H2020 5GASP Project

Grant No. 101016448

D3.3 5GASP experimentation services, middleware and multi-domain facilities continuous integration, final release

Abstract

This document provides the final status of 5GASP experimental services, interfaces, facilities support to host the Network Applications and the implementation details to create the multi-domain SDN/NFV fabric. This document is the last update to the 5GASP infrastructure and services implementation plan described in the deliverable D3.1 and D3.2. Alongside the status of various components of the 5GASP ecosystem, it also covers the changes to the proposed implementation plan and last planned enhancements in the 5GASP infrastructure.

Document properties

Document number	D3.3
Document title	D3.3 5GASP experimentation services, middleware and multi-domain facilities continue integration, final release
Document responsible	Andrei Radulescu
Document editor	Andrei Radulescu
Editorial team	Neobility
Target dissemination level	PU
Status of the document	Final version
Version	1.0

Document history

Revision	Date	Issued by	Description
Initial	15.01.2024	Kostis Trantzas (UoP)	Initial draft
V0.1	29.02.2024	Ioan Constantin (ORO)	Updated sections 7.3.6 and 7.4.6
V0.2	29.02.2024	Dirk Hetzer (EANTC)	Updated part 7.2
V0.3	01.03.2024	Miguel Ponce de Leon (VMware)	Updated Sections 3.1 and 4.4
V.0.4	26.03.2024	Andrei Radulescu (NEO)	Version ready for review
V0.5	27.03.2024	Yevgeniya Sulema (BLB)	Reviewed version
V1.0	28.03.2024	Andrei Radulescu (NEO)	Final 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	Sections 2.1.1, 2.1.2, 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 3.1, 3.2, 3.3, 4.2.1, 4.3, 4.4, 5, 7.1.1, 7.2.2, 7.2.3, 7.2.4, 7.3.1, 7.4, 7.4.1
ITAv	Rafael Direito, Diogo Gomes	Sections 3.2.9, 4.1, 4.2.2, 4.2.3, 6, 7.1.2, 7.2.1, 7.3.2, 7.4.2, Annex B
VMware	Miguel Ponce de Leon	Sections 3.1, 4.4, 7.4, 7.5
OdinS	Ana Hermosilla, Jorge Gallego-Madrid	Sections 2.1.3, 7.3.3, 7.4.3, 7.5.1
EANTC	Dirk Hetzer, Tareq Hellibia	Section 7.2
UoB	Chris Nicolaescu	Sections 4.4, 7.3.4, 7.4.4
ININ	Luka Korsic, Rudolf Susnik	Sections 7.3.5, 7.4.5, 7.5.2
ORO	Ioan Constantin	Sections 7.3.6, 7.4.6
NEO	Andrei Radulescu	Introductions, Conclusion, Editing
BLB	Yevgeniya Sulema	Review

Contents

List of Figures

List of Tables

List of Acronyms

Acronyms	Full Form
5GASP	5G Application & Services experimentation and certification Platform
3GPP	The 3rd Generation Partnership Project
API	Application Programming Interface
BPMN	Business Process Model and Notation
C-ITS	Cooperative Intelligent Transport Systems
CFSS	Customer Facing Service Specification
CI/CD	Continuous Integration/ Continuous Development
CIMs	Cloud Infrastructure Managers
CNF	Containerized Network Functions
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CSMF	Communications Service Management Function
E2E	End to End
ETSI	European Telecommunications Standards Institute
eMBB	enhanced Mobile BroadBand
gNB	gNodeB
GSMA	Global System for Mobile Communications Association
GST	General Slice Template
GTP	GPRS Tunnelling Protocol
GUI	Graphical User Interface
IKEv2	Internet Key Exchange version 2
IP	Internet Protocol
IPR	Intellectual Property Rights
IPSec	Internet Protocol Security
KPI	Key Performance Indicator
LCM	Lifecycle management
NaaS	Network-as-a-Service
MAC	Media Access Control

MANO	Management and Orchestration
NAS	Non Access Stratum
NAT	Network Address Translation
NBI	North Bound Interface
NEF	Network Exposure Function
NEST	Network Slice Template
NetOr	Network Orchestrator
NFV	Network Function Virtualisation
NFVO	Network Function Virtualisation Orchestrator
NODS	Network Application Onboarding and Deployment Services
NS	Network Service
NSD	Network Service Descriptor
NSI	Network Slice Instance
NSMF	Network Slice Management Function
OBU	On-Board Unit
OSM	Open Source MANO
OSS	Operations Support Systems
P2P	Peer to Peer
PCIe	Peripheral Component Interconnect Express
PPDR	Public Protection and Disaster Relief
qMON	Quality Monitoring System
RAM	Random Access Memory
RAN	Radio Access Network
RESTful	Representational State Transfer
RFSS	Resource Facing Service Specifications
RU	Radio Unit
SD	Slice Differentiator
SDN	Software-Defined Networking
SDO	Standard Development Organization
SMF	Session Management Function
SSL	Secure Sockets Layer
SST	Slice/Service Type
SUPI	Subscription Permanent Identifier
TC	Traffic Control
TMF	TeleManagement Forum
TRVD	Test Results Visualization Dashboard
UE	User Equipment
UI	User Interface
UPF	User Plane Function
URL	Uniform Resource Locator
USB	Universal Serial Bus
VIM	Virtualized Infrastructure Manager
VLAN	Virtual LAN
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor

VPN	Virtual Private Network
WM	Workload Management
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

1.1 Objectives of this document

The main objective of this document is to detail the tasks and advancements made in the 5GASP project's final phase, encapsulating tasks T3.1, T3.2, T3.3, and T3.4. Each task addresses a critical component of the 5GASP architecture, from the implementation of the DevOps experimentation services and the adaptation of 5GASP facilities to open interfaces, to the creation of a multi-domain NFV fabric and the tailoring of experimentation facilities to vertical Network Applications requirements.

This document is intended to cover the details of the implementation of the 5G Application & Services experimentation and certification Platform (5GASP) reference architecture explained in Work Package (WP) 2 and documented in D2.3 [1]. This document is not only an extension to the first and second versions described in D3.1 [2] and D3.2 [3], but a complete description that covers all the core building blocks of the 5GASP architecture.

Based on the technical goals presented in D3.2, this document covers the final requirements, architecture, design and implementation details of the Network Application Onboarding and Deployment Services (NODS) portal.

The reader can get the details on the interaction with the CI/CD toolchain, along with the interaction between facilities like NFVO, testing framework along with the 5GASP Network Application Store.

Finally, the document covers the implemented capabilities the 5GASP facilities are being able to support.

1.2 Approach and methodology

There have been a number of implementation decisions made in WP3 that have been influenced by the ICT-41 commonalities studies undertaken via WP2. For context, D2.3 Section 2.5 identified a number of commonalities across ICT-41 testbed architectures, specifically in the projects VITAL-5G, 5G-IANA and 5G-EPICENTRE.

From VITAL-5G, the “Network Applications Catalogue”, “Experiment Catalogue” and “Network Application Validator” from the VITAL-5G Catalogue were all identified and reviewed as common elements relating directly to the 5GASP Network Applications Onboarding and Deployment Services (NODS). In particular the “Network Application Validator” of VITAL-5G has in part influenced *the implementations to enable testing of 5GASP and those updates are described in Section 7.2 of this deliverable*.

There was also one component, the “Network Application License Storage & Manager” (VITAL-5G) that was found to be a possible interesting addition to the 5GASP architecture. This component in VITAL-5G provides a storage for the licenses associated to the Network Application and assists with the on-boarding of license information to the VITAL-5G platform.

From the 5G-IANA NOD's layer, the "NetworkApp Toolkit" and "Application Orchestrator" provide functionalities relating to modelling/design and provisioning/orchestration phases of the Vertical Service lifecycle and some revision to the 5GASP architecture has been considered.

The "Network Slice Template Management" component of the 5G-IANA SMRO layer was also identified as a common piece of functionality and formed a basis for an upgrade to the OpenSlice functionality of 5GASP.

Another area of commonality identified was the "Monitoring Platform" (VITAL-5G), "Monitoring & Analytics" (5G-IANA) and "Federated Prometheus" platform (5GASP) of all three projects. Given that traceability and logging of each Network Application streams have to be captured by the execution environment, collated together with all other streams from the application, and routed to one or more final destinations for viewing and long-term archival and analysis the 5GASP further examined this a report upon the additions in Section 7.4.

In regard to the 5G-EPICENTRE architecture there is significant overlap between the functionalities of the 5G-EPICENTRE "Front-end layer" and 5GASP "NODS Web UI", and some implementation changes to 5GASP are noted in Section 7.2 of this deliverable.

The architectural component of most common ground between 5G-EPICENTRE and 5GASP was the "Network Service Repository" from the 5G-EPICENTRE "Back-End layer" which relates directly to the functionality of the 5GASP NODS Service Orchestrator. Specifically 5G-EPICENTRE utilizes a JFrog Artifactory [4] as the private Helm [29] repository to store chart packages, whereas covering this same functionality in 5GASP is a Harbor [5] based private Helm repository.

Both implementations provide methods for uploading, retrieving, updating, and deleting Helm chart packages and so no new Repository API endpoints were identified for exposure.

1.3 Document structure

In Chapter 2 the final requirements of NODS are being described, along with the identified NODS potential actors.

Chapter 3 details the final architecture and design of NODS, along with its internal services, 5GASP portals, 5GASP experimentation APIs and other constituent services. Also, the northbound standardized interfaces are being described.

The 5GASP High Level Architecture, comprised of interfaces E1/E2/E3/E4/E7 and interaction with 5GASP Harbor, is being detailed in Chapter 4.

5GASP NODS implementation is being depicted in Chapter 5 with final aspects based upon the requirements, architecture, and interaction with the overall 5GASP ecosystem introduced in previous sections.

Chapter 6 sets the design, underlying technologies, and security considerations of the multi-domain NFV fabric.

The implemented capabilities from the 5GASP facilities are listed in Chapter 7, for each component and for each testbed.

Section 7.5 of Chapter 7 lists the specific enhancements testbeds developed in order to support vertical needs (Automotive and PPDR).

Finally, the document has two annexes. Annex A lists every revision from D3.1 and D3.2 per subsection, while Annex B documents the security compliance of the 5GASP publicly available platforms.

2 5GASP Network Application Onboarding and Deployment Services (NODS) final requirements

2.1 Supported Onboarding and deployment model

The 5GASP project aims i) to support the Network Application developer to onboard its application in an effortless and transparent way and ii) to provide an abstraction solution so that onboarding, activation, testing and certification of a Network Application can be properly performed on any NFV/3GPP compliant 5G System, besides the 5GASP facilities. Therefore, the need for a unified onboarding and deployment model emerged to cater to these needs.
Since the early steps of the project, a unified standards-based model was introduced that forms a "triplet" of entities combined into the respective deployment order for fulfilment, as depicted in Figure 1. The triplet consists of the following entities:

- The Network Application NFV artefact, that realizes the Network Applications to be deployed within the corresponding 5G network slice;
- The Network Slice, that need to be activated/configured prior to the application's deployment by a target 5G facility to meet with the specific requirement of the application;
- The Test Suite, described in terms of a test descriptor model, that will be executed after the activation of the application.

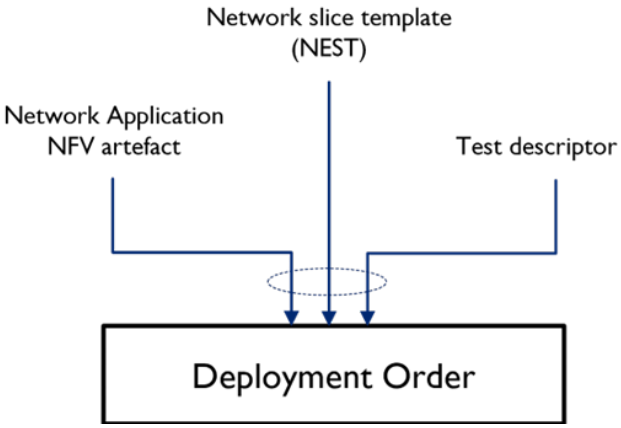

Figure 1: The 5GASP triplet

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. For this reason, a widely utilized industry standard was elected as the appropriate candidate, i.e. TM Forum. More details of the discrete entities and their final data models can be found in the following subsections, but someone may refer to D3.1 for an exhaustive description.
To end up, specifically concerning testing, a noteworthy mention is that the potential test execution against specified Test Suites, that are revised in D6.3 [6] (same publication date) may lead to the award of 5GASP certificate.

2.1.1 Network Application artefact

To support the TMF model adoption approach, onboarded Network Applications are referred as Resource Facing Service Specifications (RFSSs) expressing the resource aspects of the applications along with their respective requirements. The referred Network Applications are packaged as Virtual Network Function Descriptors (VNFDs) and Network Services Descriptors (NSDs), depending on the well-established YANG model [7]. Although, there was an initial initiative to also support TOSCA models [8] through a structural transformation service, the continuously decreasing interested on their respective representative, i.e. ONAP [9], made the 5GASP consortium to steer their attention into optimizing the YANG model approach even further. To that extent, an enhancement of the NFV architecture towards "cloud-native" was successfully achieved through the support of Cloud-native Network Functions (CNFs), as illustrated in Figure 2 and comprehensively presented in D3.1. Furthermore, the notable incorporation of a project wide and controlled Helm Chart and image repository, i.e. 5GASP Harbor, is an attempt to natively cater for the needs of a cloud-native application developer without vacating the 5GASP ecosystem, as described later in this document Interaction with 5GASP Harborin Section 4.4.

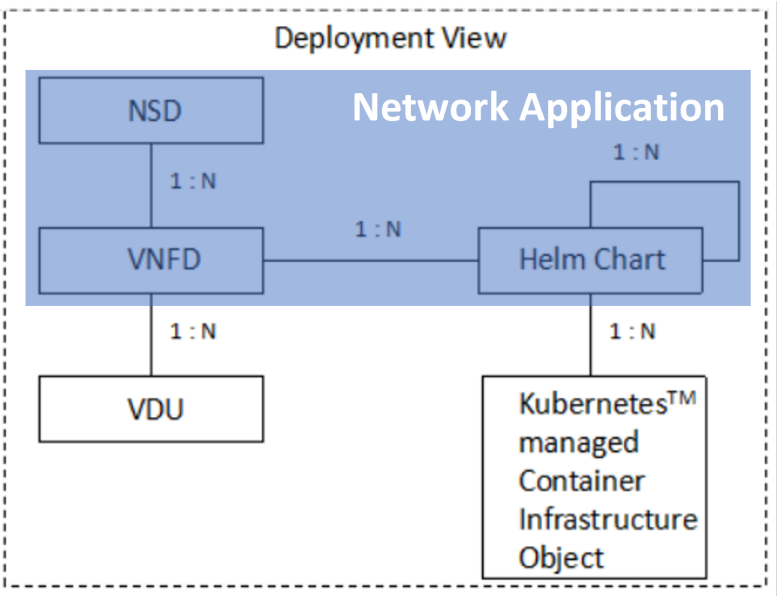

Figure 2: Network Application's deployment object mapping (edited from [26])

2.1.2 Network slice template

The following entity of the 5GASP triplet, i.e. the hosting network slice, is also defined and expressed in a standardized way. Each slice's network capabilities are aligned with the Global System for Mobile Communications Association (GSMA)'s Generic Slice Template (GST) [10] properties and each designated site provides information on the range of the support network capabilities in a value-populated subset of the aforementioned properties, i.e. Network Slice Templates (NESTs). The populated NESTs are expressed through the respective TMF data models, as described in D3.1, to equally redound towards the unified triplet model. These templates enable the Network Application developer to acquire a cognitive insight about the offered network capabilities and therefore select the appropriate accommodating network slice for its Network Application's needs. Eventually, due to the limited tangible networks slices offered by the testbeds, the 5GASP consortium decided to refrain from the idea of

automatically appointed network slices to match the Network Applications' requirement in favour of expanding the capabilities of the already offered network slices, as seen later in this document in Section 7.3.

2.1.3 Test descriptor

The last component of the 5GASP triplet is the test descriptor, which describes the test suite that will be executed to validate the Network Application and that, if successful, may get the application the 5GASP certification. The test descriptor is a YAML file that follows a custom structure developed to tailor the 5GASP framework requirements and ease its adoption by external developers and experimenters. The descriptor includes information and configuration parameters for each one of the tests that compose the test suite, and how the tests will be executed when the Network Application is already deployed. Two types of tests are considered: predefined tests, and developer-defined tests. The first ones are already available in the 5GASP platform and can be directly used by experimenters by referencing them with the correct parameters. They are mainly related to the network performance of the Network Application in the 5GASP testbeds. The later ones are defined and developed by the Network Application developers and experimenters. They are conceived to be used by them to evaluate application-specific functionality and to validate its correct operation in the 5G network.

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. The NODS requirements and its shaped architecture further in the document will stem from this input.

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: Anonymous user Use Cases.

Table 1: Anonymous user Use Cases

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. Network Applications, Network Slices, Test Descriptors
#03	Browsing the public ICT-41 repository	Browses the descriptions and potential deployment guidelines of available Network Applications across the ICT-41 onboarded projects
#04	Browsing the public catalogue of products	Browses available Product Offerings, which include the certified and/or undergoing certification Network Applications

#05	Signing in to NODS	Logs in to NODS through the corresponding form

2.2.2 5GASP Developer

The 5GASP Developer is in charge of developing vertical and cross-vertical Network Applications uploaded on 5GASP portal that, after testing and validation, are published on the 5GASP marketplace. Also, the 5GASP Developer has the role of designing and onboarding the respective NFV artefacts, i.e. VNFs/CNFs, comprising its Network Application. Throughout the project's durations, it was made apparent that the developer who designed the service aspects of the applications has usually developed or was involved in the related network functions, despite the slight differences of the technical areas. The supported use cases for the role are presented in Table 2.

Table 2: 5GASP Developer Use Cases

Use Case ID	Title	Description
#01	Account management	Browses and manages personal account information
#02	Network Application NFV packages development	Develop the Network Application's VNF/CNF, NSD artefacts and archives
#03	Network Applications NFV packages onboarding	Onboards VNF/CNF, NSD artefacts and archives
#04	Onboarded NFV packages management	Manages own onboarded VNF/CNF, NSD artefacts and archives
#05	Network Application management	Manages the constructed RFSSs of own Network Applications out of the onboarded NFV packages
#06	Service Test Specifications creation and test descriptors onboarding	Designs the Service Test Specifications defining the testing pipeline for a specific Network Application. These specifications encapsulate the respective test descriptors, i.e. YAML archives for the CI/CD services, that reference the desired tests for execution along with their parameters. Optionally provides/uploads own application-specific test archives, apart from the available predefined pool, to include in the testing pipeline.
#07	Browsing the NODS catalogue of available services	Available services refer to Network Application deployment, testing and optionally certification, e.g. Test a Network Application in a basic Enhanced Mobile Broadband (eMBB) slice
#08	Network Application deployment, testing and certification request to NODS	Deployment requests are captured through a Service Order which includes: • NEST selection, describing the hosting network slice of the Network Application – Area of Service definition • Network Application CFSS

		• Selection of a Service Test Specification that encapsulates a test pipeline, either with predefined or custom/uploaded tests • Optionally, a certification request in case the test pipeline meets the 5GASP certification-awarding requirements
#09	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
#10	Service Inventory view	A list of services linked to Service Orders are available, along with their run-time information
#11	Notifications about the deployment phases of the Network Application	Notified via email throughout the Network Application's deployment lifecycle
#12	Notifications about errors at any phase of the Network Application's deployment	Notified for potential errors via email throughout the Network Application's deployment lifecycle
#13	Access to test results	Selecting a completed Network Application deployment Service Order grants access to the CI/CD phase results
#14	Access to certification results	Selecting a completed Network Application deployment Service Order grants access to the certification phase results

There are two main procedures identified from Table 2, concerning the 5GASP Developer: the development, onboarding and management which is depicted in Figure 3 and the ordering, deployment, testing and potentially certification, which follow the triplet design (see next section) and is portrayed in Figure 4, respectively.

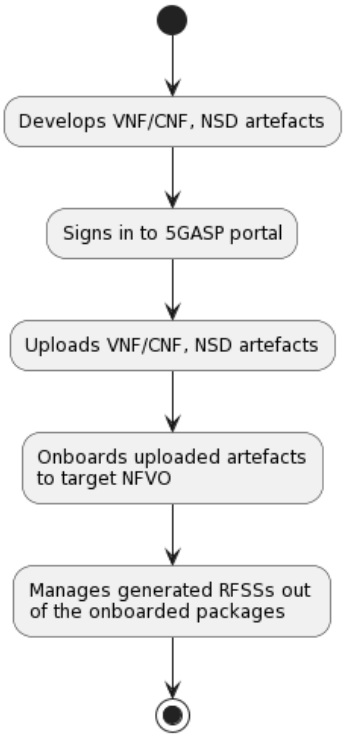

Figure 3: 5GASP Developer's onboarding procedure

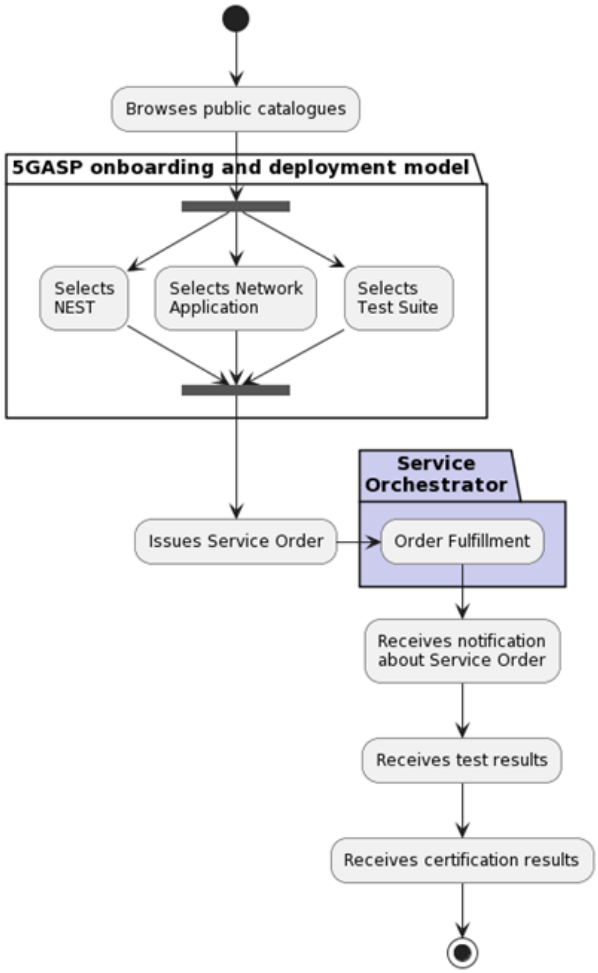

Figure 4: 5GASP Developer's ordering procedure

2.2.3 Service Designer

The Service Designer role refers to the provider of order-ready services composed by 5GASP triplet, i.e. network slice, Network Application, test suite. This role supplements the 5GASP developer offering an abstraction of the deployment artefacts towards a user-friendly executable bundle. It is also in charge for publication of the designed bundles in respective public catalogues, thus rendering them available for ordering from the verticals' end users. The supported use cases for the role are presented in Table 3.

