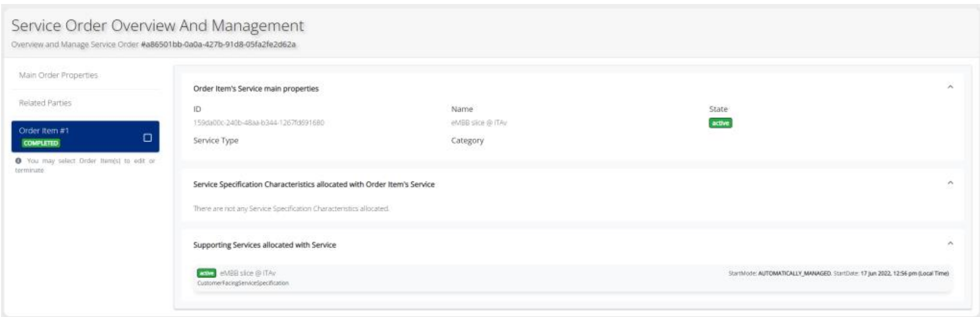


Figure 6-7: ITAv's Network Slice through NODS

Although this approach is not aligned with the ones of the other testbeds, ITAv is committed to upgrade their solution. Thus, it is expected that their API (i) offers a SOL0005 standardized API [9] and (ii) can be packaged into a VNF. It is expected that this is achieved between 2022 Q2 and Q3.

6.2.3 ININ's facility

The ININ/Ljubljana facility bases on Amarisoft [15] SDR-based 5G NR and (also Amarisoft) core solution. The solution enables flexibility for the core network component and BBU component to be deployed as containers (this is also one of the baselines for the ININ's NetApp). Figure 6-8 shows Amarisoft graphical user interface, i.e., logs for the gNB component.

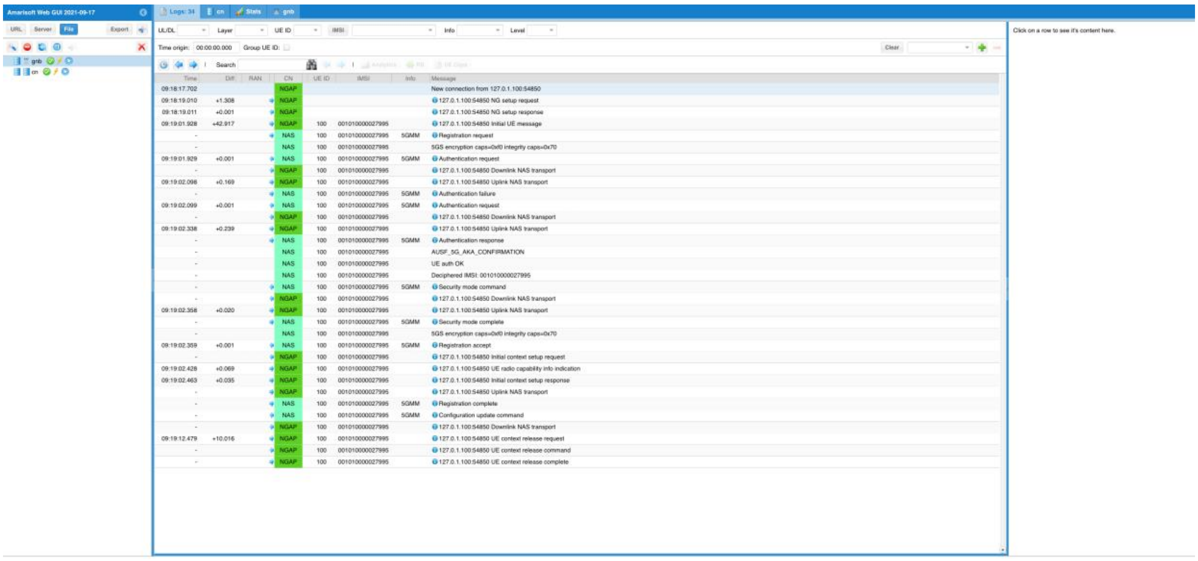


Figure 6-8: ININ

As other testbeds, ININ facility delivers eMBB slice. Following figures, i.e., Figure 6-9 and Figure 6-10, present the orchestrated deployment of the eMBB Network Slice as captured through NODS and OSM's UI, respectively.