Table 3: 5GASP Designer Use Cases

Use Case ID	Title	Description
#01	Account management	See UC #01 of 5GASP Developer
#02	Onboarded NFV packages management	See UC #04 of 5GASP Developer
#03	Network Application management	See UC #05 of 5GASP Developer

#04	Network Application designing based on other applications in the catalogue	Accesses the Network Application's catalogue and designs new CFSS bundles containing more than one Network Applications
#05	Service Test Specifications creation and test descriptors onboarding	See UC #06 of 5GASP Developer.
#06	Publication and support of services	Has the responsibility of publishing and supporting the offered services in catalogues
#07	Browsing the NODS catalogue of available services	See UC #07 of 5GASP Developer
#08	Network Application deployment, testing and certification request to NODS	See UC #08 of 5GASP Network Application Developer
#09	Service Order view and management	See UC #09 of 5GASP Network Application Developer
#10	Service Inventory view	See UC #10 of 5GASP Network Application Developer
#11	Notifications about the deployment phases of the Network Application	See UC #11of 5GASP Network Application Developer
#12	Notifications about errors at any phase of the Network Application's deployment	See UC #12 of 5GASP Network Application Developer
#13	Access to test results	See UC #13 of 5GASP Network Application Developer
#14	Access to certification results	See UC #14 of 5GASP Network Application Developer

The main process derived from the table above, concerning this role, is the service design procedure which follows the NFV package onboarding and precedes the service ordering. This process is depicted in Figure 5.

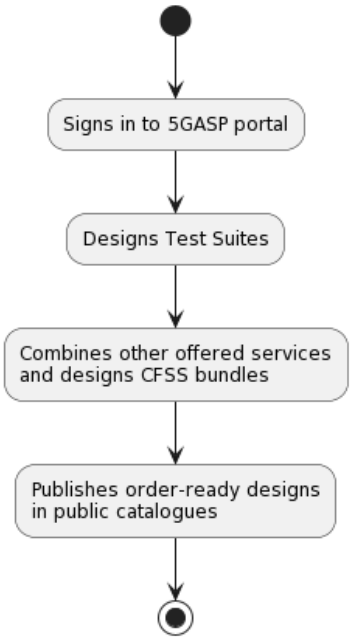

Figure 5: Service Designer's design procedure

2.2.4 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 4.

Table 4: 5GASP NODS Platform Administrator Use Cases

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 and product catalogues
#06	Category management	Can create, alter and delete service and product categories that are contained in specific catalogues
#07	Manage the platform's 5G services	Publicly exposes the offered network slices by the platform's testbeds

3 5GASP Network Application Onboarding and Deployment Services (NODS) final architecture and design

3.1 NODS architecture

This section, based upon the revised requirements that support the methodology of 5GASP as extracted form the use cases per actor of the previous sections (see section 2.2), presents the final internal design of 5GASP NODS. As it followed the principals of a service-based architecture, as depicted in Figure 6, employing a message bus system, it facilitated the everchanging topology throughout the project's duration. Its aforementioned modularity enabled numerous key changes at the architecture, which are further described at the respective following sections, and are briefly presented below:

- The introduction of the Network Application onboarding and triplet design portal
- The addition of Kroki
- The offering of the Network Orchestrator as-a-Service (aaS)
- The exclusion of the initial model transformation service

Specifically for the latter, the initial model transformation service envisaged the restructuring of YANG to TOSCA, and vice versa, so as to be universally handled by the 5GASP Service Orchestrator was excluded from the architecture. The reason behind it has been the continuously decreasing interest in the TOSCA-modeled ONAP Service Orchestration for the project's needs and its subsequent lack of support by the employed testbeds.

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

- 5GASP portals
- 5GASP experimentation APIs service
- Service registry
- Authentication service
- Kroki service
- Issue management service
- Central logging service
- 5GASP Service Orchestrator
- 5GASP Network Orchestrator
- MANO client API service
- Microservice bus

As already noted from 5GASP deliverable D3.1, the 5GASP NODS is based on the open-source project Openslice [11]. Since then, 5GASP among other initiatives and projects facilitated Openslice to become the first Software Development Group (SDG) under the umbrella of ETSI, named ETSI SDG OSL [27]. Openslice, and as a consequence the 5GASP NODS platform, employs a service-based architecture and design which utilizes industry-wide standards to deliver Network-as-a-Service (NaaS). More details can be found in the previous versions of this document, thus omitted.

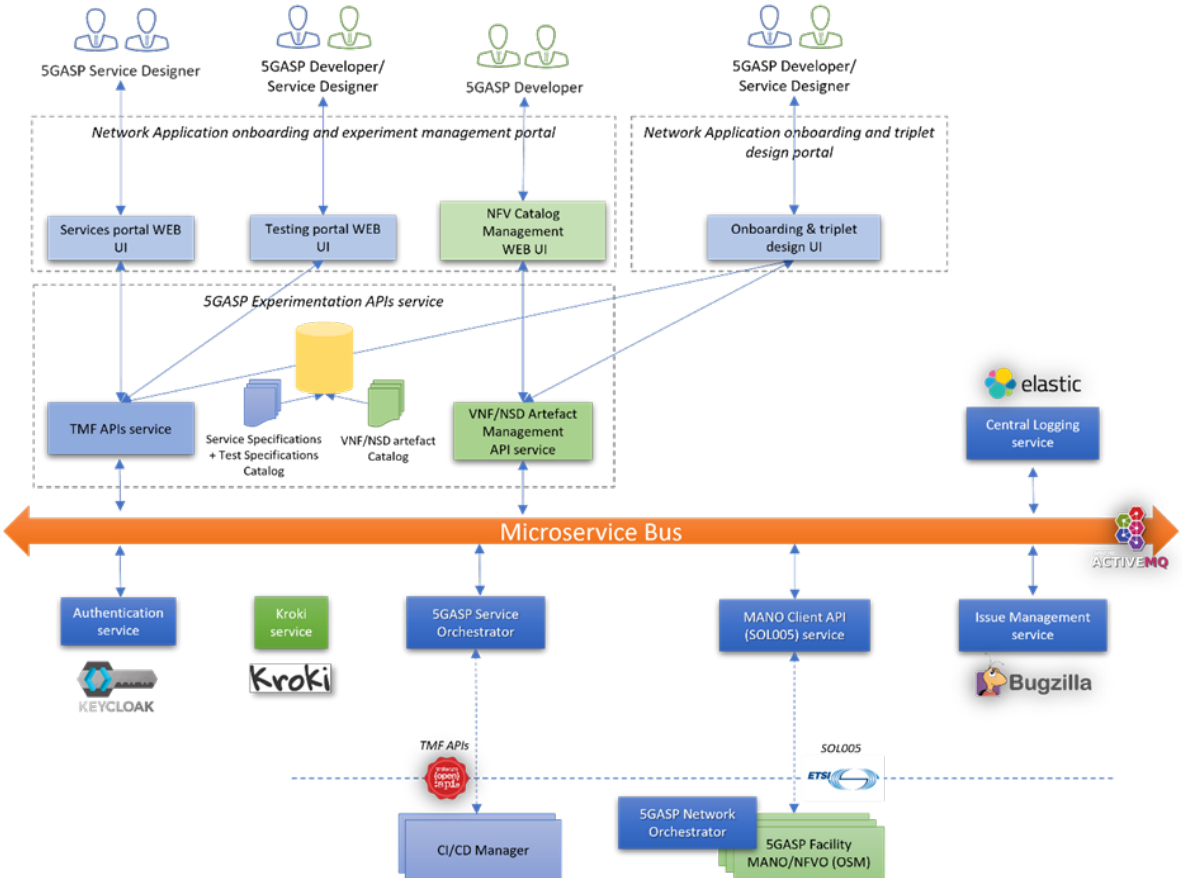

Figure 6: 5GASP NODS final architecture

3.2 Internal services

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

3.2.1 5GASP portals

The 5GASP portals comprise the 5GASP Network Application and experiment management portal and the Network Application onboarding and triplet design portal, as represented in Figure 7.

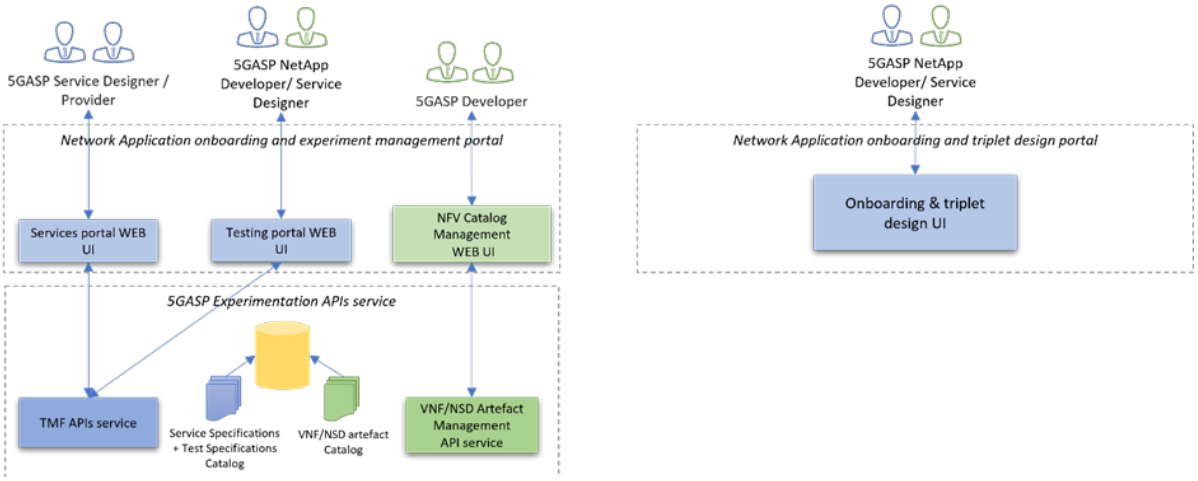

Figure 7: 5GASP portals

The 5GASP Network Application and experiment management portal was initially designed to provide a single entry-point to relevant actors. The portal comprised of three discrete User Interfaces (UI), each facilitating the requirements of the identified actors, as follows:

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

As already mentioned in D3.1 and D3.2, the NFV catalogue management UI's primary usage is to serve as a portal that enables the 5GASP developer to onboard and manage the Network Application package, i.e. NFV artefacts. Following, the services portal web UI allowed the Service Designer to leverage the automatically constructed RFSSs, out of the onboarded artefacts, combining them to design CFSSs, which are then exposed to the relevant catalogues constituting the Network Application entity of the onboarding model (see Section 2.1). Last but not least, the testing portal UI addressed the need to design tests, bundle them in a standardized manner and render them available for execution. More exhaustive details about the aforementioned portal can be found in D3.2.

Although, the previously described portal met all the requirements appointed by the relevant actors and enabled their emerged needs to i) manage NFV artefacts, ii) design offered services, and iii) design offered tests, it diverged from the initial all-inclusive portal concept and followed a "divide and conquer" approach, offering holistic support at each aspect through an exclusively designed UI. Even though, each UI could fully cover its domain of expertise, e.g. NFV artefacts management, service, and test design, it also imposed a notable learning curve to the inexperienced used to interact with the 5GASP portal. Therefore, a new universal portal was designed and introduced with the purpose of fulfilling the initial goal of a comprehensible single entry-point, namely the Network Application onboarding and triplet design portal. This UI, as seen in Figure 8, offers a cognitive step by step approach to the involved stakeholder enabling the NFV management, service and test design through a common UI, immensely augmenting its interaction with the 5GASP platform.

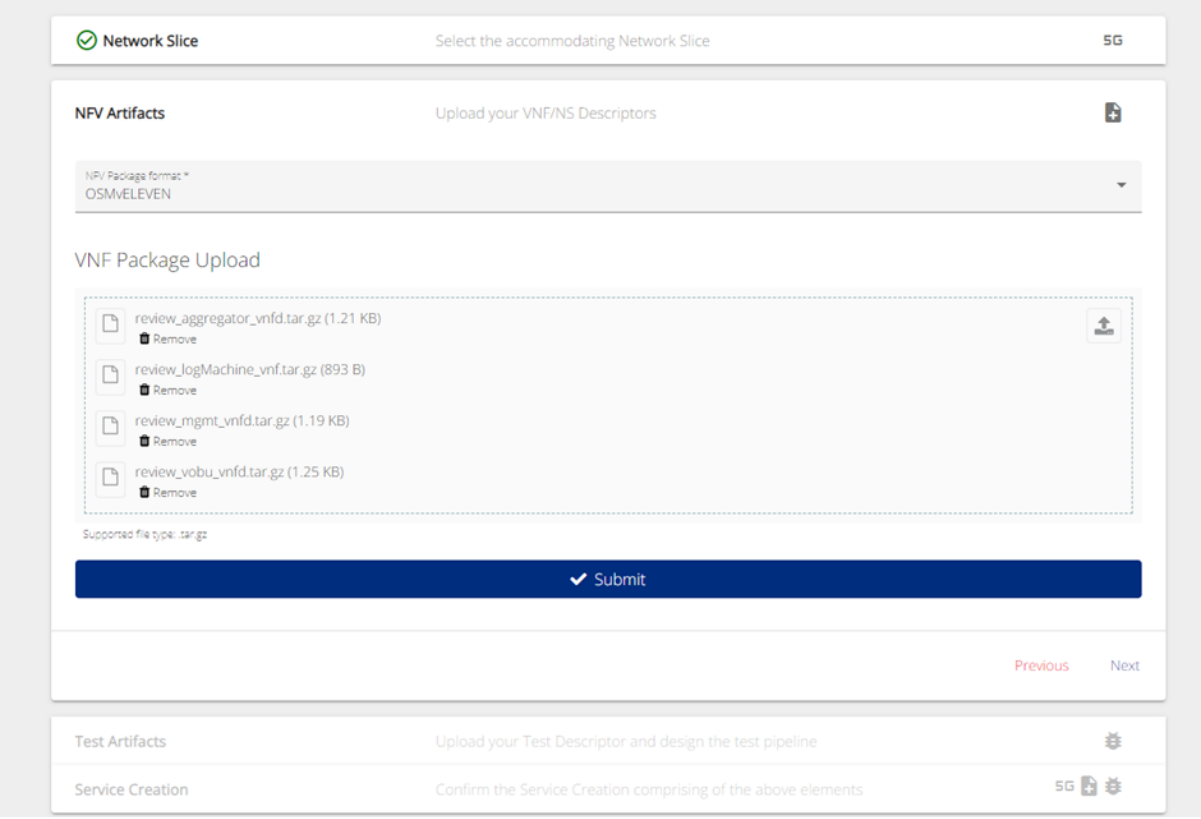

Figure 8: Network Application onboarding and triplet design portal

3.2.2 5GASP experimentation APIs service

The 5GASP experimentation APIs service retained its functionality in both of its composed segments, i.e. NFV Artefact Management API and TMF family APIs, throughout the last cycle of the project, as depicted in Figure 9.

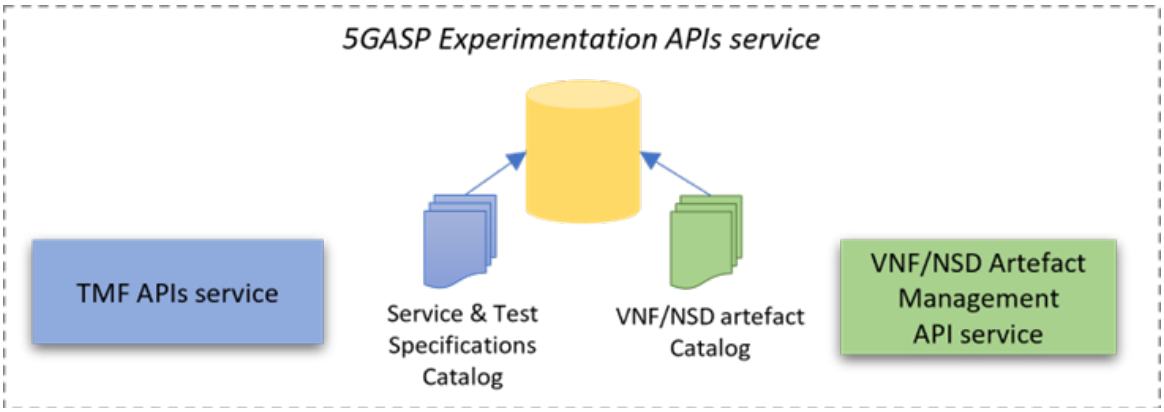

Figure 9: 5GASP Experimentation APIs service

The introduction of the Network Application onboarding and triplet design portal, as described in the previous subsection, while leveraging both the service's segments did not require any

alterations, hence their complete description can be found in the previous version of this document.

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. It is based on Consul [12], while more details can be found in D3.1.

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. The solution based on Keycloak [13] has remained unchanged and additional information can be found in D3.1.

3.2.5 Kroki service

While the project progressed, it has been observed that the complexity of the designed service bundles increasingly escalated, rendering them difficult to comprehend without a native visual representation. For that reason, NODS integrated Kroki as an open-source standalone service to provide a comprehensible view of the design service bundles. Kroki generates digestible diagrams that can be used to overview relationships among services and/or service order fulfilment flows, as seen in Figure 10.

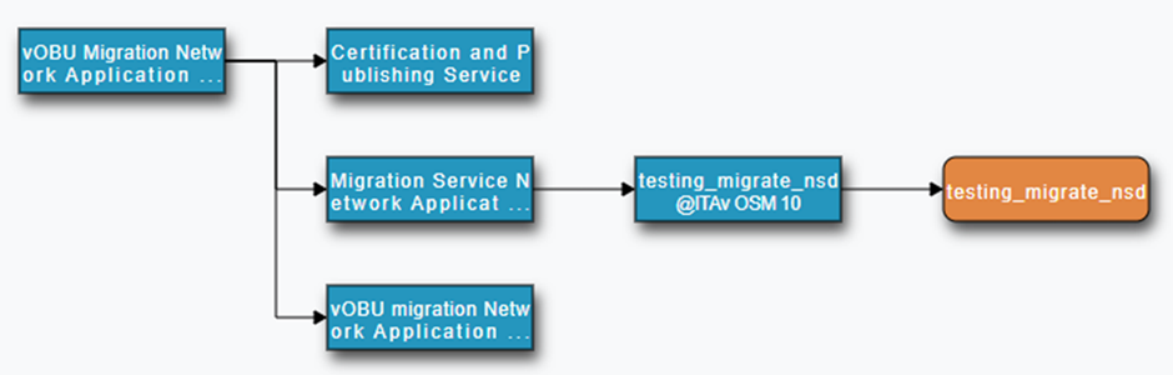

Figure 10: Service relationship Kroki representation

3.2.6 Issue management service

The issue management support of NODS relies on Bugzilla [14]. NODS encompasses an inherent Bugzilla client that utilizes the known ticketing tool to track platform requests and notify the respective stakeholders, as depicted in Figure 11. Extended details on the service can be found in D3.1.

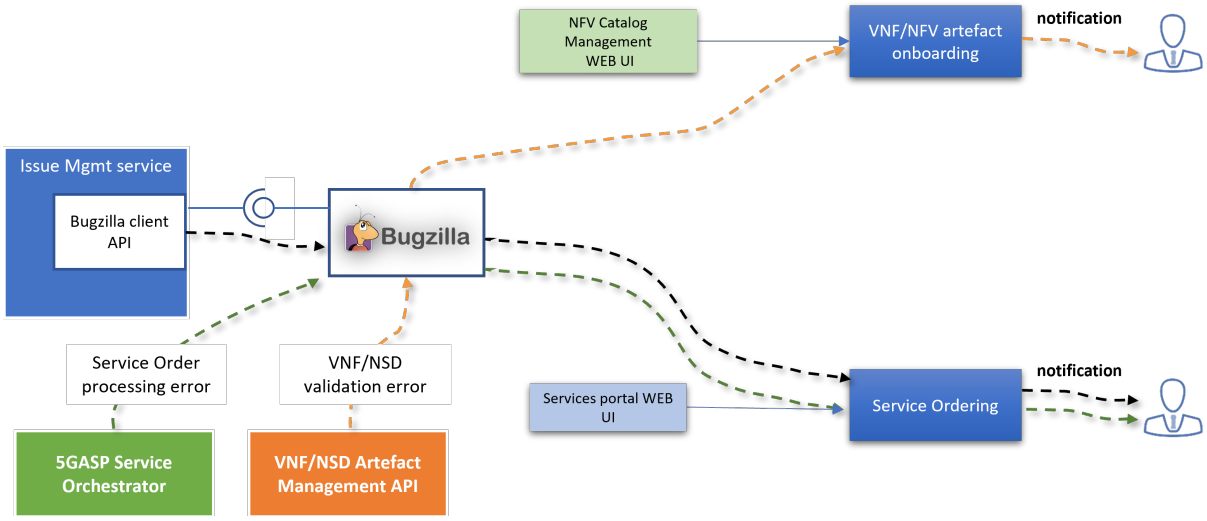

Figure 11: Issue management service

3.2.7 Central logging service

NODS embraces 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 solution. The solution is based on Elasticsearch [15] and was presented in D3.1.

3.2.8 5GASP Service Orchestrator

The 5GASP Service Orchestrator is the principal components of the 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 in D3.1, the 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.

The 5GASP Network Orchestrator underwent major additions and reconstructions during the project's lifetime and took its final shape during the second cycle of development, when it encapsulated the CI/CD automation circle to enable the orchestration of the testing artefacts. Also, due to the heterogeneous needs of the onboarded applications it soon became evident that dynamic orchestration patterns should also be supported. Hence, several landmark phases were introduced, where the aforementioned orchestration could be injected dynamically. These landmark phases derived from Network Slice Instance lifecycle as defined by 3GPP [16]. More information can be found in D3.2.

3.2.9 5GASP Network Orchestrator

As already addressed in D3.2, the Network Orchestrator (NetOr) is responsible for deploying and configuring a Virtual Private Network (VPN) tunnel between the independent administrative domains, creating a secure communication channel. Therefore, it is through NetOr that 5GASP achieves multi-domain scenarios. To achieve this, NetOr relies on a Service

Discovery approach, where all VPN nodes advertise their location and public key. This information is then propagated to the remaining peers, which will use it to establish the VPN tunnels. Since NetOr's status is the same as described in D3.2, in D3.3, we won't delve further into this 5GASP component. For more details on NetOr, you can refer to D3.2.

3.2.10 MANO Client API service

As already presented in the previous versions of this document, the MANO client API service is the intermediate component that facilitates the communication between NODS and the underlying MANO components, which reside in the engaged facilities. This component relies on ETSI SOL005 interface [17] and several plugins were developed to support different OSM versions. Due to the standard interface, it has been tested with OSM 9-13 throughout the project's duration with minimum to zero alterations. D3.2 includes more technical details on this component.

3.2.11 Microservice bus

NODS is based on a service-based architecture thus, a messaging bus is needed to support the internal exchange of messages between micro-services in a loose-coupling manner. The implementation is based on ActiveMQ [18], thus enabling the elasticity on the selection of the programming language for the development of the included micro-services.

3.3 Northbound standardized interfaces

A core objective of the 5GASP NODS is to offer standardized northbound solutions for experiment requests realizing the envisaged triplet model (see Section 2.1), achieved by encapsulating models from the TM Forum, a major Standard Development Organization (SDO). Each component of the onboarding triplet (Network Application Artefact, NEST, Test Descriptor) is categorised in a relative catalogue and made publicly available. Network Applications and NESTs are directly expressed as Service Specifications within Service Catalogues based on the TMF's resource model, offering additional grouping through underlying categories. The last triplet entity, i.e. the test descriptor, is captured and designed under the TMF's Service Test Specification [19] which incorporates the testing artefacts along with the needed configuration input and constitutes the TMF's nominated model to perform testing against the designed services. The defined TMF modelling [19] also supports a direct link of Service Specifications with Service Test Specifications, so inevitably the testing entity's final exposure is achieved through the former model entity to completely match the rest of the triplet's structure, as illustrated in Figure 12. This level of uniformity also natively enables the bundling of the matching triplet's entities under the same high-level Service Specification, clearly representing the overall 5GASP cycle, namely the network slice and Network Application deployment, followed by the test execution and certification.