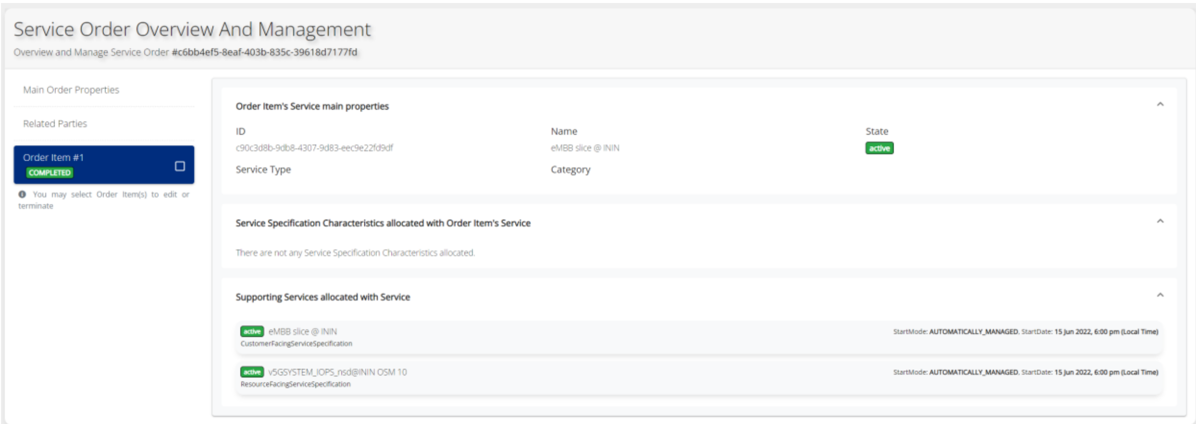


Figure 6-9: ININ's Network Slice through NODS UI

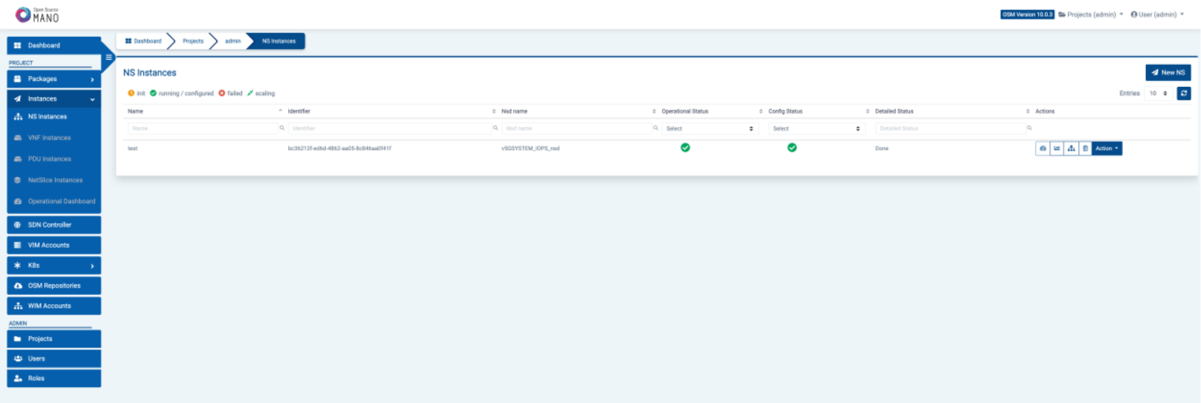


Figure 6-10: ININ's Network Slice through OSM UI

6.2.4 OdinS' facility

The Murcia facility relies on Free5GC [19] as 5G core and uses Amarisoft's [15] 5G gNB stack with SDR radios and outdoor RRH. Free5GC was selected as it offers flexibility in its configuration and deployment (as it's an open-source solution, and container-based, which allows freedom to deploy it either in a Kubernetes cluster or in a regular VM). The latter approach is used in Murcia, as the free5GC is deployed in OpenStack included in a regular VM-deployment. All the components comprising the 5G core are included on the deployed VM. Therefore, the Amarisoft RAN is configured to be connected to the deployed 5G-core. No further configuration between them is required for now in order to enable the connection between both entities, thus creating the whole functioning system.

Therefore, Figure 6-11and Figure 6-12 shows the request to deploy an eMMB network slice, from the NODS point of view and from the OSM located on Murcia Site, respectively.

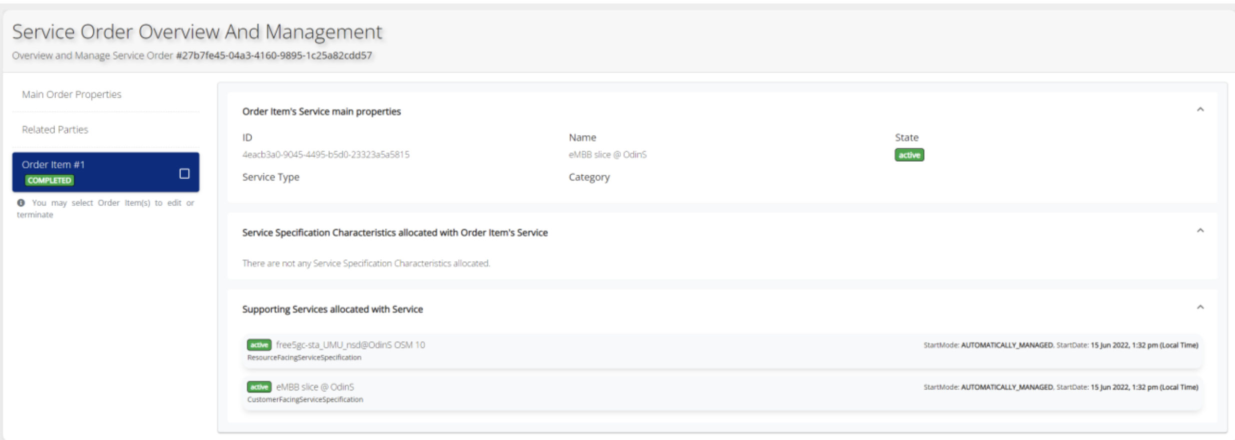


Figure 6-11: OdinS' Network Slice through NODS UI

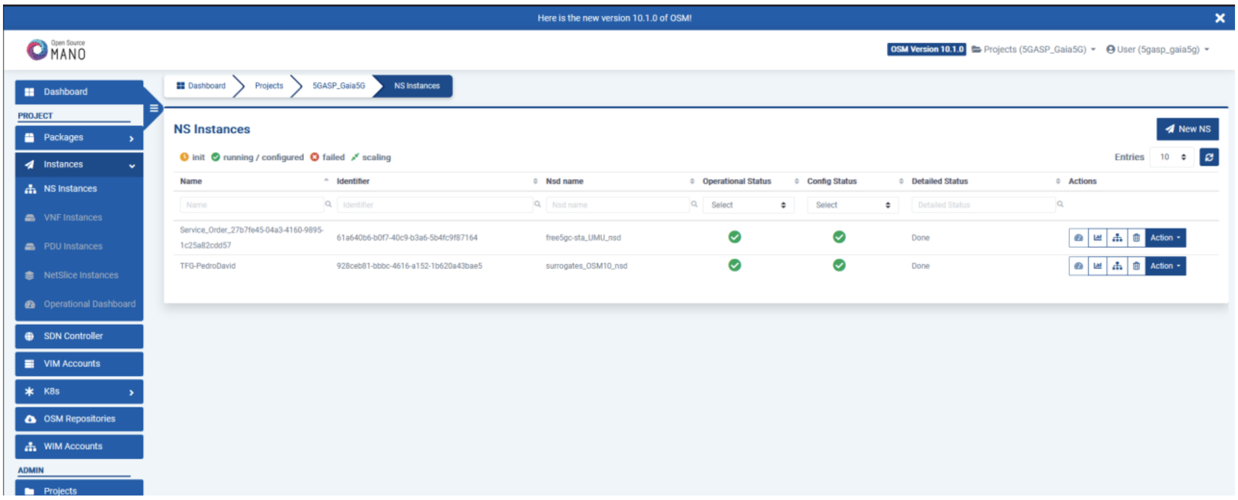


Figure 6-12: OdinS' Network Slice through OSM UI

6.2.5 Bristol's Facility

Bristol's eMBB slice is delivered through open5GS [16] core and Nokia RAN. The service is virtualized and deployed as VNFs on the OpenStack installation. The VNF is deployed the day-0 configurations using cloud-init to setup the access parameters for the deployment on the underlying OpenStack [20].

During the deployment of the eMBB slice, the service is created at the NODS and it communicates the request to the Bristol's OSM using the NODS SBI. Figure 6-13 shows the NODS request to deploy the Bristol's eMBB slice. The slice request and the deployment is shown in Figure 6-14 on Bristol's OSM.

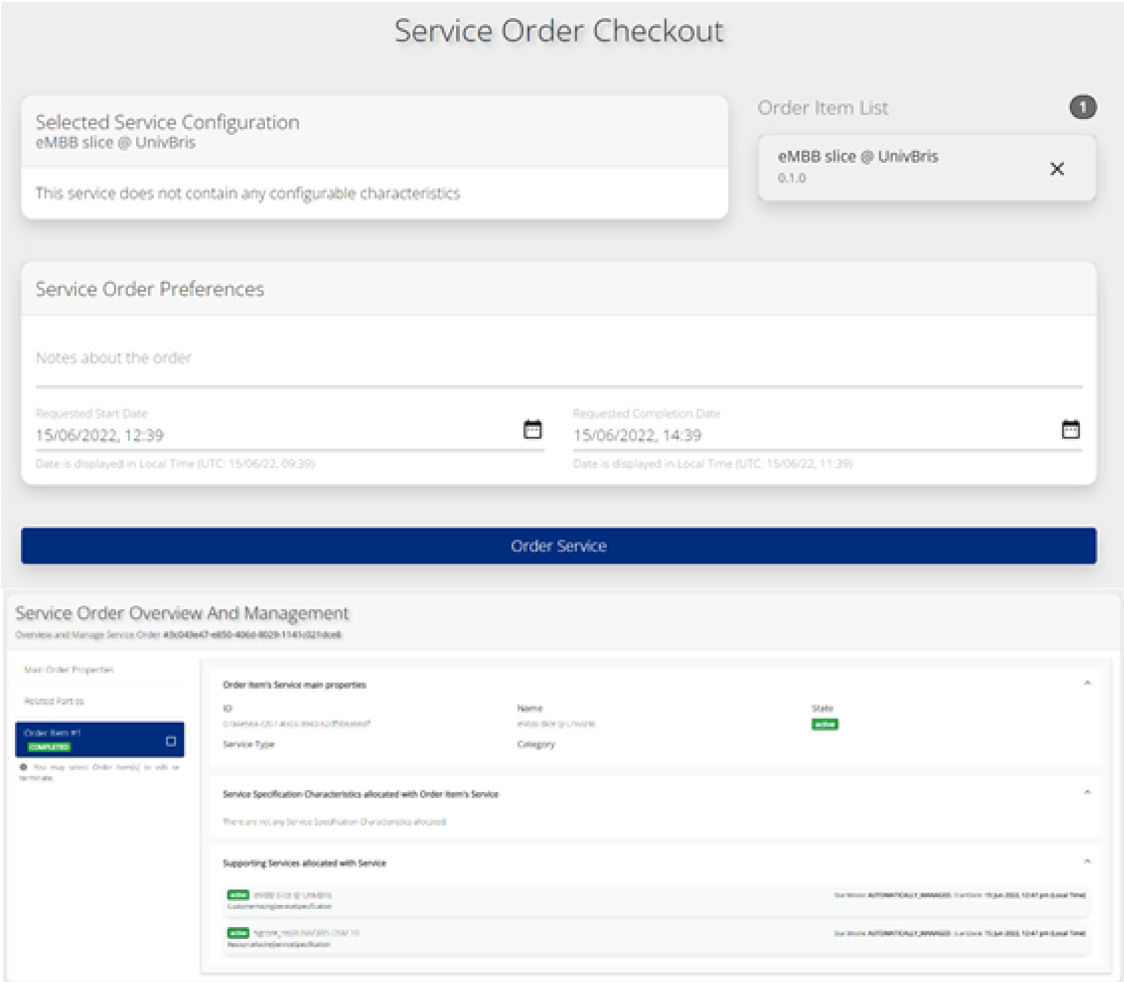


Figure 6-13: Service order to deploy Bristol's eMBB slice- NODS view

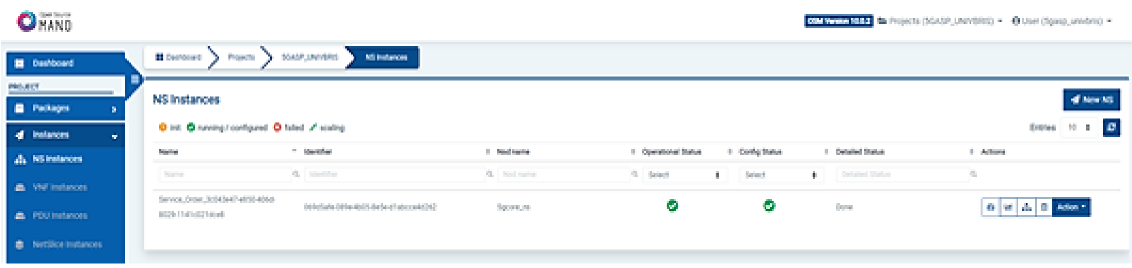


Figure 6-14: eMBB slice deployment at Bristol facility- OSM view

6.2.6 Bucharest Facility

The Bucharest facility eMBB slice delivery is based on a 5G OpenAirInterface solution built on 5G SA components, gNB RAN [21] and 5GCN Core [22]. The former are deployed in the ORO premises as OpenRAN and Open Core, running either in regular VMs (OpenStack) or Kubernetes environment, as all the core components are currently deployed as VMs, as seen in Figure 6-15.