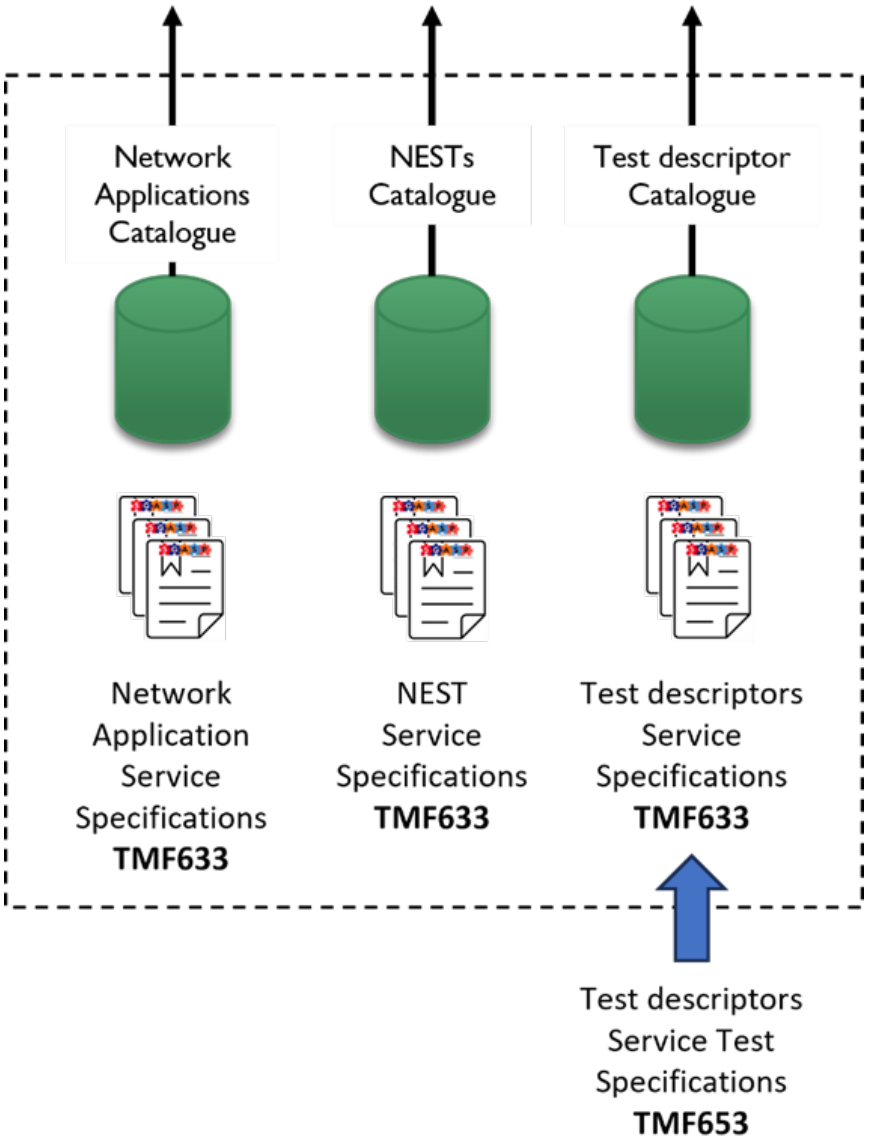

Figure 12: Northbound standardized resource models

4 Interaction with 5GASP ecosystem

This section contains 5GASP's high level architecture, defining interfaces (high level architecture Figure 13).

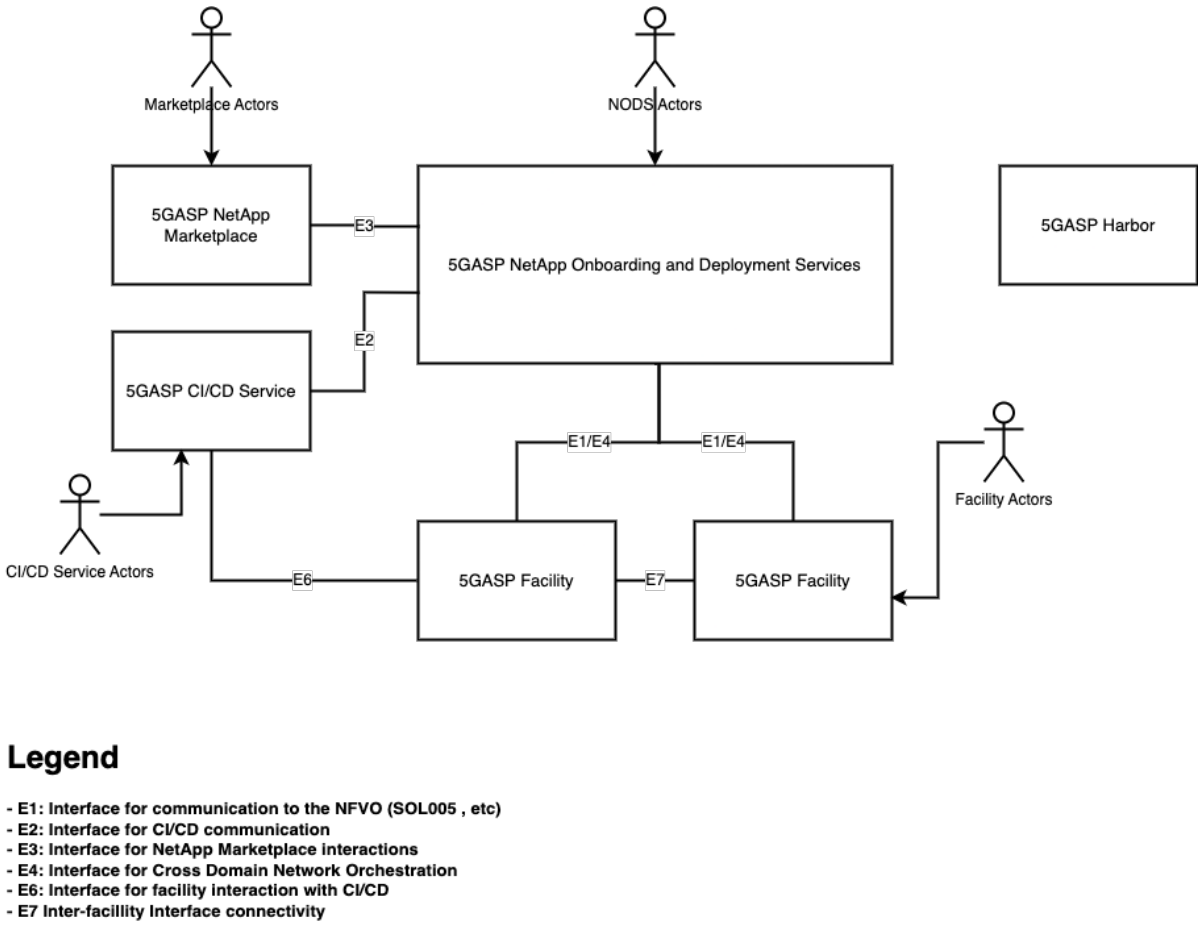

Figure 13: 5GASP High Level Architecture

4.1 Interaction with CI/CD service – Interface E2

D3.2 extensively discusses the interaction between NODS and the CI/CD Service, specifically the CI/CD Manager. It covers the CI/CD Service's design, architecture, and the exchanged data models. Additionally, D5.1 and D5.3 heavily address the interaction between the CI/CD Manager and the CI/CD Agents within 5GASP facilities. In summary, the CI/CD Manager serves as the central coordinator for all testing and validation tasks, assigning and distributing tasks to CI/CD Agents deployed in the testbeds. To enhance interoperability, the CI/CD Manager implements the TMF 653 specification [19], enabling a standardized communication with the NODS. The integration between the CI/CD Manager and the NODS was thoroughly tested within the 5GASP ecosystem by Q1 of 2022.

5GASP's CI/CD Service is triggered by the NODS through the submission of a TMF 653 Service Test payload to the CI/CD Manager, illustrated in Figure 14. The Service Test payload provided by the NODS contains all necessary information to initiate a new testing and validation process. While some information is directly included in the initial payload, other is referenced via URLs. For instance, the initial payload includes a set of test characteristics, such as the IP addresses of Network Application's VNFs, which alone are insufficient for performing the validation task. Therefore, it is required that the CI/CD Manager processes these characteristics and

afterwards requests the missing information from the NODS. It is through this process that the CI/CD Manager obtains the Testing Descriptor, which is then used, alongside the initial Service Test characteristics, to generate the testing pipeline configuration that will guide the entire testing process.

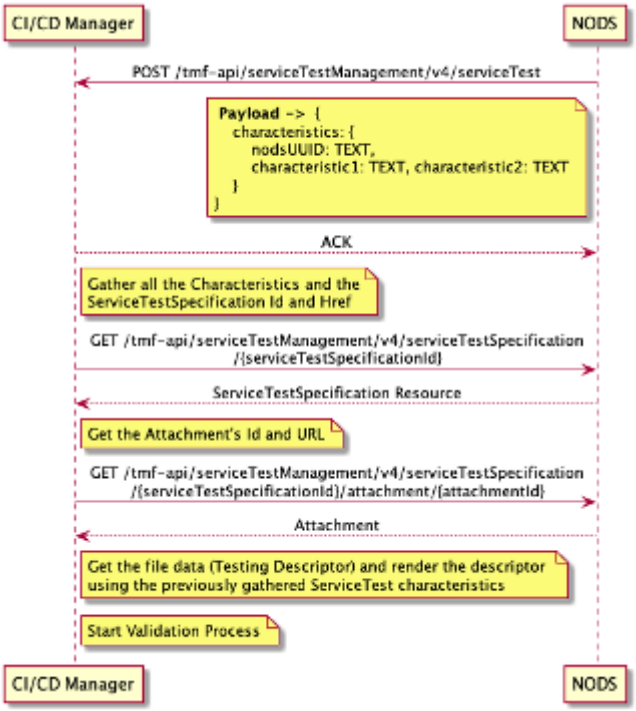

Figure 14: TMF 653 Payload Submission Workflow

Furthermore, the Service Test Payload may also include URLs to obtain developer-defined tests, should a Network Application developer onboard them. In such cases, the CI/CD Manager requests these tests from the NODS and makes them available to the CI/CD Agents. The workflow that describes the gathering of the developer-defined tests can be observed in Figure 15.

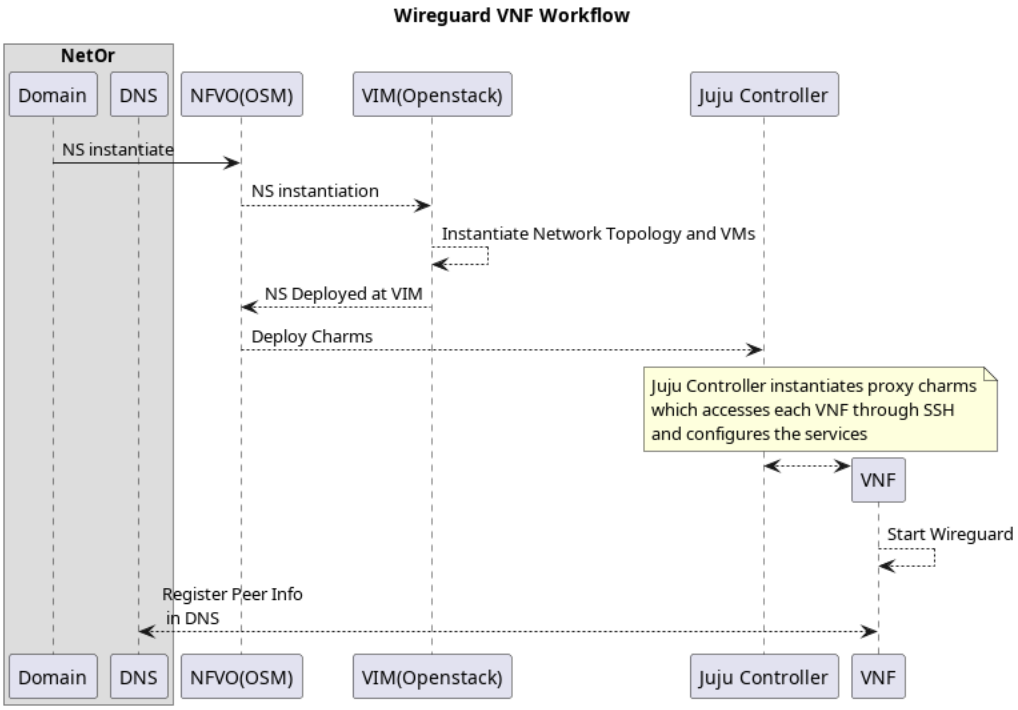

Figure 15: Developer-Defined Tests Gathering Workflow

4.2 Interaction with Facilities – Interface E1/E4/E7

4.2.1 Communication with the NFVOs - Interface E1

This interface took into consideration the different technological stacks deployed in each facility, while also attempting to converge with available standard interfaces, so as to seamlessly interconnect the 5GASP NODS platform with the underlying facilities. In the previous version of this document, namely D3.1, we have identified four available options regarding the targeted interconnectivity. Each option is thoroughly introduced at the aforementioned document, so a summary is presented 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

From the project's dawn, it was made apparent that the maintenance of the facilities would propose a great challenge should we followed an approach that was not based on uniformity. Following timely trends, all facilities were familiar with and incorporating OSM [20] for their MANO stacks. Therefore, this facilitated the initial NFV-based concept of the project and also began natively integrating state-of-the-art cloud-native deployments acting as a Kubernetes proxy, applying the desired future proofness to 5GASP's endeavour. Consequently, Option B was elected as the most convenient, simultaneously minimizing the maintenance effort and update risks, as NODS natively integrated and extended a SOL005 compliant MANO Client, as presented in Section 3.2.10. The nominated scenario is illustrated in Figure 16.

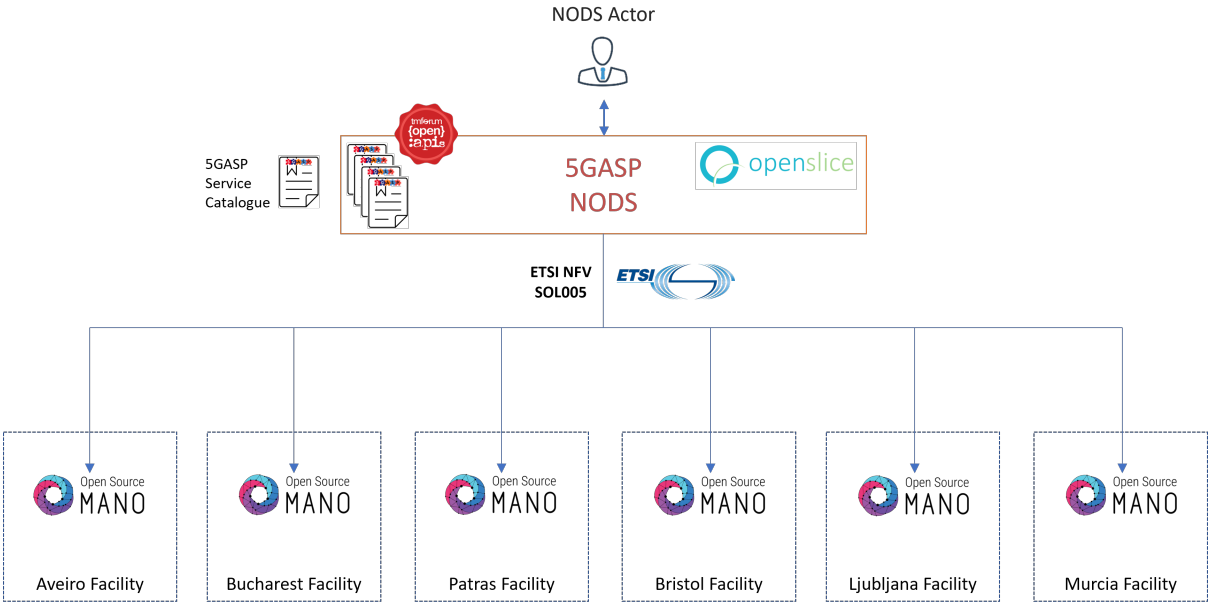

Figure 16: NODS interconnectivity with facilities

4.2.2 Cross Domain Network Orchestration - Interface E4

In D3.1 and D3.2, Interface E4 was already introduced as the designated interface for Cross-Domain Network Orchestration. This interface primarily serves the purpose of enabling ITAv's Network Orchestrator to deploy Wireguard servers at the consortium's testbeds, enabling the inter-domain scenarios in 5GASP. Until the submission date of this deliverable, numerous experiments have been performed to further refine the inter-domain requirements, aiming for the seamless orchestration of inter-testbed VPN tunnels. Through these experiments, 5GASP managed to interconnect all project testbeds through its inter-domain solution. For more details, please refer to D3.1 and D3.2.

4.2.3 Inter-facility connectivity - Interface E7

After the orchestration of the Wireguard VPN tunnels through the E4 Interface, one may rely on Interface E7 to achieve inter-domain connectivity. While at the time of submission of D3.1 5GASP only had 3 interconnected testbeds, since Q3 of 2022, we have achieved a full coverage of all 5GASP's testbeds. This means from Q3 2022 onwards all consortium's testbeds are interconnected and can cope with inter-domain scenarios. The testbeds interconnection scenario is presented in Figure 17. For more information on the E7 Interface, please refer to D3.1 and D3.2.

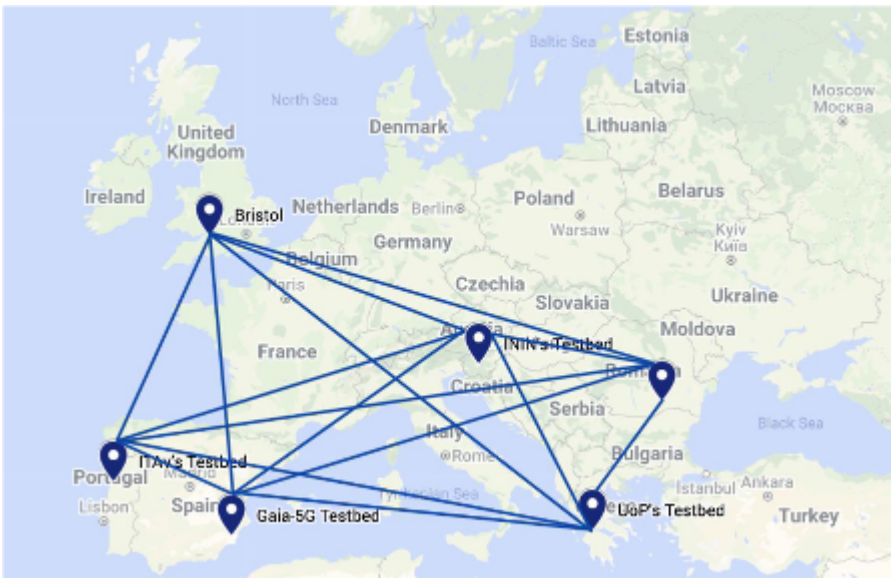

Figure 17: Testbeds Interconnection Scenario

4.3 Interaction with 5GASP Marketplace – Interface E3

This interface is utilized at the final stage of the 5GASP pipeline, namely the publication of a certified Network Application at the respective Marketplace, and by external stakeholders that would like to browse the repository. Likewise the other publicly available interfaces, the Marketplace interaction is also based on TMF API and resource model, i.e. TMF's Product [21], as substantially described in previous versions of this document. Moreover, the utilization of the aforementioned entity not only maintained the consistency between the ordering and deployment model, but also facilitated the introduction of business aspects while imposing an abstraction layer between the customer and the service provider.
This global adoption of standardized interfaces and resource models, along with the promotion of the 5GASP's Marketplace within the project's communication channels has led into the extension of the latter into a cross-project ICT-41-wide Marketplace for Network Applications, with the active incorporation of at least four projects' assets. More details on this endeavour can be found in D6.3, which share the same publication date with this document.

4.4 Interaction with 5GASP Harbor

To seamlessly follow the uprising tendency towards the cloud-native Network Applications, 5GASP had to incorporate its solution of supporting the respective artefacts, i.e. Helm charts and images. Although the common practice imposes the utilization of publicly available repositories, 5GASP intended to provide its own project-wide hosted solution. This decision has made an exceptional impact on attracting commercial and intellectual property rights (IPR)-protected Network Applications, while also minimizing the maintenance effort of separately updating all the employed facilities, in case of newly introduced artefacts, by providing a single source of truth. The described 5GASP's solution is based on the open source registry Harbor.

Therefore, the 5GASP Harbor artefact repository is an important element of the project's ecosystem as it enhances the security and management of CNF-based applications' digital artefacts, similar to the Network Service Repository from the 5G-EPICENTRE Back-End layer, as noted in D2.3. 5GASP Harbor enforces access control policies and role-based permissions on the Network Application's container images and stores the related Helm chart install packages, providing methods for uploading, retrieving, updating and deleting them. It also scans Network Application packages for critical security vulnerabilities and verifies the authenticity of a signed application.

As already mentioned, a number of the commercial or license-restricted Network Application developers wanted to ensure that there was non-public access to their respective applications' images and this requirement was substantially facilitated by the 5GASP Harbor repository, as it provided a private image hosting environment. However, at the time of writing, there has been an explicit technological restriction with the OSM versions used by the 5GASP facility providers' networks, which would deprive 5GASP of the use of private Helm chart repositories, as well. Therefore, a public Helm chart repository had to be introduced to overcome the aforementioned restriction, in parallel with its private counterpart for the respective images. This approach enabled the desired secure environment for the Network Applications' source code, i.e. the application's images, whereas facilitating the OSM's access to the Helm chart repository. The latter is figuratively presented in Figure 18.

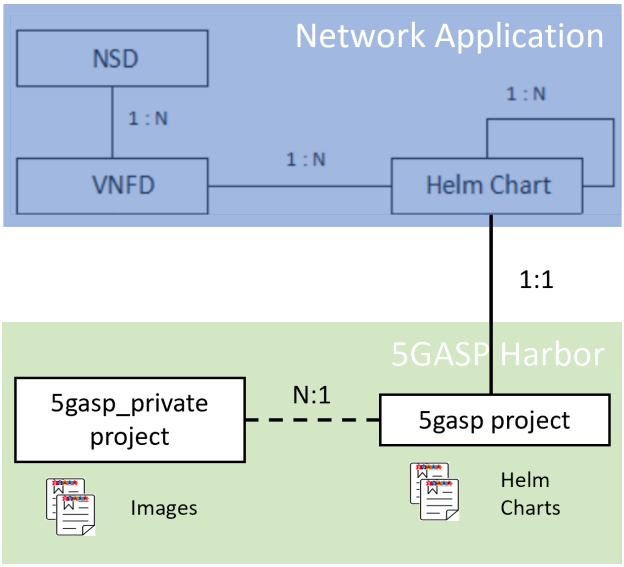

Figure 18: CNF-based Network Applications hosting in 5GASP Harbor

In summary, the 5GASP Harbor comprises of the following projects:

- "5gasp" - a publicly accessible Helm chart repository
- "5gasp_private" - a private image repository

The above are illustrated in Figure 19 and Figure 20, respectively.

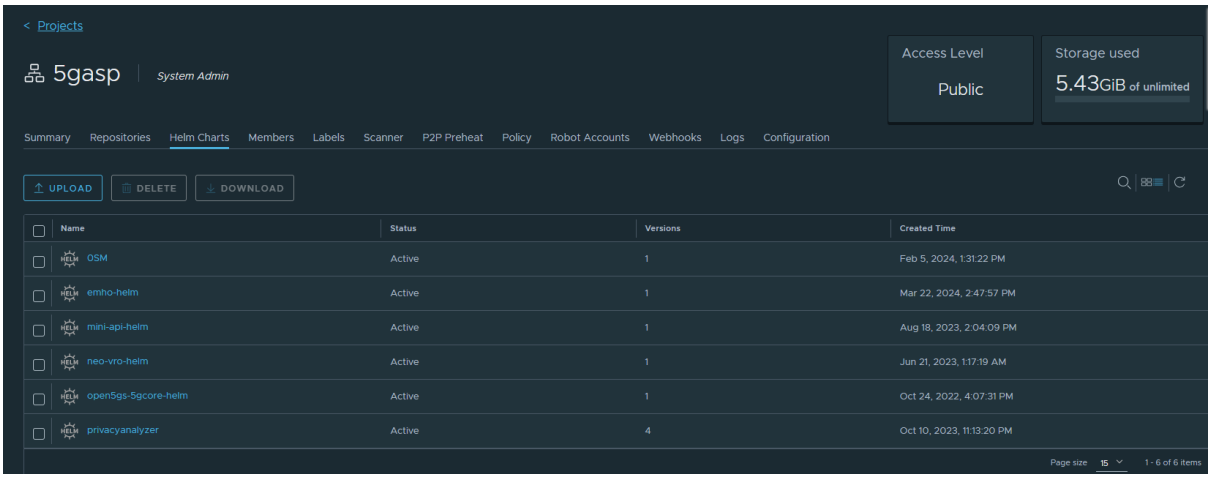

Figure 19: 5gasp project in 5GASP Harbor

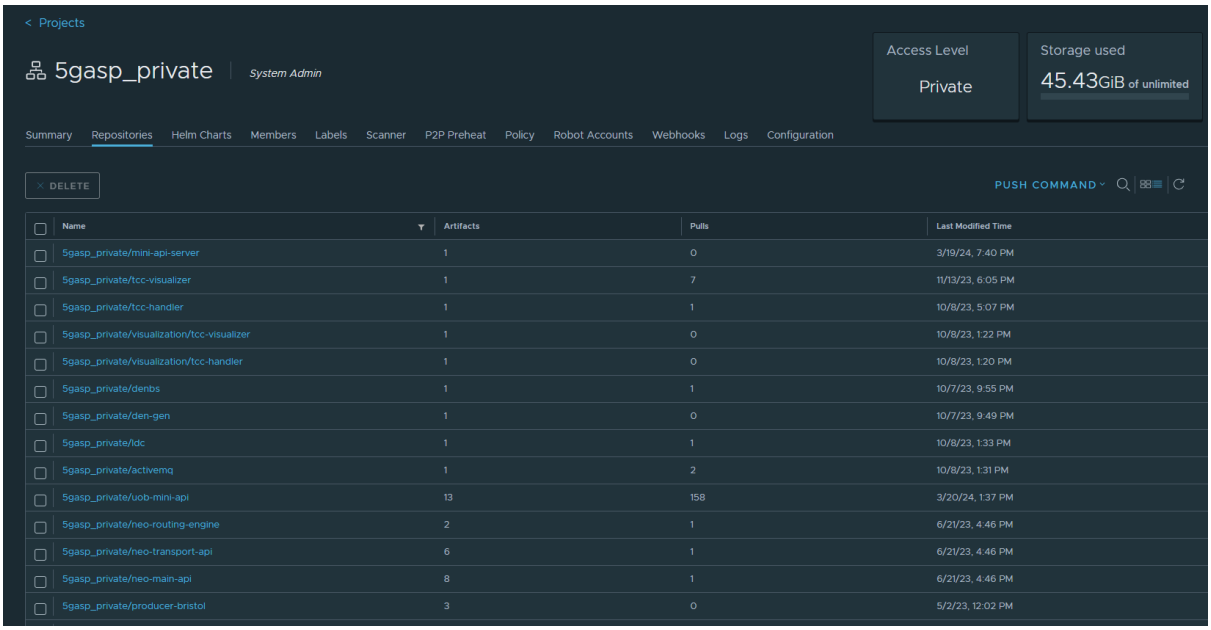

Figure 20: 5gasp_private project in 5GASP Harbor

5 5GASP NODS implementation Final Release

This section is dedicated to final implementation aspects based upon the requirements, architecture and interaction of NODS with the overall 5GASP ecosystem introduced in the previous sections. Specifically, it reflects the final version of the experimentation services regarding the 5GASP architecture validation, i.e. facility management, Network Applications and tests onboarding and execution, certification and publication. The section primarily focuses on the implementation of the latest phase of the project, while the reader could find more details about the intermediate phases in D3.1 and D3.2.

To begin with, the capability to manage the underlying facilities, offered to NODS Platform Administrator, is presented in Figure 21. The figure illustrates the overall 5GASP facility during the last phase of the project, with all its testbeds incorporated also maintaining the bidirectional synchronization capability. The latter enables the NODS Platform administrator to onboard VNF/NSD artefacts to any underlying facility, while also updating the NODS portal for any changes directly executed at the facility by updating the corresponding *onboarding status* field of the respective artefacts' list, as seen in Figure 22.

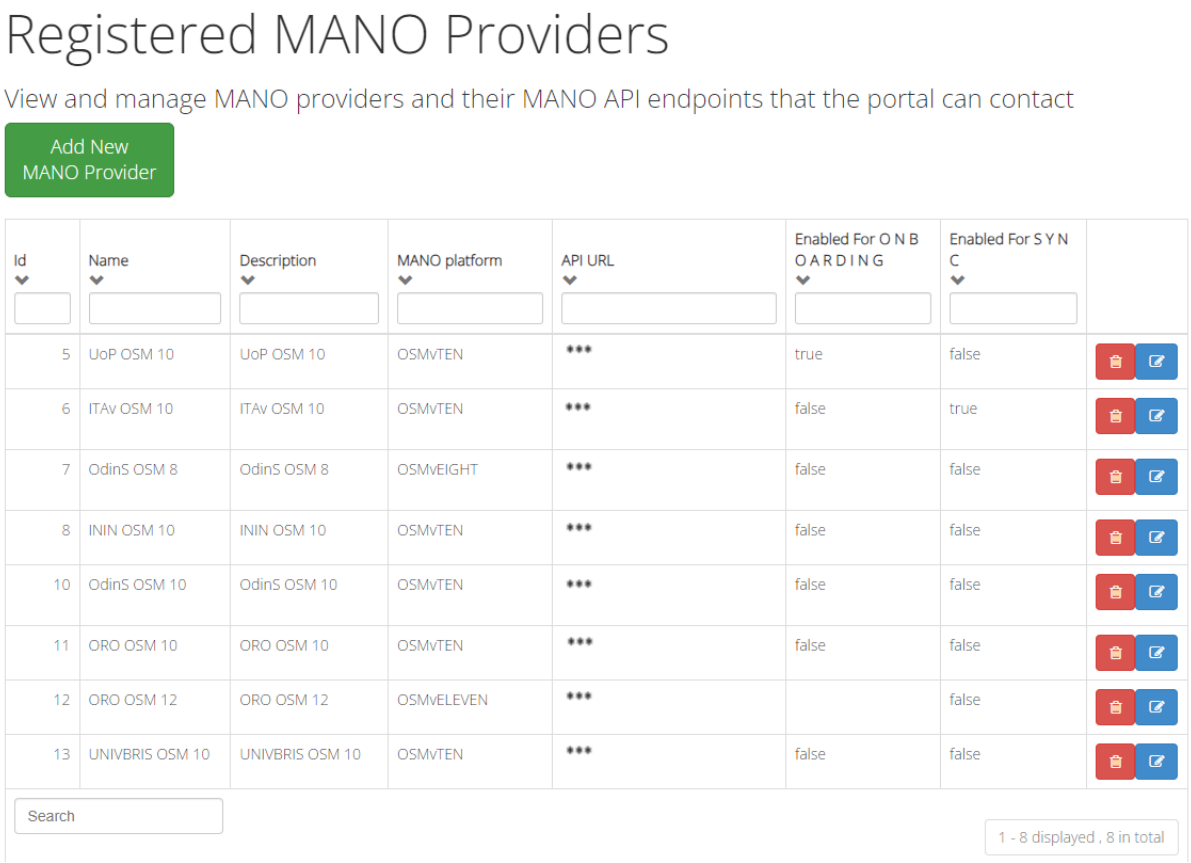

Figure 21: Facility management UI (final)

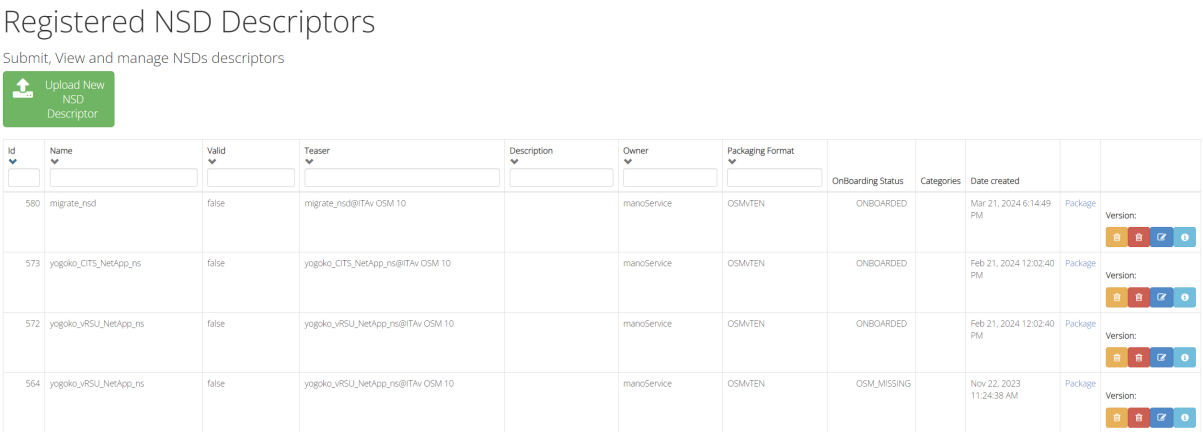

Figure 22: Onboarded NSD artefacts listing UI (final)

Regarding the facility management, the last alteration can be indicated with the incorporation of the *status* and *creation date* fields to the synchronized Virtual Infrastructure Managers (VIMs), so as to distinguish between any deprecated infrastructure and avoid potential deployment errors. This is depicted in Figure 23.

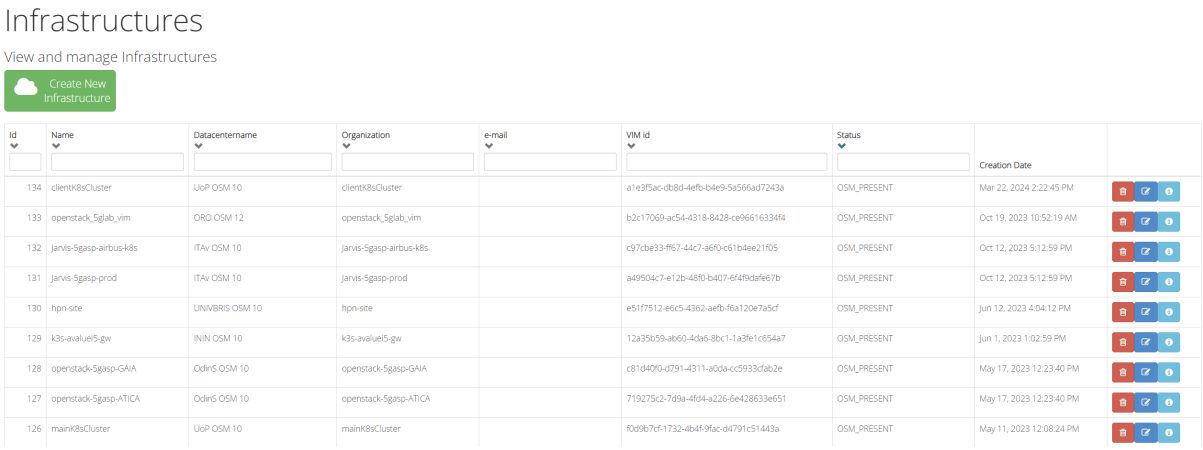

Figure 23: VIM listing UI (final)

Regarding testing enablement, a discrete testing portal UI within NODS holistically addresses the need of test designing, testing artefacts uploading (see Section 2.1.3) and test execution overview. Specifically, the designated UI for the test designer to create the respective test suites was expanded with the capability to onboard any potential developer-defined tests, along with the respective test descriptor, as seen in Figure 24.

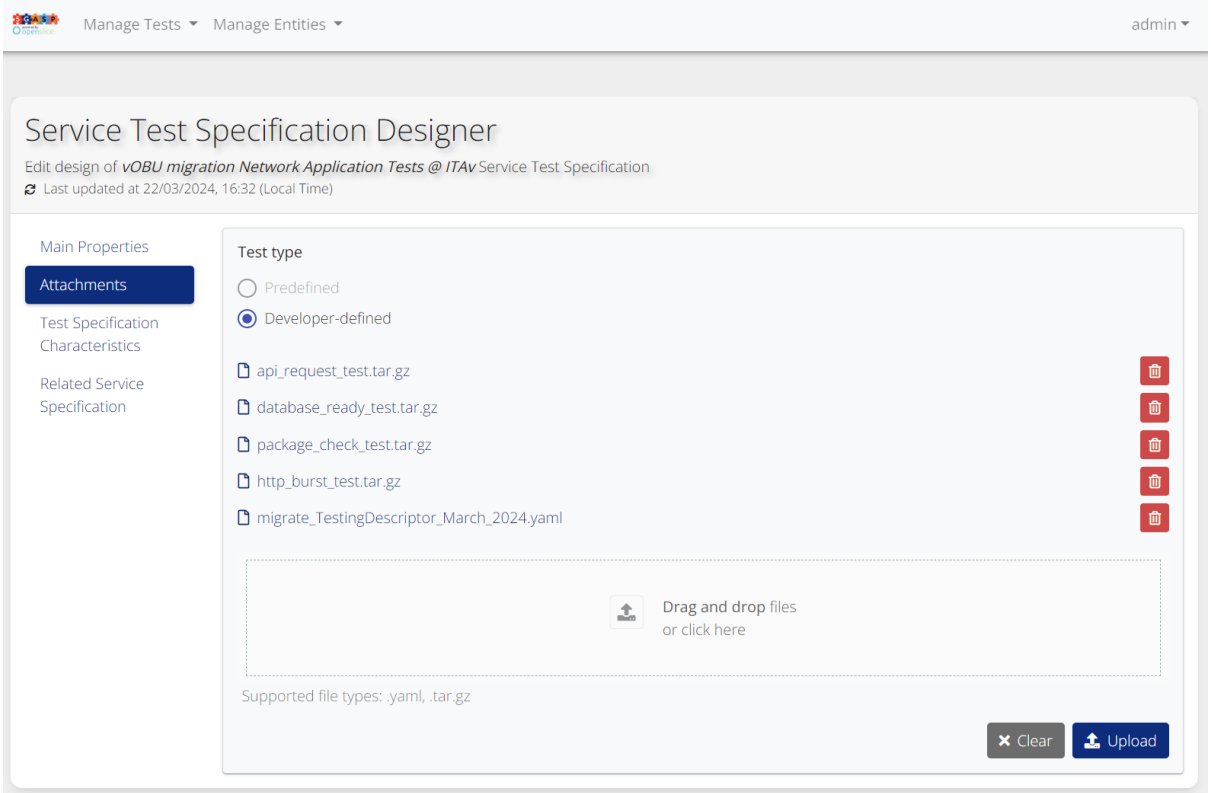

Figure 24: Test specification designing UI (final)

Regarding the service-side of the NODS portal offered by the homonymous service portal UI, all the publicly available services are showcased in the respective catalogues, e.g. network slices, network applications, testing suites, combined Triplets, etc. For instance, the 5GASP Network Applications are illustrated in Figure 25, whereas Figure 26 presents the onboarded Network Applications of the ICT-41 Marketplace.

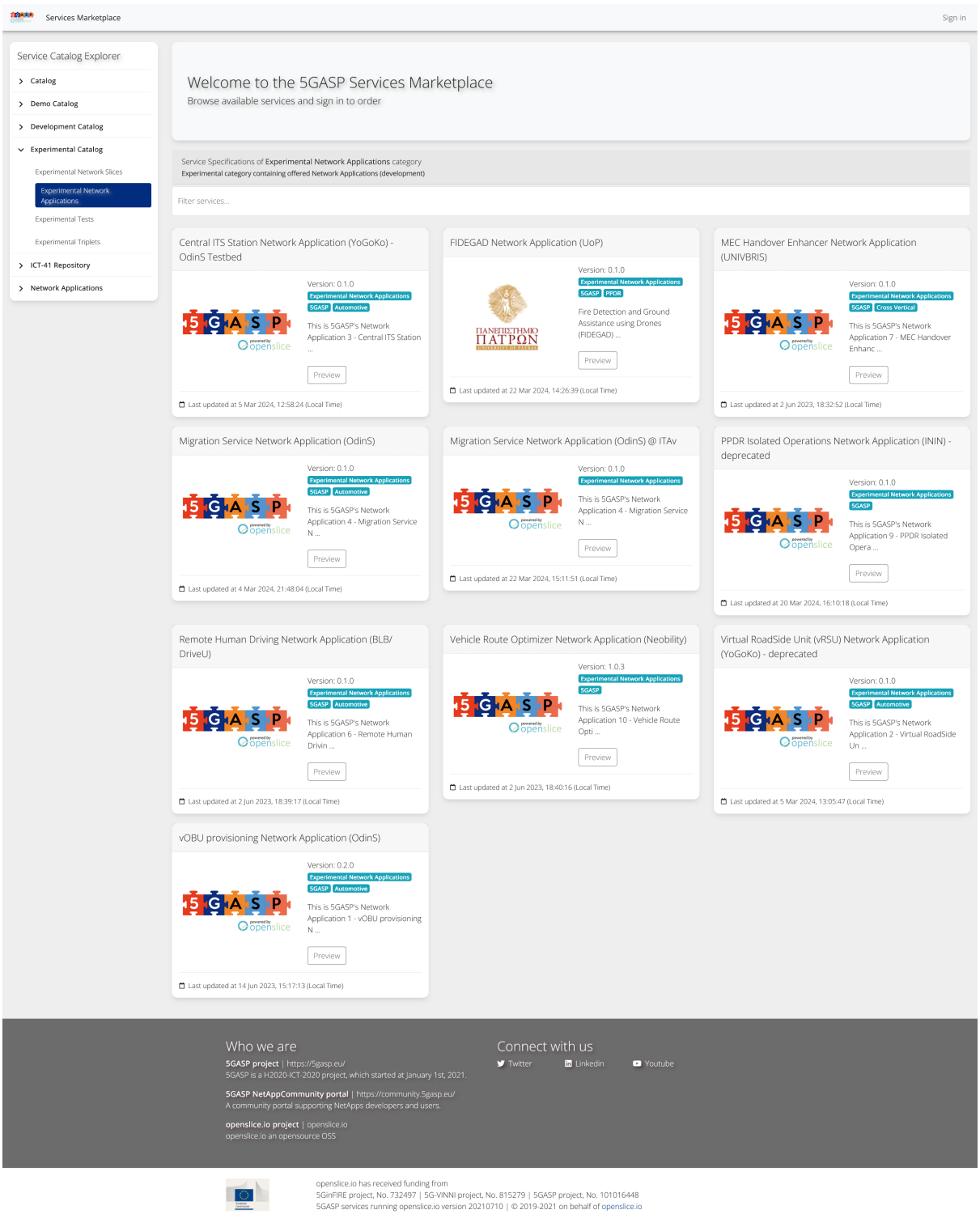

Figure 25: 5GASP Network Applications listing UI (final)

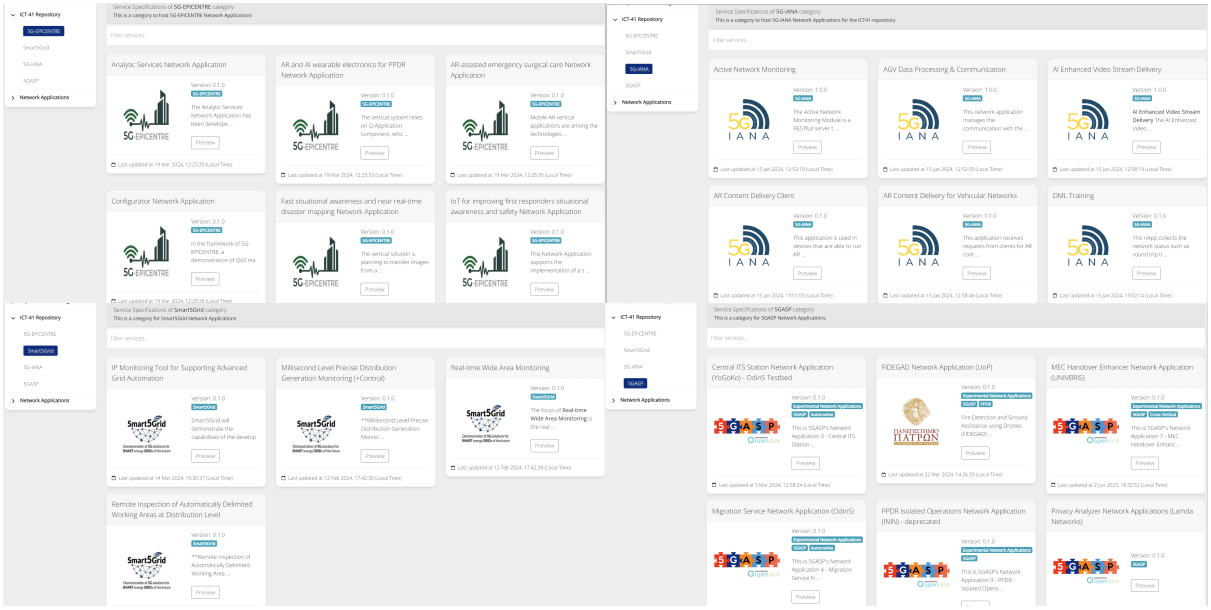

Figure 26: ICT-41 Marketplace UI (final)

Moreover, the service deployment overview was enhanced to graphically represent the deployed services' relationships and dependencies, as seen in Figure 27. This further extends the requester's ability to comprehend the execution flow, which at this example starts with the Network Application deployment followed by the replicable automated deployment information extraction implemented by the rule engine (see Section 3.2.8), as displayed in Figure 28. Only after the deployment information is available, the flow is handed over to the CI/CD Manager to conduct the specified test suite. Once the test suite is successfully performed a URL is made available through the UI (Figure 29) that leads to the test results report (Figure 30).