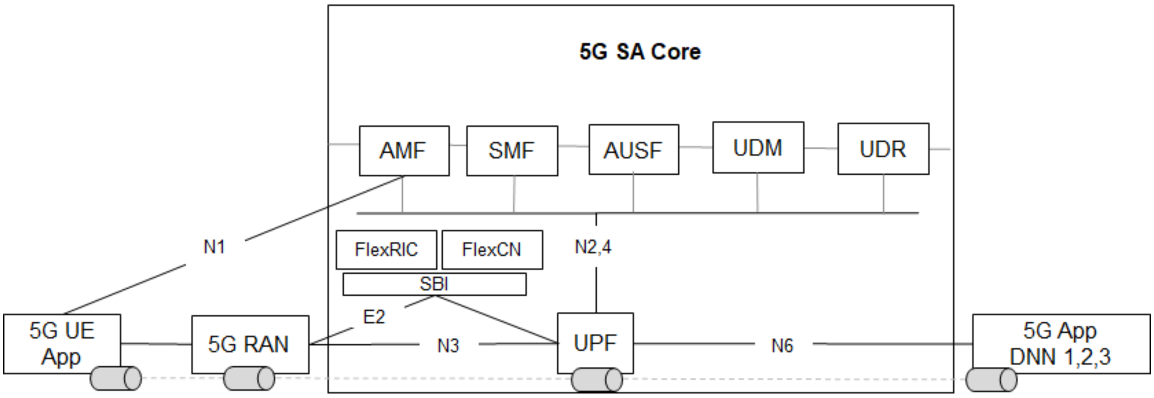


Figure 6-15 5G SA service network architecture Bucharest facility

Once deployed, the CN services are automatically instantiated with Jenkins (CICD) support and with proper provisioning of the services in the network slice(eMBB). **Figure 6-16** depicts the view of instantiated elements in the NODS system.

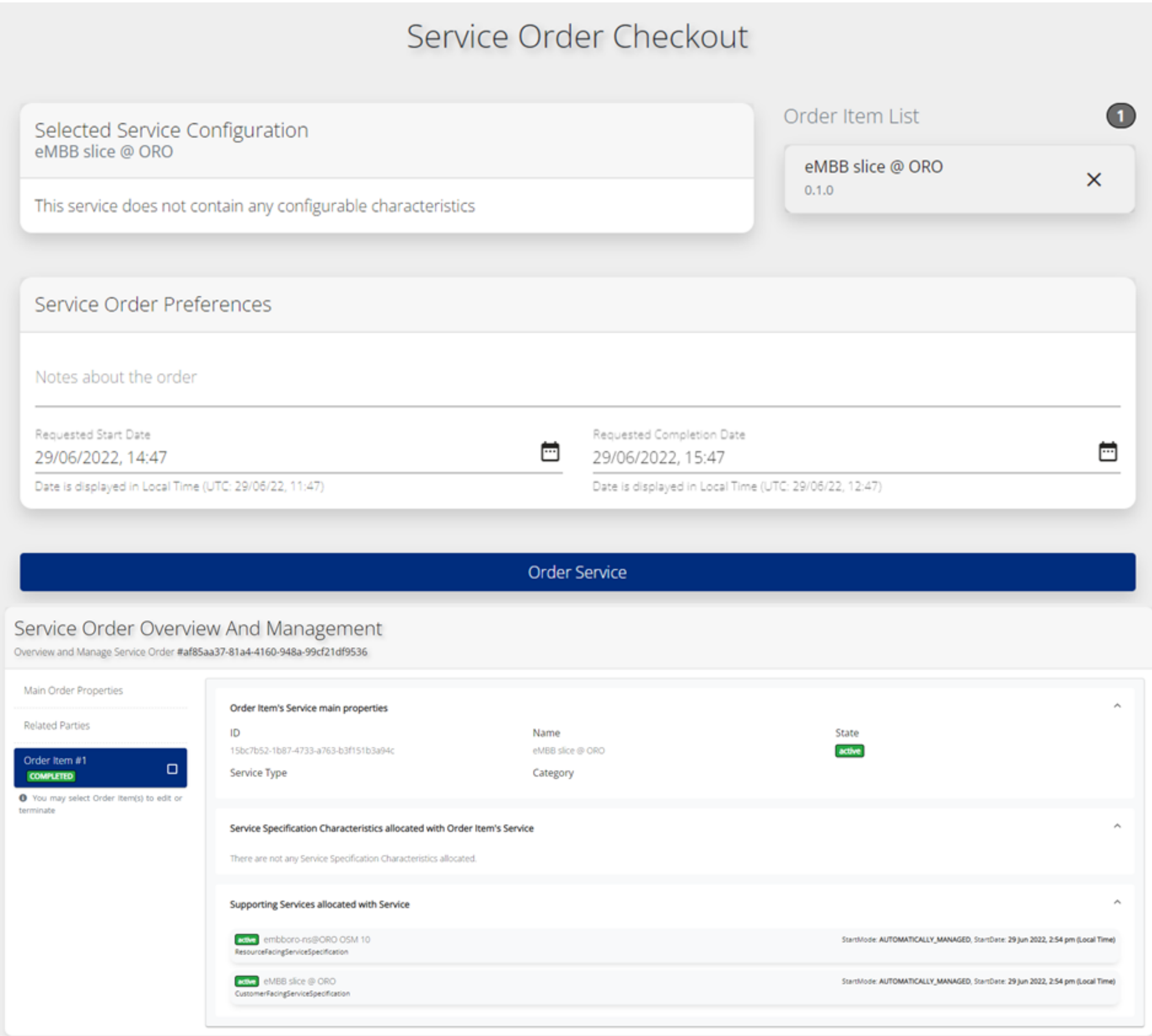


Figure 6-16 Service order to deploy ORO's eMBB slice - NODS view

The facility has OSM also integrated with the Bucharest virtualised infrastructure as shown in Figure 6-17 for NS and Figure 6-18 for VNF regarding the OSM eMBB deployment.