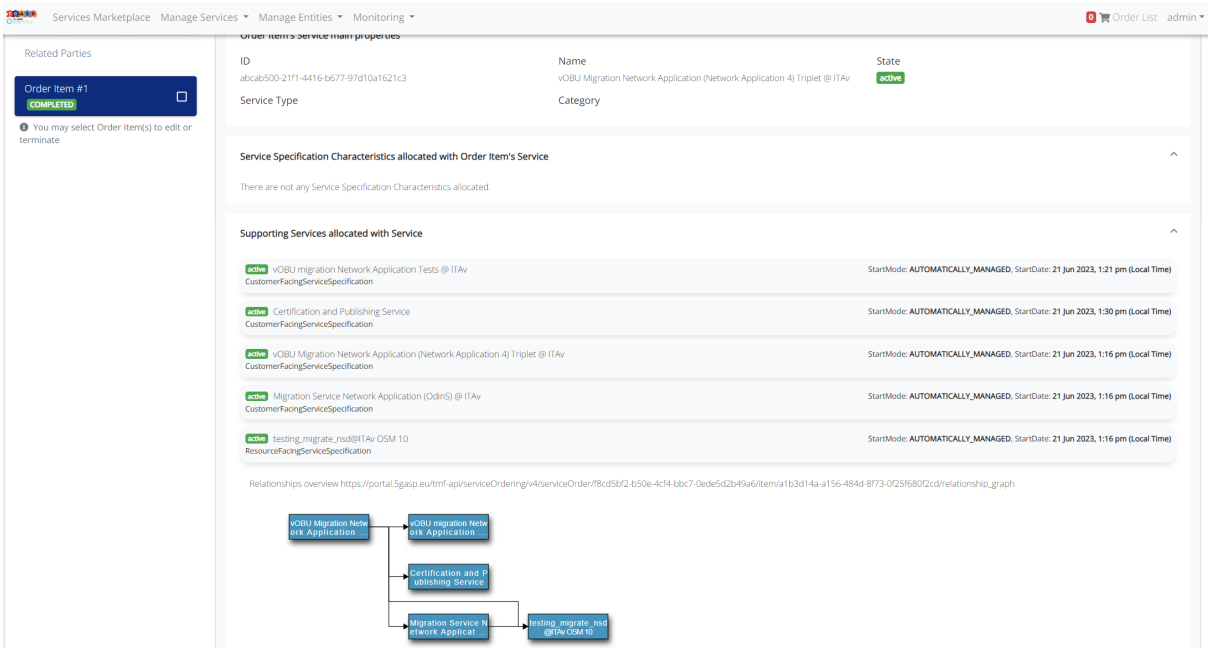

Figure 27: Deployment overview UI (final)

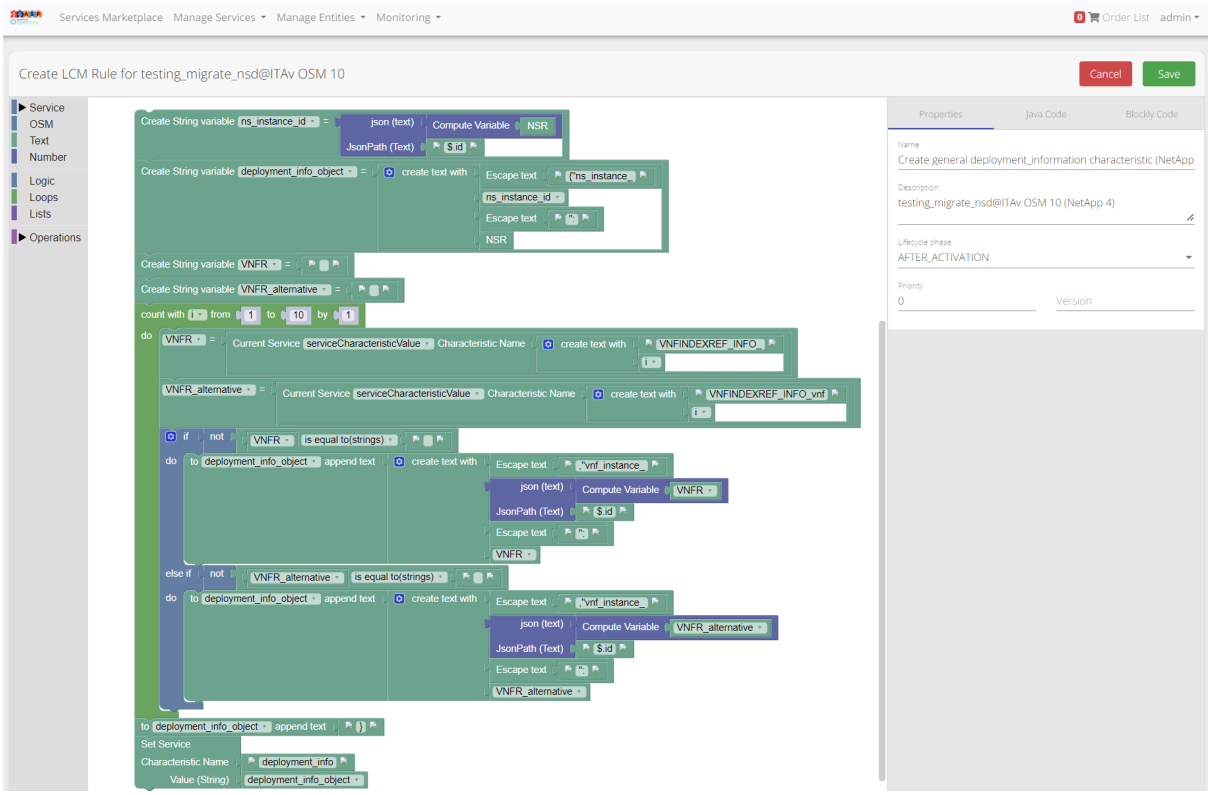

Figure 28: Replicable deployment information extraction

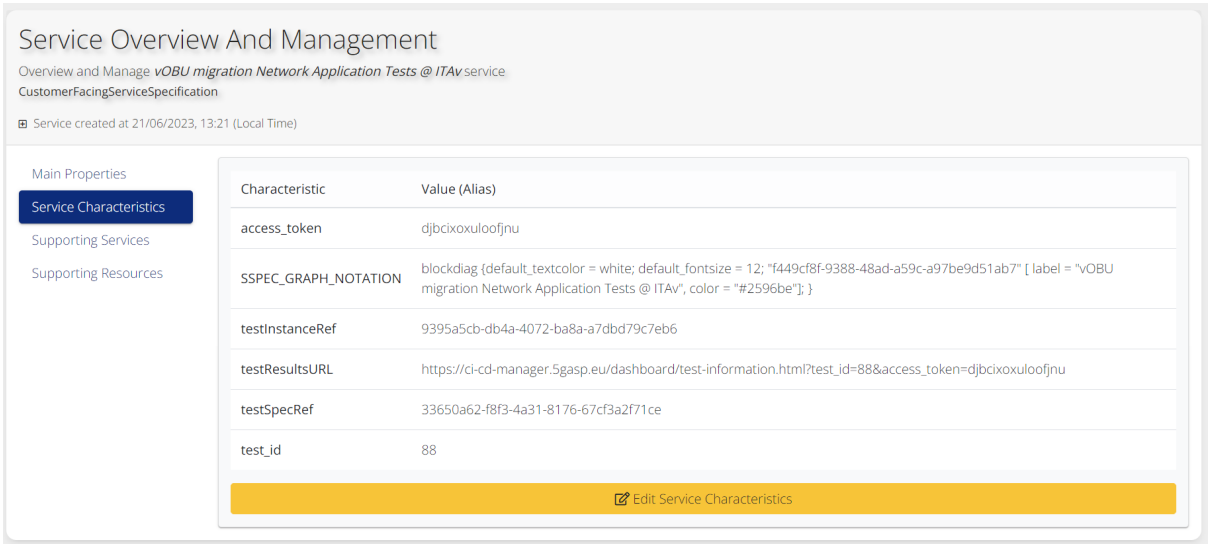

Figure 29: Test results URL exposure

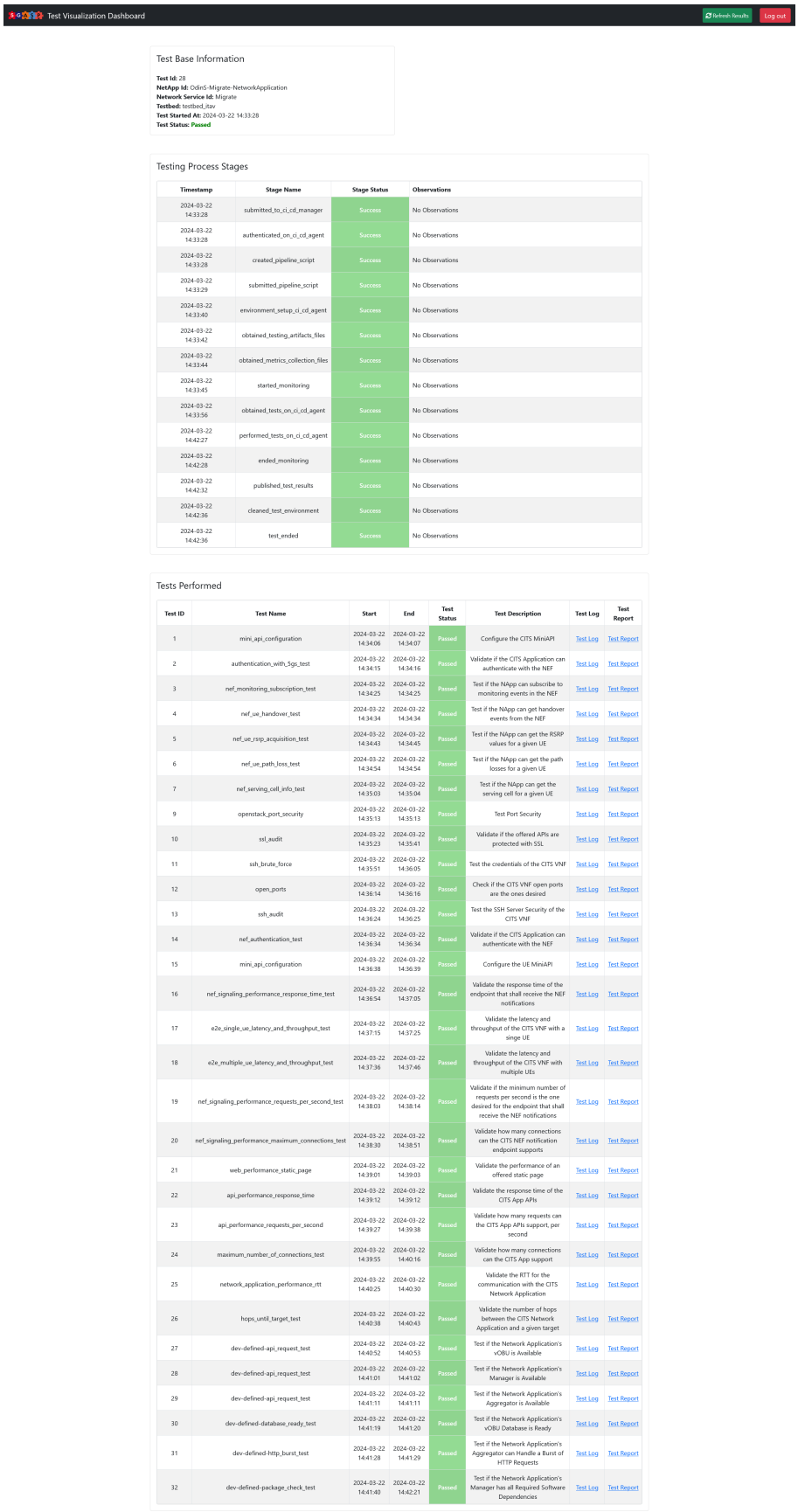

Figure 30: Test results report

The previously presented implementations were enhancements of the already existing design in regard to the NODS portal, based on stakeholders' reports and remarks.

As the project's use cases matured and an increasing number of Network Application developers onboarded their applications and ventured towards the 5GASP pipeline's lifecycle there emerged the need for the simplification of the onboarding process.

The existing design of discrete portals per required task, i.e. Network Application onboarding, test suite onboarding, triplet design, prior to execution and validation was not cognitive convenient for the wide majority of the involved stakeholders.

Thus, a new portal was designed, combining all the aforementioned processes into one actual single-entry point. Therefore, this portal followed cognitive design practices which enabled the seamless step-by-step design of the 5GASP deployment artefacts, namely the triplet. Specifically, the procedure initiates with the selection of a hosting network slice from the available pool (Figure 31), followed by the uploading of the corresponding VNF/NSD artefacts of the Network Application, which undergo a package validation based on the OSM version selected (Figure 32).

Once this step is marked as successful, the developer is prompted to define the desired test suite, either from a list of the already available public ones or by designing it from scratch (Figure 33). Finally, the automatically created triplet is displayed for confirmation and supposedly ordering and execution (Figure 34).

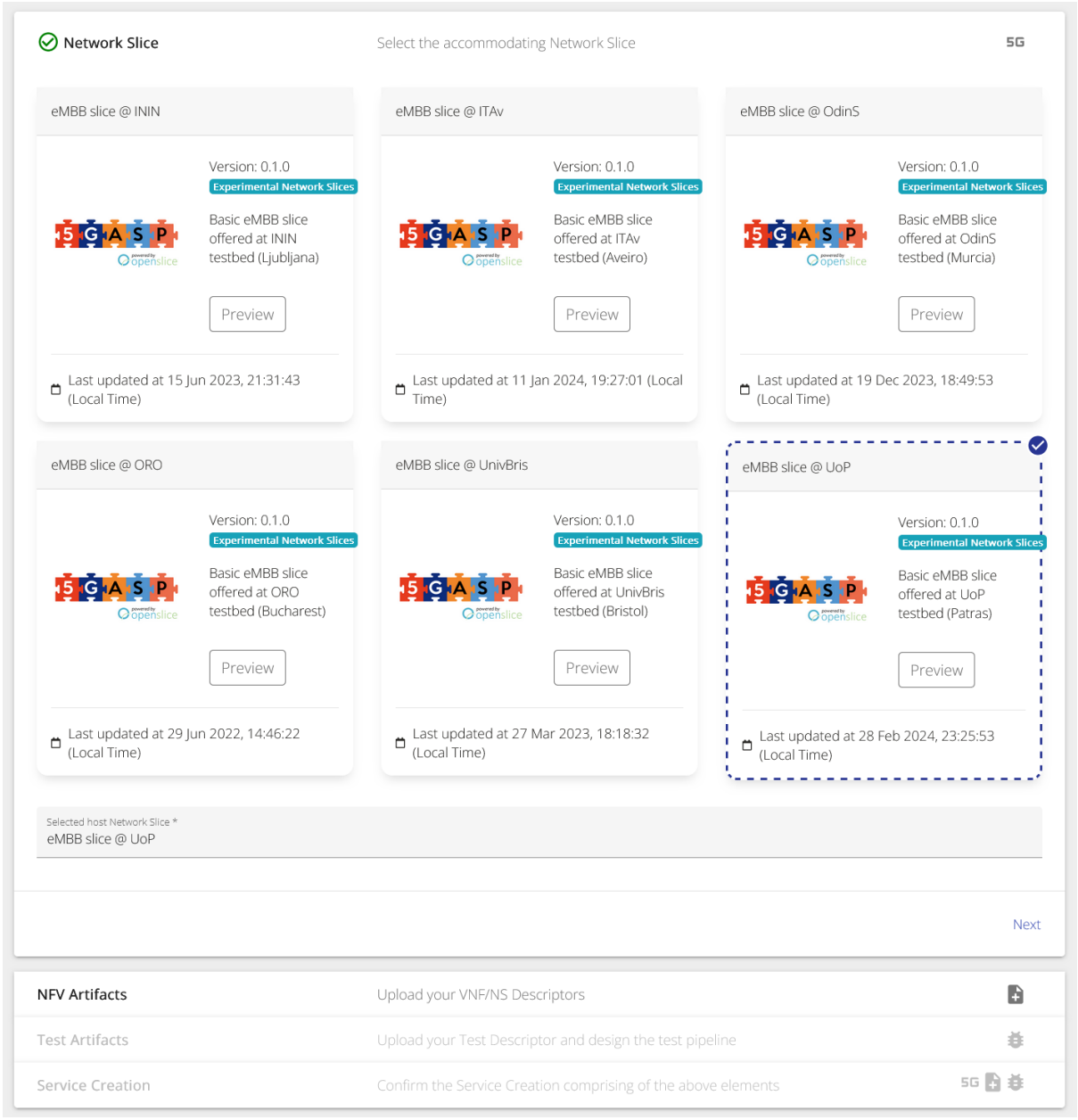

Figure 31: Triplet design portal – Network slice selection

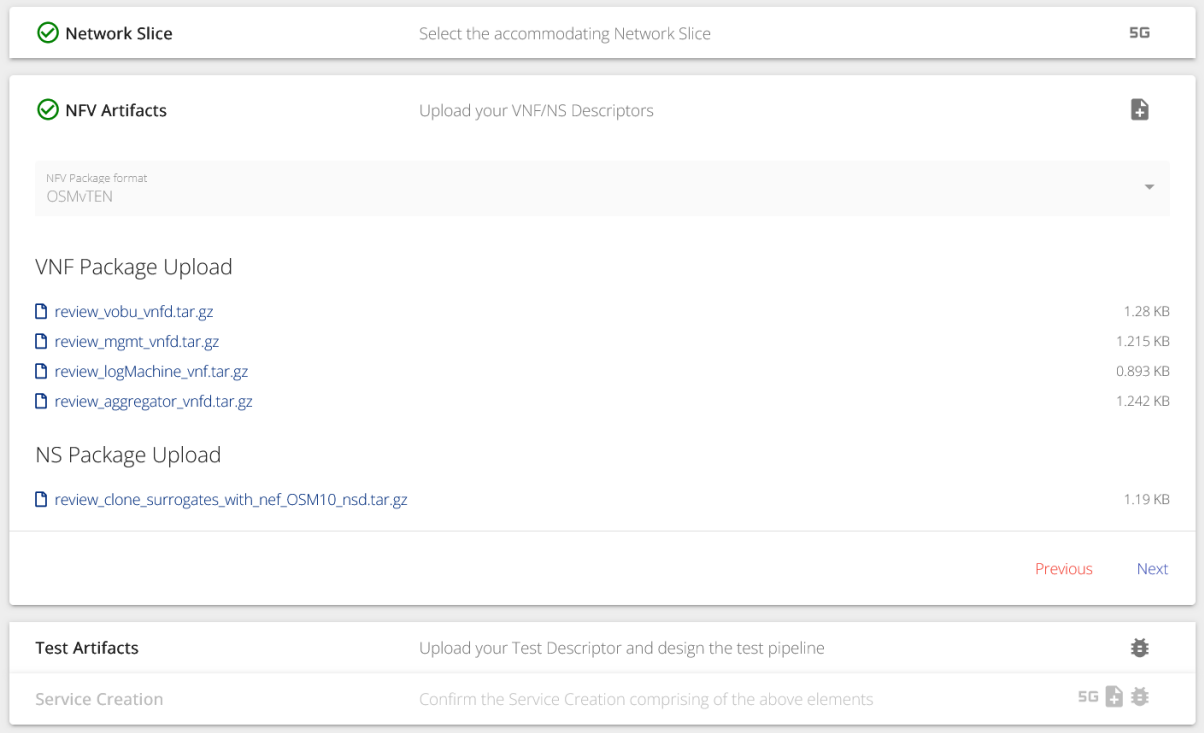

Figure 32: Triplet design portal – Network Application VNFs/NSDs onboarding

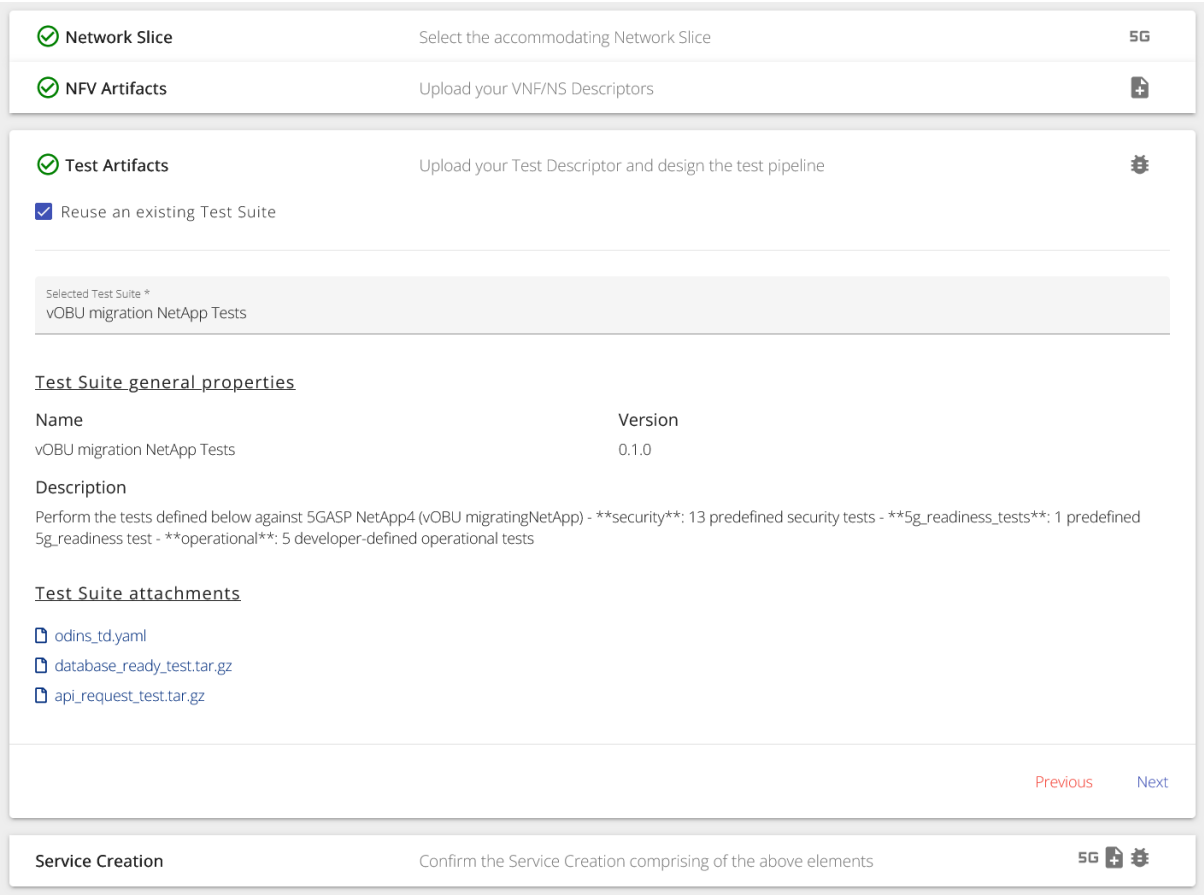

Figure 33: Triplet design portal – Test suite selection/design

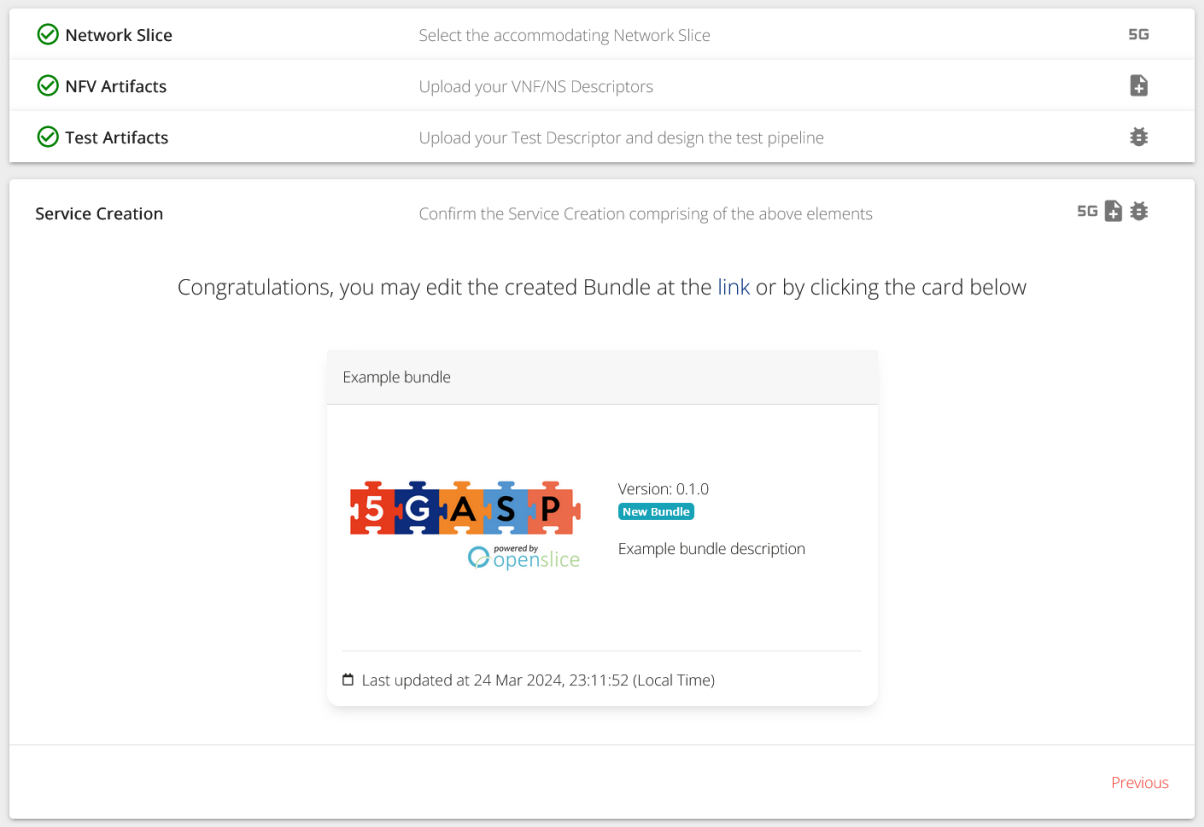

Figure 34: Triplet design portal – Created triplet confirmation

6 5GASP multi-domain NFV fabric

6.1 Design

Within the context of the 5GASP environment, NetOr [22, 23] is tasked with deploying and managing a mesh network connecting all testbeds. This entails establishing a network topology where all testbed Wireguard peers are directly interconnected, enabling efficient data routing between them. To achieve this, Peer-to-Peer (P2P) VPN connections are employed between all peers. Unlike alternative topologies such as Client-Server, P2P distribution of the load ensures that each peer shares the processing burden, preventing reliance on a single entity to handle all tasks. For the deployment of the Wireguard peers, NetOr relies on Interface E4, while the deployed peers will further rely on Interface E7. More context on the need of having inter-domain tunnels between 5GASP's testbeds is provided in D3.1 and D3.2.

For orchestrating the inter-domain tunnels, we rely on day-0, day-1, and day-2 operations. Day-0 operations are responsible for configuring the initial resources for the VPN Nodes, offered through VNFs. These operations define the allocation of CPUs, RAM, and disk sizes for the VPN nodes. Additionally, it is through day-0 operations that we select the IP addresses to be associated with the VPN node VNFs, ensuring their public exposure. Configuration of the Wireguard version to be installed in the VPN nodes is also performed during day-0. Subsequently, day-1 operations are employed to install and configure Wireguard on the VPN Nodes. These operations rely on instantiation parameters provided by NetOr prior to forwarding instantiation requests to the NFVOs in the interconnected domains. These parameters include instructions on accessing and interacting with the DNS Server provided by NetOr, along with a cipher key for encrypting sensitive data in DNS Server records. Day-1 operations enable the VPN nodes to publish their information in the DNS Server and gather details about other VPN nodes, which is crucial for establishing the VPN tunnels among them. Upon completion of day-1 operations, it is expected that all VPN tunnels are established. Finally, day-2 operations offer further configuration options for the VPN tunnels. Although manual invocation is required, these operations allow for more detailed management of the tunnels. Through day-2 operations, clients can obtain various information about the VPN nodes, including their IPs and resources. Additionally, it is also possible to update VPN node routing, list available VPN tunnels, adjust exposed networks, manage tunnel bandwidth, and remove nodes from the mesh VPN. Figure 35 describes the initial orchestration process of the VPN Tunnels, while Figure 36 showcases the workflow required for the VPN Nodes to create the tunnels between them.

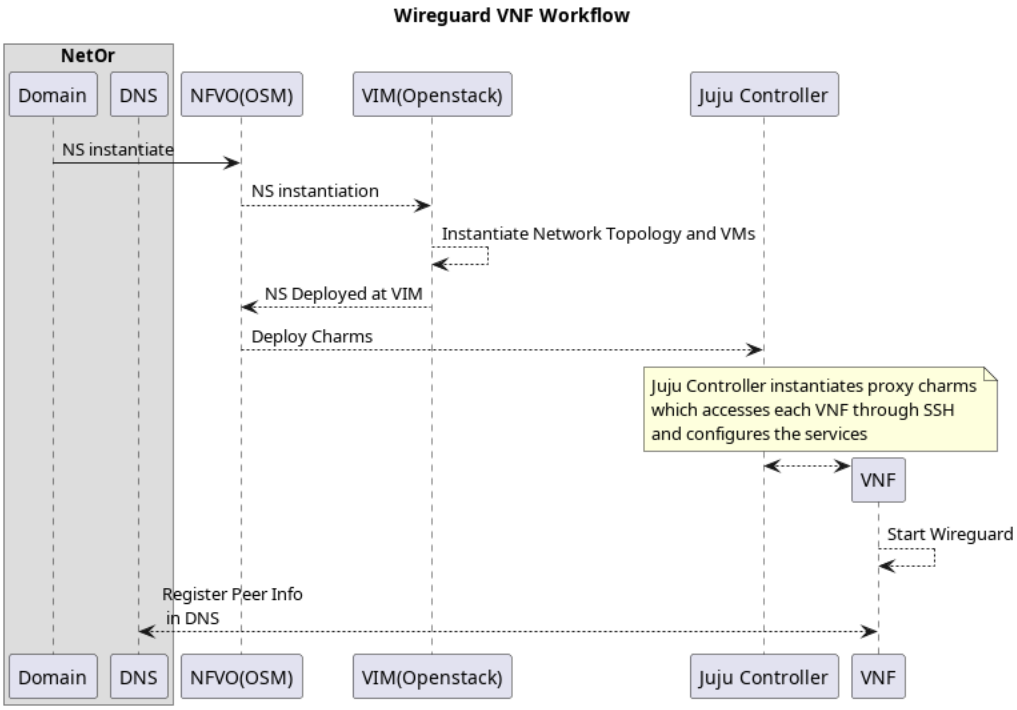

Figure 35: Initial Orchestration Process of the VPN Tunnels

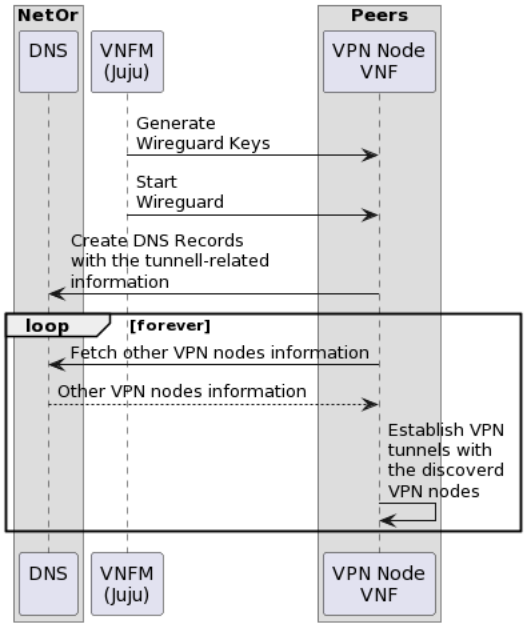

Figure 36: Tunnels Establishing Process Workflow

6.2 Underlying technologies and security considerations

NetOr's underlying technologies and security considerations have already been fully address in D3.2. Still, in this document we provide a comprehensive summary of these technologies and security considerations.

Regarding the Network Orchestrator and inter-domain connectivity, Wireguard was the tool chosen to provide the VPN tunnels between 5GASP's testbeds. Wireguard is a cutting-edge open-source tool that provides high speed connectivity and relies on advanced cryptography mechanisms also used by other VPN options like OpenVPN and IPsec/IKEv2. Its lightweight design minimizes its attack surface, making it easier for security researchers to audit for vulnerabilities. Furthermore, its ease of deployment and flexibility offers numerous advantages not found in other VPN tools.

However, it is important to highlight that the inter-testbed VPN tunnels pose security challenges, as a breach in one testbed can be exploited to compromise the others. To address this, implementing Traffic Filtering and Intrusion Detection and Prevention Systems on the Wireguard Peers can be a meaningful approach to reduce risks by limiting unnecessary traffic and stopping potential attackers.

7 Implemented capabilities from 5GASP facilities

7.1 Implementation to support the multi-domain NFV fabric

7.1.1 Implementation of E1 interface

As described in Section 4.2.1, all 5GASP facilities have elected Option B, i.e. direct exposure of OSM, as the way to communicate with NODS. This choice necessitates an uninterrupted communication channel to facilitate the synchronization scheduler's proper functioning, ensuring the continuous updating of the NFV artefact list between the portal and the underlying facilities. Furthermore, any triplet orchestration order initiated must be treated as an asynchronous task to minimize the requirement for manual intervention. These requirements underscore the importance of ensuring stable and uninterrupted VPN connectivity, a goal accomplished through the implementation of fully automated recovery scripts designed to eliminate downtime. The subsequent Table 5 encompasses the updated interface implementation for each facility.

Table 5: E1 Interface implementation per facility (final)

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: 12	VPN (SSL)
Murcia	Option B	NFVO (OSM) Version: 10	Firewall rule
Ljubljana	Option B	NFVO (OSM) Version: 10	VPN (SSL)

Additionally, recognizing the criticality of maintaining stable VPN connectivity, a web-based monitoring system has been developed to provide an overview of the health of VPN connections, as illustrated in Figure 37. This system offers real-time insights into the status and performance of VPN connections across all 5GASP components, allowing for proactive identification and resolution of any potential issues that may arise.

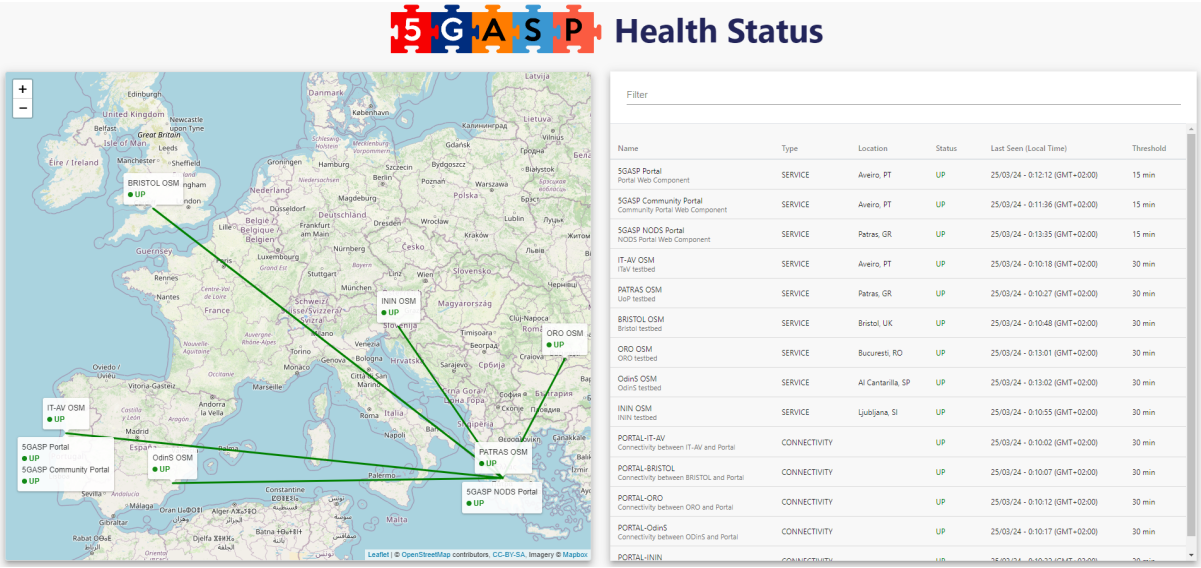

Figure 37: VPN health monitoring system

7.1.2 Implementation of facilities interconnection

As previously addresses in Section 4.2.3, a full mesh VPN has been achieved between all consortium testbeds. This mesh VPN was entirely achieved through a Zero-Touch [30] scenario, where the Wireguard peers are orchestrated and configure, later establishing the VPN tunnels between them, without the aid of a human being.

After all testbeds became part of the mesh VPN, several network performance tests have been performed. These showcased that it is possible to achieve good communication speeds between the different testbeds. Table 6 showcases the results of these tests for the connectivity between the testbeds of OdinS, ITAv, and University of Patras.

Table 6: VPN Tunnels Network Performance between ITAV, OdinS, and UoP

From/To	ITAv	OdinS	University of Patras
Throughput (Mbps)			
ITAv	-	99.49 ± 3.63	44.22 ± 6.53
OdinS	67.61 ± 13.32	-	107.10 ± 15.25
University of Patras	52.4 ± 11.22	104.02 ± 29.39	-
Jitter (ms)			
ITAv	-	0.54 ± 0.30	0.58 ± 0.9
OdinS	0.46 ± 0.45	-	0.18 ± 0.13
University of Patras	0.35 ± 0.19	0.09 ± 0.07	-
RTT (ms)			

ITAv	-	25.10 ± 0.45	72.44 ± 0.96
OdinS	25.10 ± 0.45	-	77.91 ± 0.84
University of Patras	72.44 ± 0.96	77.91 ± 0.84	-

7.2 Implementation to enable testing

The 5GASP testing benefited from the ICT-41 commonalities study in several ways:

1. **Standardization**: The ICT-41 commonalities study likely helped identify commonalities and standardization opportunities across different ICT domains, including automotive and telecommunications. By establishing commonalities in technologies, protocols, and standards, it facilitated the integration of automotive and telecommunications systems, which is crucial for 5GASP testing.
2. **Interoperability**: One of the key aspects of the commonalities study would have been to address interoperability challenges. By identifying commonalities in protocols and technologies, the study likely laid the groundwork for seamless interoperability between automotive systems and 5G networks. This is essential for testing 5GASP applications effectively.
3. **Reduced Redundancy**: By understanding the commonalities between ICT domains, redundant efforts in developing testing frameworks and methodologies could be minimized. This streamlining of efforts would have allowed for more efficient utilization of resources in the development and deployment of 5GASP testing infrastructure.
4. **Faster Deployment & Testing:** With a clear understanding of commonalities, the implementation of 5GASP testing could have been accelerated. Leveraging existing ICT standards and frameworks from the commonalities study would have reduced the time required for designing and deploying testing infrastructure specific to 5GASP applications.

In essence, the ICT-41 commonalities study likely provided a foundation of knowledge and standards that facilitated the integration of telecommunications technologies, ultimately benefiting the testing of 5GASP applications by streamlining development efforts, ensuring interoperability, and reducing costs.

7.2.1 CI/CD Service

5GASP's CI/CD Service relies on 4 main entities: (i) the CI/CD Manager, (ii) the Test Results Visualization Dashboard (TRVD), (iii) the CI/CD Agents, and the (iv) Local Test Repositories (LTRs). In order to enable testing and validation processes among the different testbeds, it is required that:

- An LTR is available on each testbed
- A CI/CD Agent is provided by each testbed
- The CI/CD Manager can communicate with the CI/CD Agents, to send them testing and validation tasks
- The TRVD can communicate with the CI/CD Manager

In respect to the CI/CD Manager and TRVD, both entities have been made available by ITAV's testbed. These entities have been publicly exposed and can be reached through the following URLs:

- https://ci-cd-service.5gasp.eu/manager
- https://ci-cd-service.5gasp.eu/dashboard

Additionally, the status of fulfilment of the previous requirements is showcased in Table 7.

Table 7: Status of fulfilment of the CI/CD Service's requirements

Testbed	CI/CD Agent available?	Has the connectivity between the CI/CD Agent and the CI/CD Manager been achieved?	LTR available?
Aveiro	Yes	Yes	Yes
Bucharest	Yes	Yes	Yes
Bristol	Yes	Yes	Yes
Ljubljana	Yes	Yes	Yes
Murcia	Yes	Yes	Yes
Patras	Yes	Yes	Yes

7.2.2 NEF Emulator

Throughout the project's duration, it became evident that 5GASP would need to support programmable interaction towards a 5G System component of the underlying infrastructure, in order to facilitate readiness tests for external Network Applications in the context of 5G. Specifically, the Network Exposure Function (NEF) stands out as a pivotal component within the 5G System, enabling the exposure of network services and functions to external entities, including third-party applications and services.
However, existing implementations of 5G Cores within the project lack a fully operational NEF, and those that do exist exhibit significant inconsistencies in their implementations. Consequently, the project sought a standards-based, reprogrammable, and deterministic solution, leading to the adoption of the NEF Emulator [28], an open-source project initially developed within the context of the EVOLVED-5G project under the ICT-41 umbrella. Further enhancements were made to the NEF Emulator by 5GASP to meet the project's requirements, such as the development of the reporting API. The aforementioned API records the NEF Emulator's north-bound API calls enabling the confirmation of Network Application interaction with it.
The emulator provides standardized 3GPP NEF APIs within a configurable emulated environment, allowing for the definition of specific simulation configurations (e.g., number and type of User Equipment, positioning of gNBs, etc.), thereby establishing the necessary groundwork for testing Network Applications against the interaction with emulated NEF APIs, as presented in the next sections.

7.2.3 Linux Traffic Control

The Linux Traffic Control (TC) service is a tool that allows for the manipulation and shaping of network traffic within a Linux system. It offers functionalities such as bandwidth limiting, delay emulation, packet loss simulation, and prioritization of network packets. In the context of emulating network restrictions between application components, the TC service can be employed to simulate various network conditions, such as network congestion, latency fluctuations, packet loss, and bandwidth limitations. By configuring traffic control rules using TC, an experimenter can accurately replicate real-world network scenarios and assess the behaviour of the application under different conditions.
Moreover, the repeatability and programmability of the TC service enable the realization of the above test scenarios, facilitating consistent testing and deterministic results across the diverse 5GASP deployment environments. By incorporating the TC service into its automated testing workflows, 5GASP intended to evaluate a Network Application reliability, through specific availability and continuity assessment test cases. For instance, a test could evaluate how well an application performs under high traffic loads or when network latency is elevated.

7.2.4 Summary

This section aims to summarize the different testing setups with the context of 5GASP, incorporating the latest introduced testing tools of the previous sections, namely the NEF Emulator and Linux Traffic Control.
The first and simplest scenario incorporates only the core testing entity of each testbed, namely the CI/CD Agent, and is illustrated in Figure 38. The latter utilizes a robot script to trigger a test execution against a deployed Network Application. The test results are accumulated back to the CI/CD Agent and are exported as an HTLM test report (see Section 5). During this test setup, the Network Application is considered as a passive element.

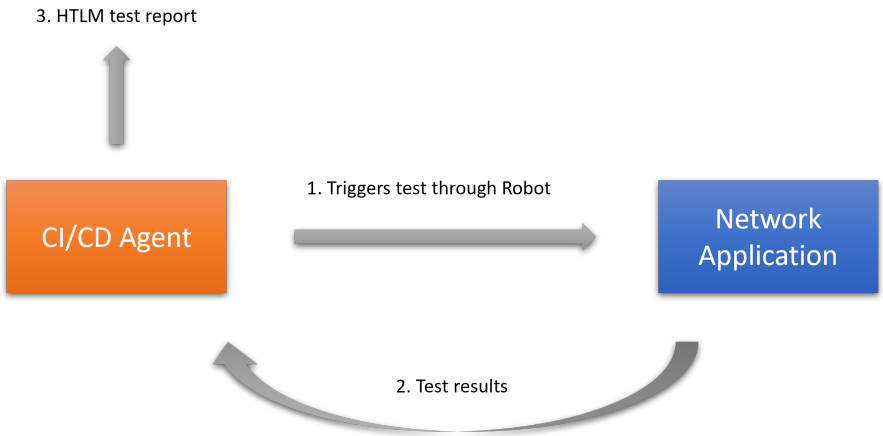

Figure 38: Test setup A

The second scenario introduces the use of the NEF emulator in the testing pipeline and is utilized when a Network Application is tested for its 5G readiness, as seen in Figure 39. The flow initiates with a corresponding robot script which now instructs the Network Application to interact with the NEF emulator through its NBI. The interactions are predefined and are based on the NEF's API. At this current setup, the Network Application actively participates in the testing procedure as it has to incorporate the code block to perform the requested NEF interaction request. Should the request towards the NEF Emulator is successful, the Network

Application receives the emulated control plane events while simultaneously the CI/CD Agent validates that the request was executed by querying the reporting API of the NEF Emulator (see Section 7.2.2).

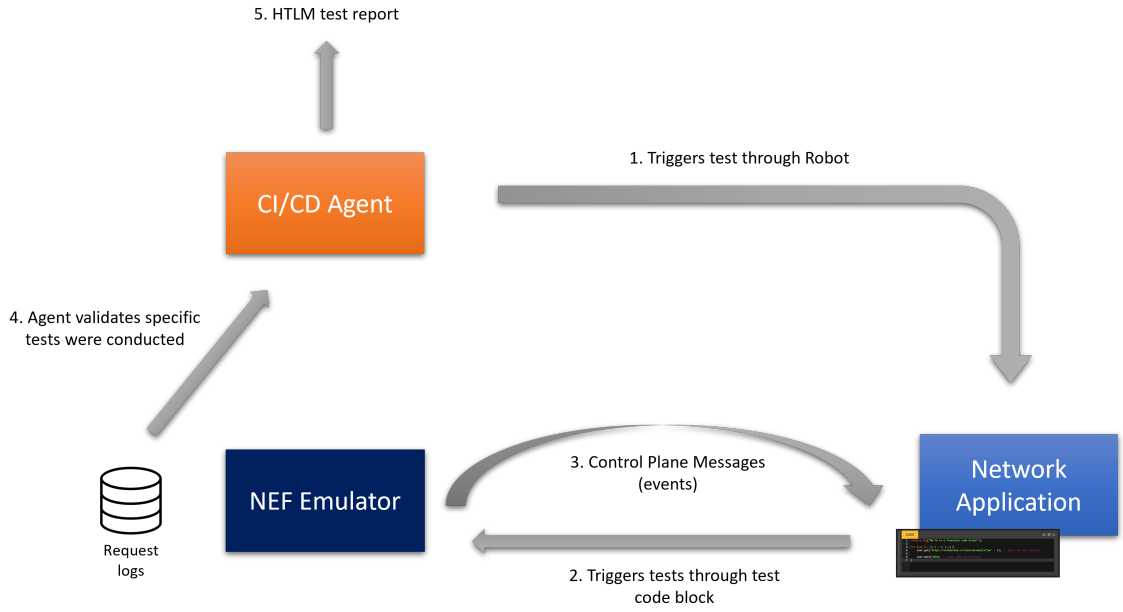

Figure 39: Test setup B

Finally, the third scenario encompasses all the aforementioned testing tools, i.e. the NEF emulator and the Linux Traffic Control, as depicted in Figure 40. Typically, the process starts with the robot script instructing the Network Application to interact with the NEF Emulator. Consequently, the Network Application receives the respective emulated control plane events, but simultaneously these events are propagated towards the Linux Traffic Controller as well, which enforces specific network restrictions based on the test scenario. Now, like the previous setup, the CI/CD Agent validates that the request was executed but also confirms the resilience of the Network Application towards the imposed network restrictions via a provided healthcheck probe.