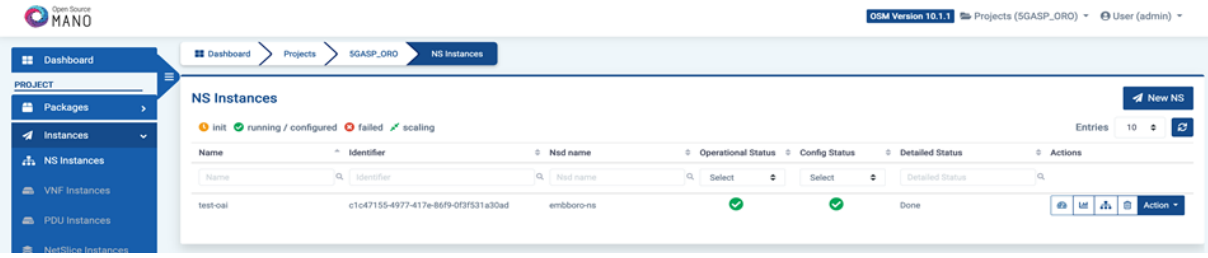


Figure 6-17 eMBB NS slice deployment at Bucharest facility

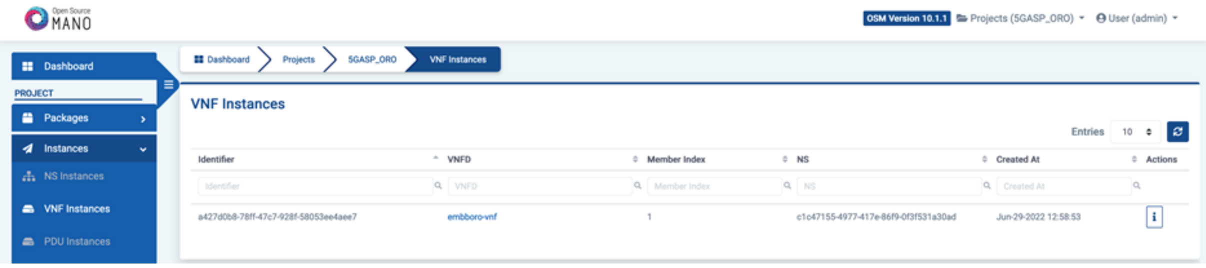


Figure 6-18 eMBB VNF slice deployment at Bucharest facility

As a pre-requisite for the service slice implementation, the OSMv10 is integrated in this scenario with the ORO/Bucharest VIM, as highlighted in Figure 6-19 for the OpenStack and **Figure 6-20** for VNF deployed in the Openstack infrastructure.

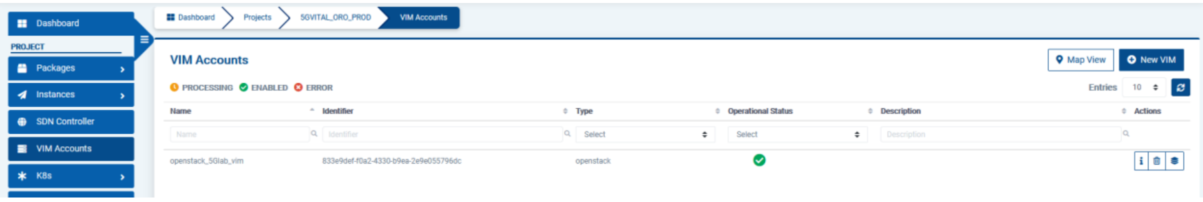


Figure 6-19 OSM integration with VIM in Bucharest facility

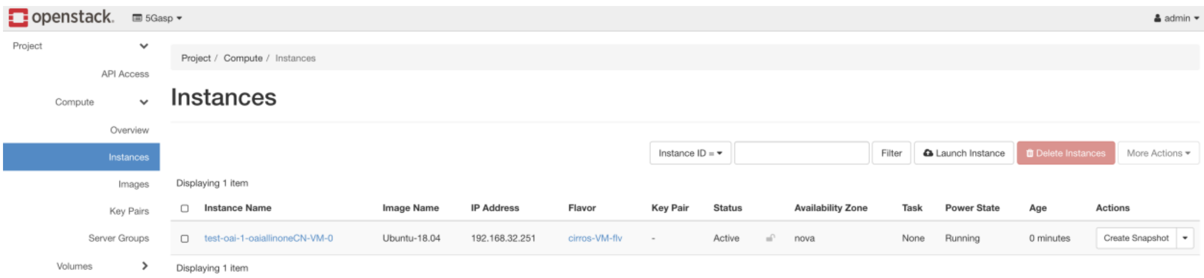


Figure 6-20 Openstack VMs

The eMBB slice deployed in this context is evaluated in terms of performance in line with the NetApps requirements and by using N78 5G spectrum.