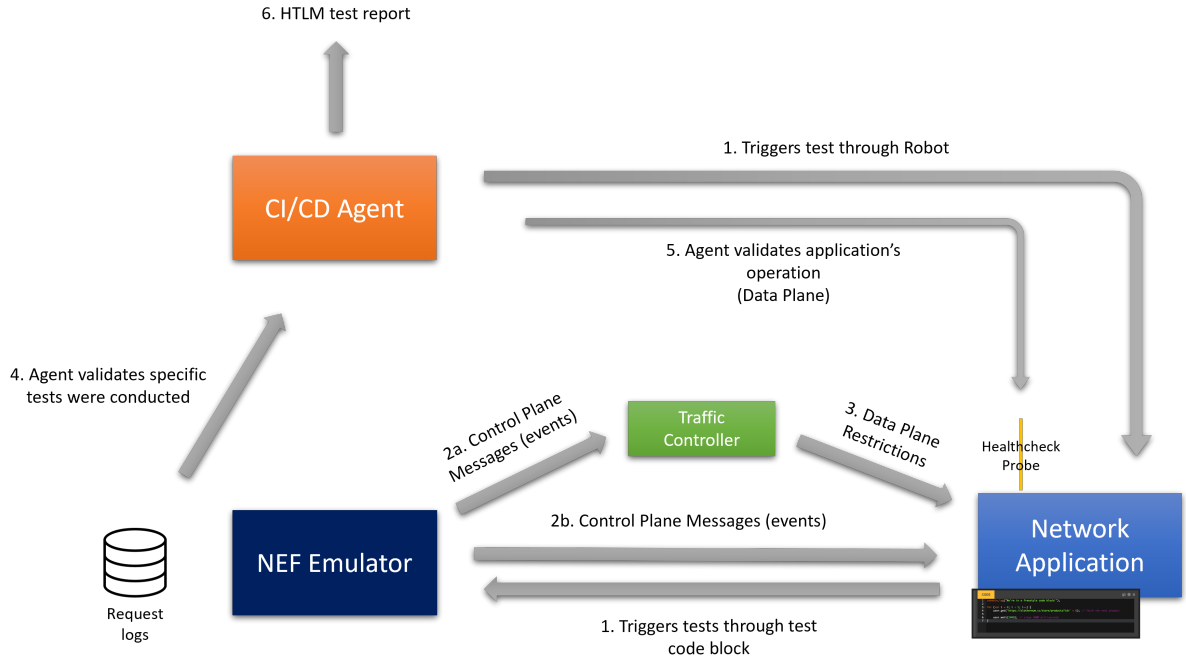

Figure 40: Test setup C

7.3 Implementation to support network slices

7.3.1 Patras (UoP) facility

The final implementation from the Patras' facility to support network slices is based on the availability of four 5G radio nodes (from different vendors, i.e. Amarisoft and AW2S) and the Open5GS solution as the provided 5G Core. The overall stack is fully orchestratable and capable of supporting both indoor and outdoor experimenting, while also enabling handover test scenarios. Moreover, during the latest phase of the project, the facility expanded its compute resources towards additional geo-distributed sites, serving as edge nodes. All the edge nodes encompass their dedicated Kubernetes clusters and are expected to host any requested workloads or even facilitate fully-orchestrated multiple-UPF deployments. Furthermore, as already addressed in D3.1, the facility also maintains an OpenStack cluster to provide the anticipated backwards compatibility towards legacy VNF-based deployments.
Finally, apart from the solemn selection of available radio nodes to support a requested area coverage, the Patras facility developed and maintained its own dedicated and cloud-native solution to extend the potential radio configuration offering advanced network slices. This solution, i.e. the Radio Management Server, provides programmable (de)registration to a 5G Core, profile selection and transmission power definition, severely augmenting the versatility of the potential experimentation.

7.3.2 Aveiro (ITAv) facility

ITAv's eMBB slice is delivered through a Huawei 5G Core and a Huawei 5G RAN. Since ITAv's Huawei 5G Core does not provide APIs for its management, one must rely on the Web UI it

provides. Another approach is to rely on SSH commands executed on each component of the Core.

Given this restriction, ITAv developed a RESTful API to perform 5G-related configurations through SSH commands. The capabilities of this API are presented in D3.2.

Previously, as addressed in D3.2, the orchestration and further configuration of ITAv's Network Slices were managed by the NODS, by invoking the previously introduced API. However, in Q3 of 2022, ITAv packaged that API in a VNF, that can be seamlessly orchestrated by OSM. Therefore, ITAv is now aligned with the remaining consortium members, who provide VNFs that can be orchestrated through a SOL005 standardized API, such as the NBI provided by OSM. This means that the NODS can create a bundled service that orchestrates the Slicing VNF and the Network Application itself, simplifying the workflow of deploying a Network Slice before the deployment of a Network Application.

7.3.3 Murcia (OdinS) facility

As introduced in D3.2, Murcia testbed relies on Free5GC [31] as 5G core and it uses the 5G gNB stack from Amarisoft to complement it. The orchestration is made using OpenStack, thus the free5GC core is deployed in it in a regular VM-deployment. This dynamically deployed core is complete, containing all the necessary elements. The Amarisoft 5G gNB is then configured to connect to the deployed core. The 5G network operates in the n78 band using 40 MHz of bandwidth.

The slicing is achieved in the RAN by managing the Amarisoft 5G gNB via its built-in websocket to read and modify the configuration. With the current setting, it is possible to create eMBB slices to which a client can connect and leverage a higher level of Quality of Service than the one delivered to regular clients.

7.3.4 Bristol (UoB) Facility

Bristol facility was enhanced with one additional (mobile) 5G cell, aiming at expanding the 5G SA coverage in band n77 within the city of Bristol. The network performance is improved by utilizing the increased spectrum availability in band n77 (100 MHz). Potentially more radio and core functionality, higher network performance and higher stability would also be expected after upgrading the software of the RAN. Specifically, the software upgrades should provide throughput gains and better latency management functionality, and additional 5G slicing functionality.

7.3.5 Ljubljana (ININ) Facility

The ININ/Ljubljana facility moved from VNF/VM-based deployment of containerized 5G SA system components (as described in D3.2) to fully supported CNF/Kubernetes 5G SA slice orchestration using in-house developed Helm chart. Such type of deployment greatly reduces the deployment time of the whole 5G system, provides additional probing and restoration mechanisms built in Kubernetes and simplifies upgrading/restoring components.

Orchestrating day-1 and day-2 operations are fully supported by using Kubernetes ConfigMap resources which enables that the 5G slice characteristics or the slice itself to be changed directly from the orchestrator (OSM) after the initial deployment by invoking the appropriate day-2 configuration command.

The 5G SA deployment is provided indoor using SDR-based RAN (40 MHz) and outdoor using RRH-based RAN (50MHz), both deployed in 5G n78 band.

7.3.6 Bucharest (ORO) Facility

ORO's eMBB slice delivery capabilities have been achieved in March 2022, with incremental upgrades since, as reported in D3.2. The capabilities are based on an 5G Open Air Interface solution, comprising of gNB RAN and 5GCN Core, deployed on-premises in ORO's 5G Lab as OpenRAN and Open Core running in an Kubernetes Environment. Service orchestration is achieved through a CICD pipeline (Jenkins). Furthermore, ORO's Facility integrates OSMv10 and can provide full 5GSA capabilities, limited to the geographical coverage of the localized site by using N78 5G Spectrum.

7.4 Implementation to support monitoring

Traceability, logging and monitoring of each Network Application stream and collation with the streams from the underlying 5G infrastructure, finally routed to a central destination for viewing and long-term archival / analysis is an important multi-domain feature of 5GASP.

Leaning on the ICT-41 commonalities study from D2.3, and in review of the Monitoring Platform of VITAL-5G and the Monitoring & Analytics toolset of 5G-IANA, the Federated Prometheus platform of 5GASP has become a meaningful way for Network Application developers to profile deployments along with assessing an application's behaviour over time.

Embracing a unified approach, the 5GASP monitoring platform offers invaluable insights into infrastructure performance across the entirety of the 5GASP lifecycle, allowing developers to observe application behaviour within a dynamic, real-time system. To achieve this goal, we have proposed the adoption of the widely used Prometheus [24] / Grafana [25] stack as the common monitoring platform. Prometheus, being an open-source software that excels at collecting and storing time-series data that enables efficient querying and alerting of various metrics, was already employed within some 5GASP testbeds to gather metrics from various infrastructure nodes, so it was elected as the perfect candidate for data aggregation. Conversely, Grafana offers an intuitive interface for creating interactive dashboards and visualizations based on data collected by Prometheus. Its seamless integration with Prometheus, coupled with the ability for individual testbeds to generate custom dashboards showcasing heterogeneous data under a unified platform, makes Grafana the ideal visualization tool.

Through the implementation of the Prometheus stack, we establish a centralized monitoring solution that seamlessly integrates with various components of the 5GASP infrastructure, as depicted in Figure 41. This unified approach ensures consistent monitoring practices

throughout the infrastructure, facilitating efficient troubleshooting and performance optimization of Network Applications across the 5GASP CI/CD lifecycle. Additionally, the common monitoring platform provides stakeholders with access to metrics via a centralized portal, known as 5GASP Central Grafana. This portal serves as a hub for visualizing and analysing collected metrics, enabling real-time monitoring throughout the experiment lifecycle. Through this centralized portal, developers gain valuable insights into infrastructure behaviour and performance, enhancing resource utilization efficiency.

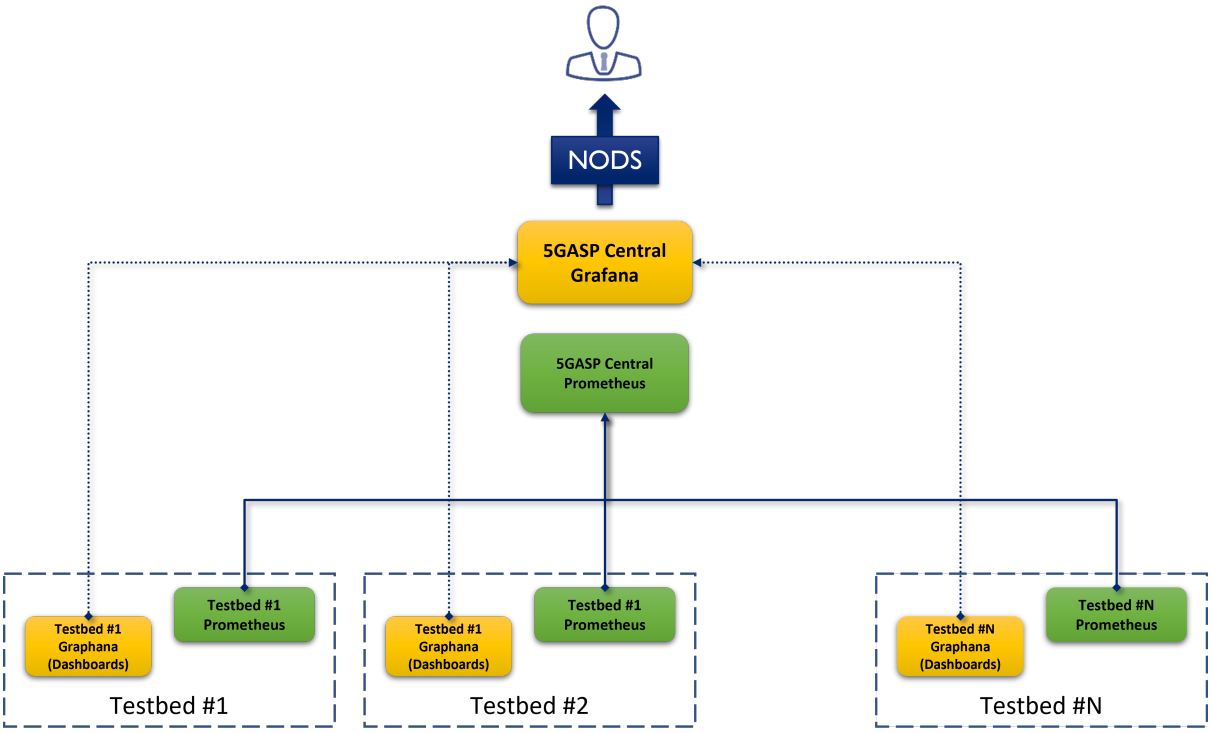

Figure 41: 5GASP federated Prometheus platform

7.4.1 Patras (UoP) facility

The Patras facility offers a dedicated Prometheus-based monitoring stack exposing the testbed resources utilized by 5GASP, e.g. 5G System, Kubernetes clusters and CPEs, towards the Central Prometheus. Specifically, for the 5G System the facility mainly provides RAN-oriented metrics at the measurement points illustrated in Figure 42.

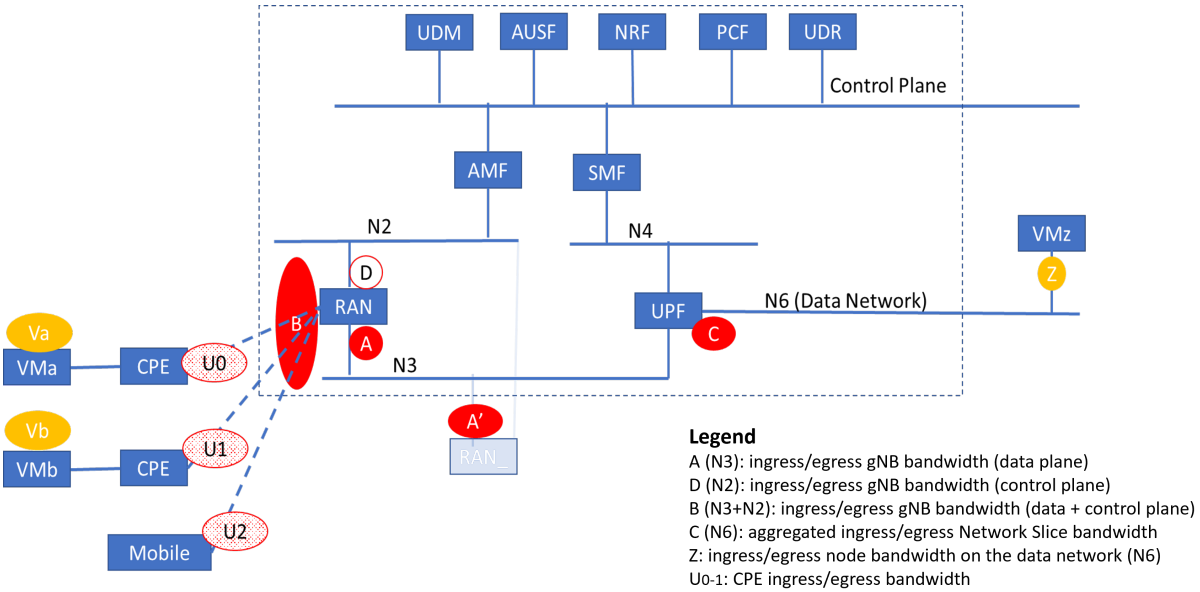

Figure 42: Patras facility 5G System monitoring measurement points

The Grafana dashboards that the facility provides for the monitoring of the 5G System and the employed Kubernetes clusters are depicted in Figure 43 and Figure 44 respectively.

Figure 43: Patras facility 5G System dashboard

Figure 44: Patras facility Kubernetes dashboard

7.4.2 Aveiro (ITAv) facility

The implementation of monitoring mechanism at ITAv's testbed was challenging due to their Huawei 5G Core not offering APIs to programmatically collect metrics. Therefore, an independent and proprietary monitoring system was developed by ITAv. This system relies on a SDN Switch, running OpenVSwitch, that acts as a man in middle for the communication between the 5G Core and the Infrastructure that hosts the Network Applications. Even though this system cannot collect meaningful data from the 5G RAN, it still provides insights regarding the traffic generated/consumed by the Network Applications.

To implement the above-mentioned monitoring system, ITAv encapsulated the traffic between the 5G Core and the Network Function Virtualization Infrastructure in VLANs. To this end, the traffic of each Network Slice is encapsulated inside an independent VLAN. These VLANs are then propagated through the SDN Switch, from where ITAv collects network related metrics.

To obtain and process the network metrics from the SDN switch, a dockerized custom service was developed. This service is responsible for periodically requesting metrics from the SDN Controller (Ryu) that manages the SDN Switch. Furthermore, this service also exposes the obtained metrics through an HTTP Server, offering and endpoint that can be scrapped by Prometheus. It is precisely through Prometheus that ITAv collects and stores monitoring information. Moreover, ITAv's Prometheus can be used as a data source for a Grafana Dashboard, thus enabling the visualization of the collected metrics.

To illustrate the architecture of the presented monitoring system, Figure 45 is provided. Furthermore, Figure 46 showcases the Grafana Dashboard.

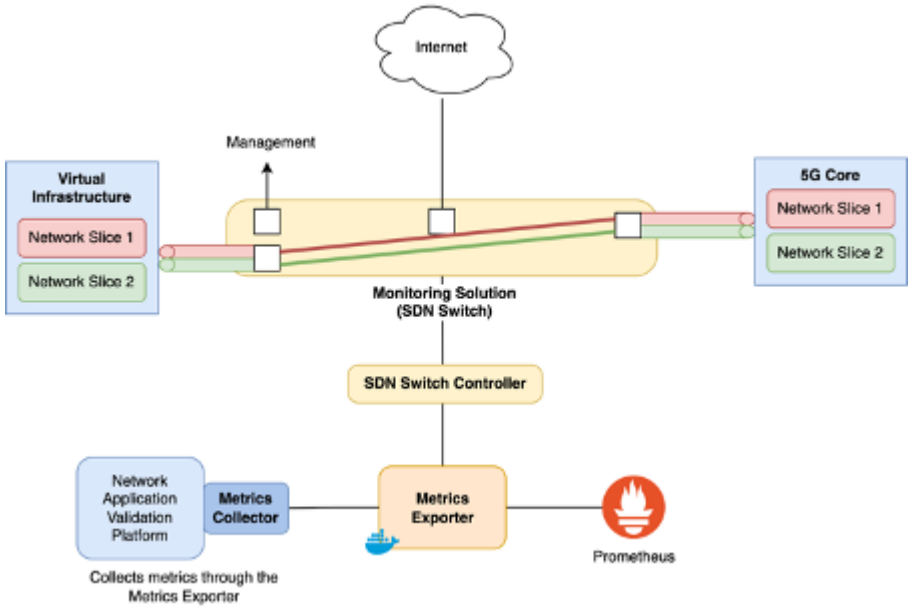

Figure 45: ITAv's Monitoring System - Architecture

Figure 46: ITAv's Monitoring System - Dashboard

7.4.3 Murcia (OdinS) facility

The Murcia facility provides a complete monitoring platform based on Prometheus, Grafana, collectd and custom scripts. The solution for the project is tailored to 5GASP's needs and it is fed by the central platform, where all the data is available. In this way, a federated Prometheus is deployed for the project use with all the monitoring data related to the 5GASP resources.

As it can be seen in Figure 47, the implementation and the architecture has not been modified since its presentation in D2.3. However, significant enhancements have been made by developing new custom scripts and improving the existing ones to monitor the 5G network. Besides, new Grafana panels has been configured to improve the readability and enable the inclusion of new metrics. In Figure 48, the main Grafana panel for the 5G network can be seen, where information about the backbone network, radio network, computing resources and also metrics related to the UEs can be explored.

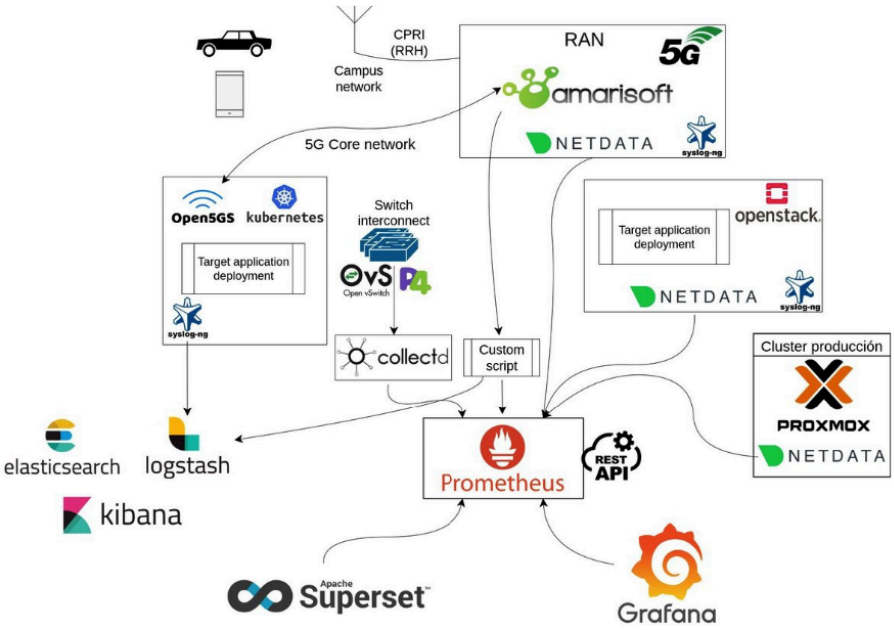

Figure 47: Murcia monitoring platform architecture

Figure 48: Murcia site Grafana main panel

7.4.4 Bristol (UoB) Facility

The Bristol facility has the monitoring deployed separately, due to some provider-specific restrictions, and the Core being owned by other projects (also within the lab), however, both the 5G NR core monitoring dashboard and the OpenStack and OSM-integrated K8s cluster's resource monitoring are available, via the centralised Grafana interface (Figure 49 and Figure 50).

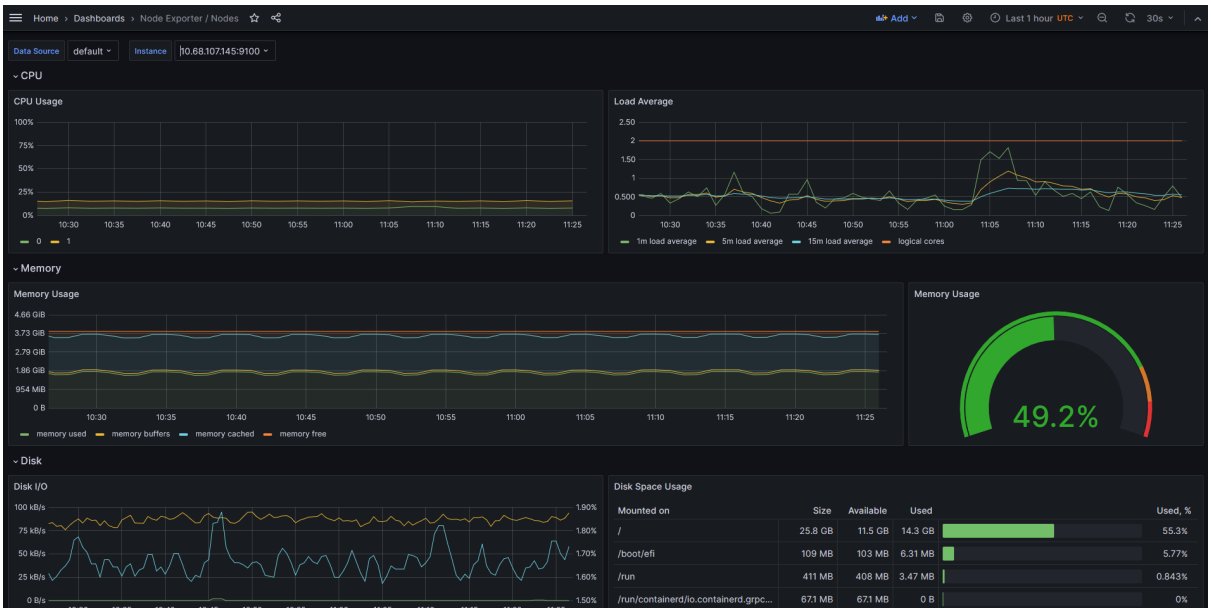

Figure 49: Bristol's testbed K8s Node Monitoring via NodeExporters

Figure 50: Bristol's testbed K8s Open5GS Core Monitoring via NodeExporters

The current monitoring architecture is presented in Figure 51, with no other updates from D3.2.