6.3 Future implementation plans

6.3.1 Bristol facility

In document 3.1 [2] we have discussed the design of the testbed at Bristol's site. This section will provide an update to the implementation plans discussed in D3.1 [2] and the future deployment plans in context of the 5GASP. Following table Table 6-4 summarises the status of the infrastructure deployments discussed in D3.1 [2]:

Tasks	Status
Phase1: Infrastructure setup and orchestration integration	
Openstack deployment	Done
OSM and Openstack Integration	Done
Kubernetes deployment	Done
OSM and Kubernetes integration	Done
MEC nodes availability	Done
Network design and setup	Done
VPN service to access UNIVBRIS testbed	Done
Phase 2: Setting up 5G core VMs and network services with initial configuration	
Setting up platform and images for 5G SA core deployment	Done
Deployment of a simple eMBB slice	Done
Phase 3: VNFD and NSD definition for the 5G core and UPF VNFs with initial configuration	
Development of VNFDs and NSDs for eMBB slice deployment	Done
Phase 4: Configuring gNB and integration with 5G core	
2x gNBs configuration for 5GASP	2gNBs integrated, one with RU in band n77, the other with RU in band n78 but we don't have licence to operate in band n78 now
Phase 5: Testing and evaluating eMBB slice	
Deploy 5GSA core using OSM	Done
Verify gNB connection with the eMBB slice	Done
KPI measurements	In Progress

Table 6-4: UNIVBRIS-Infrastructure deployment status

Future enhancements:

For the next steps, the Bristol facility will be enhanced with additional 5G cells aiming at expanding the 5G SA coverage in band n77 within the city of Bristol. The network performance will improve by utilizing the increased spectrum availability in band n77 (100 MHz). Additional radio and core functionality and higher network performance and stability are expected after upgrading the software of the RAN. Specifically, the software upgrades should provide throughput and latency gains, additional 5G slicing functionality, as well as improved cell handovers. The planned updates is summarised in Table 6-5.

Upgrades	Quantity	Description

5G NR (n77)	2	Additional 5G NR to improve coverage and throughput performance, while creating more handover locations. Three n77 cells will be available by the end of the project.
Latest gNB Software	3	Software upgrade for three Bristol sites improving network stability, throughput, latency and the handover procedure.

Table 6-5: UNIVBRIS-Planned infrastructure updates

6.3.2 ITAV's facility

D3.1 [2] and D5.2 [23] defined various milestones regarding ITAv's testbed design and the services it shall provide. In this deliverable we examine the status of the previously defined milestones, and present future objectives we aim to achieve. Table 6-6 presents this information.

Task	Status
Infrastructure setup and orchestration integration	
OSM's REL11 deployment	Done
Openstack's resources upgrade, to support the deployment of the consortium's NetApps	Done
Deployment of a Kubernetes Cluster	In progress
Interconnection between OSM and the Kubernetes Cluster	In progress
Interconnection between Openstack and 5G Core's Data Network	In progress
5GASP's Validation Service	
Implementation of the TMF653 API Standard over the CI/CD Manager	Done
Integration of the CI/CD Manager and NODS, enabling the triggering of new validation processes, by the NODS	Done
Support for Developer-Defined Tests	Done
Infrastructure Monitoring	
Deployment of a Prometheus and Grafana stack to monitor VNF-related metrics	In progress
Deployment of a monitoring solution to collect and process 5G Core's metrics	In progress
eMBB Slice	
Deployment of a Static eMBB Slice	Done
Interconnectivity between the gNBs and the eMBB Slice	Done
KPI evaluation	Done
Development of an API to enable the programmatic deployment of new eMBB Slices	Base Implementation Achieved
Implementation of mechanisms to support runtime operations on the eMBB Slice	In progress

Table 6-6: ITAv's Milestones Status

6.3.3 ININ's facility

The following tables presents current status and milestones of ININ's facility as an evolution from the deliverable D3.1 [2]. The tasks are grouped based on the main areas of work, such as infrastructure and orchestration, 5GASP's CI/CD, monitoring and 5G slice. All mandatory tasks to support NetApp deployment have been realized, some more advanced mechanisms are currently work in progress.

Task	Status
Infrastructure and orchestration	
5GASP dedicated Open Source MANO (OSM) Release 10	Done
5GASP dedicated OpenStack tenant	Done
5GASP dedicated Kubernetes cluster	Done
OSM10 and OpenStack integration	Done
OSM10 and Kubernetes cluster integration	Done
Private docker image registry	Done
Network connectivity between 5G core and 5GASP tenant in OpenStack and Kubernetes	Done
VPN access for 5GASP NODS	Done
VPN access for 5GASP CI/CD service	Done
5G UE supporting containerized deployments	Done
CI/CD pipeline	
Deployment of 5GASP CI/CD agent	Done
Support for Developer-Defined Tests	In progress
Monitoring	
Deployment of a Prometheus and Grafana stack to monitor VNF-related metrics	In progress
qMON measurement tool for end-to-end network performance monitoring (5G RAN + CN) – backend deployment	Done
qMON measurement tool for end-to-end network performance monitoring (5G RAN + CN) – agent deployment	Done
Verify 5G UE SA connectivity on Industrial GW device (based on SierraWireless EM9191 5G modem)	Done
Verify 5G UE SA connectivity on 5G Android devices (OnePlus 8T, OnePlus 9T)	Done
Development of 5G UE API to allow automated end-to-end testing	In progress
5G slice	
5G mobile system components containerization	Done
5G mobile system VNF and NS	Done
eMbb slice deployment via OSM10	Done
Initial/on-demand radio and service KPI measurements	Done
Continuous radio and service KPI measurements	In progress
Implementation of mechanisms to support day-2 operations on a NS/VNF providing the eMBB Slice	In progress

Table 6-7: ININ's Milestones Status

Planned enhancements

Current planned enhancements are as follows:

- Upgrades to latest 5G software releases available and required extensions and adjustments on the VNF/NS side,
- Provide 5G UE with Kubernetes-cluster capable to integrate into 5GASP orchestration environment to allow deployment of NetApp components while also supporting complex end-to-end testing scenarios based on qMON monitoring tool running on the same UE.
- Kubernetes-based [17] deployment of 5G mobile system components.

6.3.4 OdinS facility

The following table presents the milestones achieved in OdinS testbed. The deployment of the dynamic eMMB is supported, as well as the automatic deployment of tests through the CI/CD and of NetApps through NODS. Further work includes the enhancement of infrastructure monitoring.