With the deployment of some more O-RAN [32] and Free5GC solutions, both the implementation and monitoring integration of the 5G core and O-RAN stack could be enhanced.

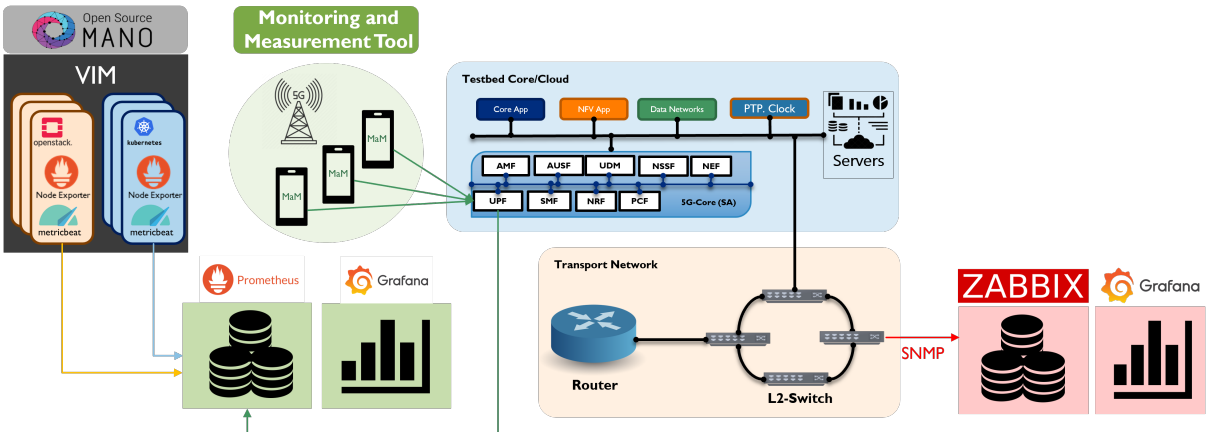

Figure 51: Overall monitoring architecture of 5GUK testbed

7.4.5 Ljubljana (ININ) Facility

The facility provides a dedicated Prometheus-based monitoring stack for 5GASP testbed resources (Figure 52) which allows for a simple integration in the central 5GASP monitoring portal. Collecting monitoring KPIs is realized on multiple layers:

- Cloud-based infrastructure, e.g. physical nodes hosting OpenStack and Kubernetes;
- 5G System monitoring that includes Core Network KPIs (e.g., number of connected/registered devices, NAS signalling, GTP bitrates), gNB KPIs (e.g., cell bitrates) and per UE KPIs (e.g., UE bitrates, UE retransmissions)
- UE end-to-end monitoring using qMON Network Performance Tool including ININ's 5G Gateway with Sierra Wireless EM9191 modem acting as an end-user.
- Environmental sensing providing temperature, humidity and CO/CO2 readings.

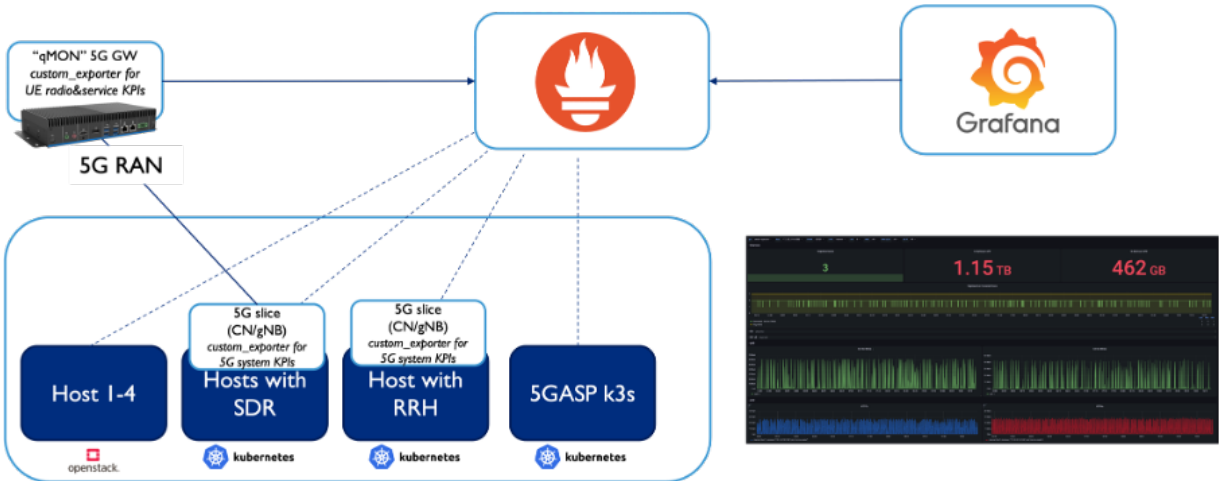

Figure 52: ININ's testbed monitoring architecture

7.4.6 Bucharest (ORO) Facility

A full-stack SBA Monitoring Platform, based on Prometheus and Grafana has been enabled in the 5G Test bed, alongside the existing, custom-build Click-based monitoring platform. This monitoring platform has been integrated in the Centralized Monitoring Solution, available across the 5GASP Test beds.

The Monitoring Platform collects inputs from agent and agent-less deployments, from the following sources:

- VIMs - Kubernetes, Open Stack
- Prometheus Agents deployed on IaaS / CaaS Compute x86 Servers
- ORO's 5G Lab internal framework for KPIs and metrics aggregation
- 5G Network Elements: RAN, Core, Transport, as 3GPP KPIs

The KPIs collected are network loads, slice(s) loads, packet loss, service network E2E delays, WMs state, CPU, RAM loads, CNFs loads.

All metrics are stored in a Time-Series database, based on Elastic and can be exposed through standardized RESTful APIs.

7.5 Implementation to support vertical needs

As highlighted early in this deliverable a ICT-41 commonalities study was carried out via WP2 and this had an impact on the revised implementation of 5GASP capabilities for the vertical facilities as it helped inform on how the automotive industry focused projects of VITAL-5G, 5G-IANA and PPDR focused 5G-EPICENTRE integrated the ability to manage and orchestrate Services/Network Applications across an extended compute continuum and comprised of multiple interconnected and virtualized segments. Most notable 5G-IANA had integrated virtualized infrastructure for Cooperative Intelligent Transport Systems (C-ITS) equipment, namely the On-Board Units (OBU) on the vehicles and Road-Side Units (RSU) and this guided 5GASP on how to implement enhancements in this final phase of the project.

7.5.1 Automotive

The 5GASP testbeds have been enhanced during the development of the project to accommodate in an easier manner the automotive vertical 5G network applications. In first place, On-Board Units (OBUs) have been configured to host the vehicular applications in a straightforward way so the developers can have a generic platform to execute their applications. These OBUs have 5G connectivity via PCIe or USB modems and are easy to deploy on any kind of vehicle, as they only need power. Also, in the case that specific equipment is needed to run the applications, these OBUs can provide 5G connectivity to other devices via Ethernet connections acting as a CPE.

Besides, the radio configuration in the gNBs of the testbeds hosting vehicular applications has been enhanced to better support these kinds of scenarios. In consequence, the coverage has

been extended and the behavior of the handovers among different cells has been adjusted to avoid connectivity losses. With these improvements, the overall performance of the network in terms of bandwidth and latency is now better.

Finally, the most complex issue to handle in these scenarios is the possibility to schedule real vehicle rides for the OBUs executing the 5G network applications through the testbeds. To do so, the developers and experimenters can contact the facility owners to schedule these rides and coordinate at the same time the deployment and evaluation of the network applications.

7.5.2 PPDR

5GASP testbed ecosystem provides some advanced PPDR features such as Dynamic slice change and Isolated operations to be tested with the network applications.

Dynamic slice change capability support in the testbed provides the option to network application to request the 5G slice to be changed dynamically by invoking the NPCF Policy Authorization API endpoint on the 5G system. The API can be seen in Figure 53. Some mandatory request parameters should include:

- afAppId – Application Function Identifier,
- dnn – Data Network Name,
- SUPI – Subscription Permanent Identifier,
- sliceInfo – requested SST/SD.

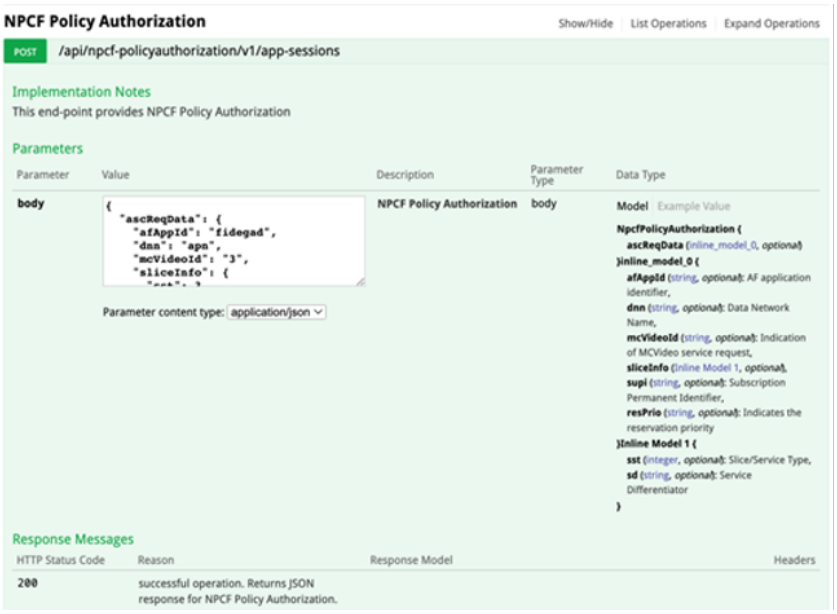

Figure 53: Example of a NPCF Policy Authorization API

Example of a such slice change operation is presented with the following steps:

- UE (identified by SUPI) is connected to the 5G network with the assigned slice (SST/SD 1/0),

- Request is triggered from the network application when the UE needs a different slice capability,
- Request includes mandatory parameters (i.e. SUPI and SST/SD 3/10),
- The UE receives de-register request with the flag signaling it has to reconnect immediately,
- The 5G system is reconfigured to assign a different slice to specified SUPI,
- The UE is reconnected and assigned the changed slice (SST/SD 3/10).

Another option is to support a disaster scenario in terms of PPDR vertical where the Isolated operation feature in the testbed is required. This allows the network application developers to operate and test the behaviour of their network applications during the switchover between public and local-only services, operation under isolated operation mode (i.e., only local services are available) and after the network is restored.

8 Conclusion

This deliverable being the last release of the status update, considering the technical goals and capabilities presented in D3.1 and D3.2, it successfully completed the objective to detail the tasks T3.1, T3.2, T3.3, and T3.4, each addressing a critical component of the 5GASP architecture.

ICT-41 commonalities studies were undertaken and a number of commonalities across ICT-41 testbed architectures have been identified, which raised opportunities across different ICT domains. This had an impact on the revised implementation of 5GASP capabilities.

This document also highlights the collaborative effort between Work Packages, presenting the finalized version of a common sheet that meticulously maps the technical requirements of Network Applications to the capabilities of the testbeds, ensuring a seamless deployment process.

This final release not only showcases the successful implementation and operational readiness of the experimentation infrastructure but also sets the stage for future innovations. The lessons learned and the infrastructure established by the 5GASP project lay a solid foundation for continuous advancement in the realm of 5G networking and services, ensuring a lasting impact on the field of telecommunications research and development.

Annex A – Revisions

This includes a table with every revision per subsection.

Section	Revision
2.1	Added the aspect of the potential 5GASP certification, explicitly mention "NFV" at the respective Network Application artefacts.
2.1.1	Vacated from TOSCA model integration and steered towards the inherent support of "cloud-native" Network Applications.
2.1.2	Added a note about the inherent expansion of the offered network slices capabilities.
2.2.1	Added Use Case #03 and #04.
2.2.2	Merged the previously identified 5GASP NF Developer and 5GASP Network Application Developer actors into 5GASP Developer. Added Use Case #02, specifically mention "NFV" packages in #03 and #04, updated the testing pipeline description in #06, added certification aspect in #07, added testing and certification aspects in #08, added #14 to capture certification results' access, update figure to include NFV artefact development, update figure to include testing and certification aspect.
2.2.3	Merged the previously identified Service Designer and Service Provider actors into Service Designer. Explicitly mentioned that role is responsible for providing the YALM test archive in Use Case #05, added #06, added certification aspect in #07, added testing and certification aspects in #08, added #14 to capture certification results' access, update figure to exclude network slice design.
2.2.4	Updated Use Case #05 to also incorporate the product entity, added #06 and #07.
3.1	Reflected the latest NODS architecture. Removed the Network Application transformation service from the architecture, 5GASP Service Orchestrator is not a static component but can be deployed following the orchestration flow and offered as-a-Service, introduced the Network Application onboarding & triplet design UI, introduced the Kroki service, added 5GASP contribution towards ETSI SDG OSL.
3.2.1	Added the implementation of Network Application onboarding and triplet design portal.
3.2.2	Added the support of 5GASP experimentation APIs service towards the Network Application onboarding and triplet design portal.
3.2.5	Added Kroki in the internal services of NODS.
3.3.3	Reflected the latest modelling of the 5GASP triplet, which incorporates also the testing entity as a matching Service Specification with direct connection with a respective Service Test Specification.
4.2	Removed unused E5 interface.
4.2.1	Finalized the Option B as the nominated NODS-facilities interconnectivity choice.

4.3	Added the support of the common ICT-41 Marketplace through TMF standard interfaces.
4.4	Added this section about 5GASP Harbor Helm Chart and images repository.
5	Updated the infrastructure management UIs with the latest additions, updated the testing portal UI with developer-defined tests support, updated the service portal UI with the latest public catalogues and the ICT-41 Marketplace, updated the service deployment overview with the graphical representation of the deployed services' relationships, added a replicable deployment information extraction mechanism via the rule engine, updated the results report, added the new triplet design UI.
6.1	The workflow of the Inter-domain solution was updated. Previously, the coordination between the VPN Nodes was achieved by having NetOr mediating it. This document describes the newly implemented decentralized coordination approach, where the VPN Nodes coordinate through a DNS Server. Thus, the VPN Nodes are entirely responsible for propagating their information and establishing the VPN tunnels.
7.1.1	Added the VPN health monitoring system.
7.2.2	Added the NEF emulator as a testing tool.
7.2.3	Added Linux Traffic Control as a testing tool.
7.2.4	Added the three supported test setups.
7.3.1	Updated Patras facility to reflect the addition of several new edge sites.
7.3.2	This section was updated to reflect the standardization of ITAv's Network Slicing VNF, which, from Q3 2022 onwards offers a SOL005 API.
7.3.6, 7.4.6	Updated descriptions of ORO's slicing capabilities and Monitoring Platforms.
7.3.4, 7.4.4	Updated the "Future enhancements" section of the same sections of both slicing capabilities and monitoring implementations and any ongoing work.
7.4	Added this newly introduced section to cover the 5GASP federated Prometheus monitoring platform.

Annex B – Security Audit

This annex documents the security compliance of the 5GASP publicly available platforms, according to the OWASP Application Security Verification Standard (ASVS). Therefore, we analyzed the 5GASP Website, the Community Portal, the Developers Forum, the CI/CD Portal, the Portal, and the Harbor platforms in terms of Authentication, Session Management, Access Control, Validation, Sanitization and Encoding, Stored Cryptography, Error Handling and Logging, Data Protection, Communication, Malicious Code, Business Logic, Files and Resources, API and Web Service, and Configuration controls. The most problematic issues found in each of these controls are as follows.

In the Authentication control, all platforms with authentication (the Community Portal, Developers Forum, Portal, and Harbor) failed to ultimately ensure that users create passwords with the proper length (at least 12 characters), and some of them (the Community Portal and Harbor) wrongly forced users to make passwords with composition rules. Moreover, all of them, except for the Developers Forum, did not verify if the password the user tried to create was already compromised, a lack of action that contributes to the poor security of the authentication mechanism. In addition to these problems, some functional-related ones are specific to some platforms. First, the Community Portal's reCAPTCHA was not adequately implemented, allowing a crafted attacker to still guess a user's password through brute force. Second, all applications, except the Developers Forum, do not verify the user's email (although, in the case of the Harbor platform, the user account was pre-created by an administrator). Not only that, but the email verification on the Community Portal and the Portal seems not to be working (for example, when the user indicates that he forged the password, neither one can send an email with the indications on how to create a new password). Finally, another issue found when analyzing the authentication-related subcontrols was that the Portal's user account security-related management page was not directly accessible through the main webpage. We, as assessors, were forced to request the endpoint for this page from an administrator to be able to update the user's password, for instance. As a last note about the Authentication control, we did not analyze the CI/CD Portal's authentication as it does not have a proper authentication method. Although the web application's initial page resembles one of authentication, its form requests a Test ID and an Access Token, which are supposedly given to an administration by the Portal. We would recommend a better link between both platforms, the Portal and CI/CD Portal, for the latter to have a proper authentication phase and management.

Then, regarding the Session Management control, all assessed platforms (the Community Portal, Developers Forum, Portal, Harbor, and CI/CD Portal) have problems with session cookie handling. All of them have session cookies without the 'Secure' attribute set, and most of them do not have the 'SameSite' attribute set (the Community Portal, Portal, Harbor, and CI/CD Portal), while the Developers Forum has it set to 'Lax.' Furthermore, the Portal application stores the user's session tokens in the browser's local storage instead of using a session cookie. Since browsers nowadays have mechanisms to protect session tokens against attacks such as cookie stealing through Cross-site Scripting (XSS), the method hinders that protection. Another serious problem with the Portal's session management is the lack of token invalidation after the user logs out, allowing an attacker to steal the cookie to hold the session still after the user logs out. Lastly, in line with the note given for the Authentication control, the CI/CD Portal does not appropriately manage its sessions. On the one hand,

because there is no user authentication and authentication, and session are usually in pairs. On the other hand, the session tokens are always the same and equal to the Test ID and Token requested upon the platform's "authentication" procedure.

As for Validation, there are only a few problems regarding the Website and the CI/CD pipeline. Both have close to no input, so the attack surface is minimal. In terms of the Community Portal, Portal, and Harbor, they need more verification in terms of data validity. These, however, are not significant. In the Community Portal, it is possible to insert anything as a company URL, while in the Portal, it is possible to insert non-existing times to filter results. As for Harbor, the internal file structure is not verified thoroughly, meaning inserting invalid YAML files is possible. These are all problems that will only create problems for the user. Finally, the Developers Forum accepts input without limiting size. This characteristic means that it is possible to starve the system of resources by creating a large draft.

Regarding Cryptography, the black box tests did not reveal any instance where the cryptographic modules failed insecurely. However, this result is limited by the nature of the test.

In terms of the Error Handling and Logging controls, the assessment is limited, as there is no access to internal logging systems. Two systems with default error pages leak the NGINX server version: the CI/CD Portal and Portal. The Website does not throw an adequate error when accessing /update. This endpoint should not be this visible, and the error thrown makes it clear that it differs from all others. As for the Community Portal and Developers Forum, if enough pressure is put on either system, a CISCO switch login page is accessible. This is a severe security problem and should be addressed.

As for Data Protection, there are two main problems. The first is the lack of anti-caching headers, which affects the Website and CI/CD portal. The second is the lack of ability to delete user data. This applies to the Community Portal, CI/CD portal, the Portal, and Harbor.

As for the Communications control, most subcontrols are valid on all platforms. Nevertheless, we discovered that the Portal and Harbor platforms still allowed the usage of TLS 1.0 and 1.1, and neither accepted the latest version of TLS (1.3). Moreover, although the subcontrol 9.2.1 is of Level 2, and we only assessed Level 1 subcontrols, it is essential to note that the Harbor's web application certificate is invalid since it was created for some other than the "5gasp.eu" domain.

Concerning the Malicious Code controls, all systems import and use external components without verifying their integrity. Moreover, Harbor has an invalid SSL certificate.

There were Business Logic issues in the Portal, where an experimenter could alter the state in the business flow arbitrarily, causing (for instance) unauthorized deployments to begin.

The Files and Resources had limited testing due to the black box methodology. However, the Developers Forum allows for potential Denial of Service (DoS) attacks by exploiting the profile image uploading form. Even though the form will throw an error if the file is over 4 MB, it only does so after uploading the whole file to the server, hindering the effectiveness of that control.

As for API and Web Services, many invalid request methods either are accepted or have inconsistent responses on the Website, Community Portal, CI/CD Portal, Portal, and Harbor. Moreover, the Developers Forum has inconsistent responses to invalid characters. The biggest problem, however, is with the CI/CD portal, which presents its access tokens in the URL.

Finally, in the Configuration control, all systems have outdated components. Many are associated with well-known vulnerabilities, so they should be updated soon. There are also many headers that are not present.

References

[1] 5GASP, “D2.3 : The 5GASP Revised Reference Architecture and Community Components”, 2023. [Online]. Available: https://www.5gasp.eu/assets/documents/deliverables/D2.3%20The%205GASP%20Revised%20Reference%20Architecture%20and%20Community%20Components.pdf. [Accessed September 2023].

[2] 5GASP, “D3.1 5GASP experimentation services, middleware and multi-domain facilities continuous integration”, 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] 5GASP, “D3.2 5GASP experimentation services, middleware and multi-domain facilities continuous integration, 2nd release”, 2022. [Online]. Available: https://www.5gasp.eu/assets/documents/deliverables/D3.2%205GASP%20experimentation%20services,%20middleware%20and%20multi-domain%20facilities%20continuous%20integration.pdf. [Accessed September 2023].

[4] JFrog Artifactory, [Online]. Available: https://jfrog.com/artifactory/. [Accessed March 2024].

[5] Harbor, [Online]. Available: https://goharbor.io. [Accessed March 2024].

[6] 5GASP, “D6.3 Final progress report on Network Applications Community & Certification process”, 2024. [Submitted].

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

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

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

[10] 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].

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

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

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

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

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

[16] 3GPP, "TR 28.801; Telecommunication management; Study on management and orchestration of network slicing for next generation network (Release 15)," 2018.

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

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

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

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

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

[22] NetOr1: R. Direito, D. Gomes, J. Alegria, D. Corujo, and D. Gomes, “NetOr: A Microservice Oriented Inter-Domain Vertical Service Orchestrator for 5G Networks”, JISA, vol. 14, no. 1, pp. 136–150, Sep. 2023.

[23] NetOr2: https://github.com/ATNoG/netor [Online]

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

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

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

[27] ETSI SDG OSL, “ETSI Software Development Group for OpenSlice,” [Online]. Available: https://osl.etsi.org/. [Accessed March 2024].

[28] NEF Emulator: https://github.com/EVOLVED-5G/NEF_emulator [Online]

[29] Helm, [Online]. Available: https://helm.sh/. [Accessed March 2024].

[30] Zero-touch provisioning [Online]. Available: https://en.wikipedia.org/wiki/Zero-touch_provisioning. [Accessed March 2024].

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

[32] O-RAN. [Online]. Available: https://www.o-ran.org/. [Accessed March 2024].