Task	Status
Infrastructure setup and orchestration integration	
OSM's REL10 deployment	Done
Openstack's resources upgrade, to support the deployment of the consortium's NetApps	Done
Interconnection between OSM and OpenStack tenant	Done
Interconnection between Openstack and 5G Core's Data Network	Done
5GASP's Validation Service	
Integration of the CI/CD Manager and NODS, enabling the triggering of new validation processes, by the NODS	Done
Support for Developer-Defined Tests	Done
Infrastructure Monitoring	
Deployment of a Prometheus and Grafana stack to monitor VNF-related metrics	In progress
Deployment of a monitoring solution to collect and process 5G Core's metrics	In progress
eMBB Slice	
Deployment of a Static eMBB Slice	Done
Interconnectivity between the gNBs and the eMBB Slice	Done
KPI evaluation	Done

Table 6-8: OdinS Milestones Status

6.3.5 Patras facility

The section provides an update to the milestones and implementation plans imposed in D3.1 [2] and D5.2 [23] concerning Patras facility. Below, the Table 6-9 represents the aggregated status of the aforementioned milestones, as well as on going implementation tasks.

Task	Status
Infrastructure setup and orchestration integration	
OSM REL10 deployment (5GASP dedicated tenant)	Done
Openstack deployment (5GASP dedicated tenant)	Done
Kubernetes deployment (5GASP dedicated tenant)	Done
Interconnection between OSM and Openstack/Kubernetes	Done
Interconnection between NODS and underlying facilities	Done
5GASP docker and helm registry setup	Done
Design and implement testing orchestration within NODS	Done
Integrate TMF653 API in NODS	Done
MEC nodes availability	In Progress
5GASP's Validation Service	
5GASP CI/CD Agent deployment	Done

Interconnection between NODS and CI/CD Manager	Done
Infrastructure Monitoring	
Deployment of a Prometheus and Grafana stack to monitor VNF-related metrics	In progress
eMBB Slice	
Deployment of an orchestrated eMBB Slice	Done
District UPF deployment(s)	Done
Interconnectivity between the eMBB Slice and gNBs	Done
Runtime operations support	Done

Table 6-9: Patras Milestones Status

6.3.6 Bucharest facility

The following tables presents current status and milestones of Bucharest facility as an evolution from the deliverable D3.1 [2]. The table is grouped based on the main areas of activities, as virtualised infrastructure, orchestration, CI/CD, monitoring and 5G network slices. In relation with the preliminary network slice, the mandatory activities have been accomplish, as more effort will be oriented to the advanced 5G features, currently work on progress tasks, described in Table 6-10.

Task	Status
Infrastructure and orchestration	
5GASP dedicated Open Source MANO (OSM) Release 10	Done
5GASP dedicated OpenStack tenant	Done
5GASP dedicated Kubernetes cluster	Done
OSM10 and OpenStack integration	Done
OSM10 and Kubernetes cluster integration	Done
VPN access to the testbed	Done
Network connectivity between 5G core and 5GASP tenant in OpenStack and Kubernetes	Done
VPN access for 5GASP CI/CD service	Done
5G UE working with the 5G network	Done
CI/CD pipeline	
Deployment of 5GASP CI/CD agent	Done
Monitoring	
Deployment of a Prometheus and Grafana stack to monitor VNF-related metrics	Done
iperf3 measurement tool for end-to-end network performance monitoring	Done
5G UE measurements and connectivity metrics	Done
5G slice	
5G Core network elements(RAN/Core); N78	Done
eMbb slice deployment	Done
Initial/on-demand radio and service KPI measurements	In progress
5G RAN and service KPI measurements	In progress

Table 6-10 Bucharest Milestones status

Further enhancements:

For the next steps, Bucharest facility will implement enhancements in the 5G area, by upgrading the 5G RAN network to cope up to 50MHz bandwidth in N78, including additional RAN and Core stability.

6.4 Support for vertical's needs

The following Table 6-11 maps the technical requirements of the NetApps and the ability of the testbeds to support the NetApp(s) deployment:

Id	Requirement	NetApps	ITAv (PT)	UoP (GR)	ORO (RO)	OdinS (SP)	ININ (SL)	UNIVBRIS(GB)
1	UE with specs similar to Raspberry Pi4 - Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz - 1GB to 8GB LPDDR4-3200 SDRAM - 8GB storage - 5G Connectivity	All	Yes	No	No	Yes	No	No
1.a	Vehicular vertical specific - 802.11p/ITS-G5 or C-V2X connectivity - Geolocalization - C-ITS software stack	1,2,3,4	No	No	No	Yes	No	No
1.b	PPDR vertical specific - K8S installation - Camera module	11	In progress	Yes	No	Yes	Yes	Partial
1.c	PPDR vertical specific - GPU module	5,6	No	No	No	No	No	Yes
1.d	Android smartphone with 5G connectivity	7,10	Yes	Yes	Yes	Yes	Yes	Yes

2	USB3 5G SA modems that can be connected to an UE if no built-in connectivity	All	Yes, 1. But more can be acquired	No	When available	Yes	No	Yes. In-house built 5G CPE devices on Linux. Support for 5G SA, WiFi AP and 2xGbE.
3	802.11p/ITS-G5 modems that can be connected to an UE if no built-in connectivity	1,2,3,4	No	No	Yes	Yes	No	No
4	C-V2X modems that can be connected to an UE if no built-in connectivity	1,2,3,4	No	No	No	No	No	No
5	Vehicle available (usage slots scheduled with testbed owners) to deploy vehicular NetApps under mobility conditions. It would host the physical OBU.	1,2,3,4	If needed	No	N/A	Yes	No	Shared with other groups. Possibly depending on car state and availability.
6	eNB with 4G/3G data support	2		Yes, 4G	Yes, 4G	Yes, 4G	Yes, 4G	Yes
7	UEs with 4G/3G data support	2	Yes	Yes	Yes	Yes, 4G	Yes	Yes
8	High number of instantiated containers	3	Under study	Yes	Yes	Under study	Under study	Yes
9	Interaction with the SDN controller of the domains to be able to react to the domain changes. Although this is the initial approach that was considered when developing the initial PoC service, other options can be explored to do this, such as a dynamic allocation of a network slice. This is a WIP by OdinS,	4	Under study	No	Yes, limited to SDN transport network	Under study	No	Under study. Could use the Zeetta Automate APIs for vlan/service creation

	further details will be given.							
10	MEC/Cloud with GPU resources available for containers or VNFs	5,6	Just RTX GPUs, but Jetsons may be acquired, if needed	No	Yes (4 x nVIDIA Tesla T4)	Yes (waiting material arrival)	No	Yes
11	Radio monitoring data from the UE or the RAN available	7	No	Yes (data from RAN)	Yes	Under study	Yes	Not readily available. Currently under development.
12	Radio access points with edge compute nodes 1:1	7	Under study	Under study	No	Under study	Yes	Yes
13	Modified RO's OSM10 to allow selecting flavors with extra specs	9	No, but we can try to implement it	No	Yes	No	Yes	Yes
14	Openstack flavor with extra specs (PCI PASSTHROUGH:ALIAS, HW:CPU_SOCKETS, HW:CPU_CORES, HW:CPU_THREADS)	9	No, but we can try to implement it	No	Yes	No	Yes	Yes
15	Openstack host with SDR card exposed to OSM	9	No, but we can try to achieve this	No	N/A	No	Yes	No
16	Wide area of coverage for vehicular scenario testing, not just indoor or near the gNB	10	Yes	No	Yes, NSA	No	No	More radios are required

Table 6-11: NetApp requirements mapping to the testbeds

7 Conclusions

This deliverable is the second release of the status update on the actual implementation and the readiness status of the 5GASP testbed, considering the earlier specified technical goals and capabilities presented in D3.1 to define the 5GASP project's experimental services, interfaces, and continuous integration. The reported status updates are aligned with the progress of the design and implementation details of the NetApp Onboarding and Deployment Services (NODS) portal and the facility's support to host the NetApps. Alongside the implementation details status of various components to create the multi-domain SDN/NFV fabric, this document also reports the changes in the implementation plan and enhancements in the 5GASP infrastructure compared with the initial status reported in the deliverable D3.1.

The core progress updates are on the interdomain connectivity between the 5GASP test sites and the interaction with the CI/CD toolchain in regards to the interaction with the other facilities, testing framework, and the 5GASP NetApp store, as well as updates regarding the secure deployment of Network Services across the testbeds.

In terms of design, development, testing, and integration, the status of the ten identified and developed services for the 5GASP platform is completed. As for the interface development status, E1/E2/E4/E7 are all reported completed, whereas E3 and E5 are still in the test and design phases, respectively. E6 is briefly mentioned and is more suitable to be covered in WP5 and will be reported in detail in WP5 deliverables. Moreover, all testbeds reported the completion of manual deployment of static eMBB Slices in all testbeds.

As a cross task between WP-4 and WP-3, this deliverable presented the 1st version of a common sheet to conclude and map the technical requirements of the project NetApps and the ability of the testbeds to support the NetApp(s) deployment .

For future work, NetApps and testbed owners will gather additional requirements and annotations from a user's perspective to further enhance the implementation and deployment details of the 5GASP experimentation infrastructure for NetApps. This deliverable already provided details on the further implementation and enhancements where an updated version (D3.3), which shall evaluate and report the final status of this document based on the learned lessons from the 5GASP framework users, should be expected by the end of the project (M36).

8 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] 5GASP, “D3.1,” 2021. [Online]. Available: https://www.5gasp.eu/assets/documents/deliverables/D3.1%20Experimentation%20Services,%20Middleware%20and%20Multi-Domain%20Facilities%20Continuous%20Integration.pdf. [Accessed June 2022].

[3] OpenSlice, [Online]. Available: http://openslice.io. [Accessed June 2022].

[4] TMForum, “TMF633 - Service Catalog Management API REST Specification”.

[5] 5GASP, “D5.1,” 2021. [Online]. Available: https://www.5gasp.eu/assets/documents/deliverables/D5.1%20Initial%20Report%20on%20Test%20Plan%20Creation%20and%20Testing%20Methodologies.pdf. [Accessed June 2022].

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

[7] 3GPP, "TR 28.801; Telecommunication management; Study on management and orchestration of network slicing for next generation network (Release 15)," 2018.

[8] Google, “Blockly,” [Online]. Available: https://developers.google.com/blockly. [Accessed June 2022].

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

[10] ETSI, “Open Source MANO (OSM),” [Online]. Available: https://osm.etsi.org/. [Accessed June 2022].

[11] wireguard, “wireguard,” [Online]. Available: https://www.wireguard.com/. [Accessed june 2022].

[12] 5GASP, “D5.3 - Development of testing tools and framework for the automotive and PPDR verticals,” 2022.

[13] 5GASP, “D6.1 Initial report on building NetApps Community & defining a certification process,” 2022. [Online]. Available: https://www.5gasp.eu/assets/documents/deliverables/D6.1%20Initial%20report%20on%20building%20NetApps.pdf. [Accessed June 2022].

[14] T. Forum, “TMF620 - Product Catalog Management API REST Specification”.

[15] Amarisoft. [Online]. Available: https://www.amarisoft.com/. [Accessed June 2022].

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

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

[18] Helm. [Online]. Available: https://helm.sh/. [Accessed June 2022].

[19] free5GC. [Online]. Available: https://www.free5gc.org/. [Accessed June 2022].

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

[21] O. A. Interface, “OAI,” [Online]. Available: https://openairinterface.org/oai-5g-ran-project/. [Accessed June 2022].

[22] O. A. Interface, “OAI 5GC,” [Online]. Available: https://openairinterface.org/oai-5g-core-network-project/. [Accessed June 2022].

[23] 5GASP, "D5.2 - Integration guide and API reference manual," 2021. [Online]. Available: https://www.5gasp.eu/assets/documents/deliverables/D5.2%20-%20Integration%20guide%20and%20API%20reference%20manual.pdf. [Accessed June 2022].