VertIcal demos over Common large scale field Trials fOr Rail, energy and media Industries

D2.5 5G VICTORI Infrastructure Operating System (5G-VIOS) – Initial Design Specification

This project has received funding from the European Union's Framework Programme Horizon 2020 for research, technological development and demonstration

5G PPP Research and Validation of critical technologies and systems

Project Start Date: 1st June 2019 | **Duration**: 36 months

Call: H2020-ICT-2019 | **Date of delivery**: 2020-07-31

Topic: ICT-19-2019 | Version 1.0

Project co-funded by the European Commission

Under the H2020 programme

Dissemination Level: **Public**

Grant Agreement Number:	857201
Project Name:	VertIcal demos over Common large scale field Trials fOr Rail, energy and media Industries
Project Acronym:	5G-VICTORI
Document Number:	**D2.5**
Document Title:	5G-VICTORI Infrastructure Operating System – Initial Design Specification
Version:	1.0
Delivery Date:	2020-07-31
Responsible:	Digital Catapult (**DCAT**)
Editor(s):	Kostas Katsaros (**DCAT**)
Authors:	Kostas Katsaros (**DCAT**), Charles Turyagyenda (**DCAT**), Shadi Moazzeni (**UNIVBRIS**), Navdeep Uniyal (**UNIVBRIS**), Anderson Bravalheri (**UNIVBRIS**), Anna Tzanakaki (**UNIVBRIS**), Adriana Fernández (**i2cat**), Eric Troudt (**FhG**), Tanya Polity (**UoP**), Christos Tranoris (**UoP**), Panagiotis Papaioannou (**UoP**), Spyros Denazis (**UoP**), Marius Iordache (**ORO**), Catalin Brezeanu (**ORO**), Ioan Constantin (**ORO**), Reschl Philipp (**ORO**), Vasileios Theodorou (**ICOM**), Marievi Xezonaki (**ICOM**), Paris Flegkas (**UTH**), Jesús Gutiérrez (**IHP**).
Keywords:	MANO, 5G-VIOS, Multi-Domain Orchestration, OSM, ONAP, Service Brokering
Status:	Final
Dissemination Level	**Public**
Project URL:	https://www.5g-victori-project.eu/

Revision History

Rev. N	Description	Author	Date
0.0	Draft Table of Contents (ToC)	Kostas Katsaros (**DCAT**)	2020-02-19
0.1	Revised ToC, initial input	Kostas Katsaros (**DCAT**)	2020-03-13
0.2	Contributions assignment	Kostas Katsaros (**DCAT**)	2020-04-27
0.3	Requirements collection	Kostas Katsaros, Charles Turyagyenda (**DCAT**)	2020-05-05
0.4	Section 2, 5G-VIOS Architecture, Security	Kostas Katsaros, Charles Turyagyenda (**DCAT**) Marius Ioardache (ORO) Shadi Moazzeni, Navdeep Uniyal (**UNIVBRIS**) Adriana Fernández (**I2cat**) Marievi Xezonaki (**ICOM**)	2020-06-20
0.5	Section 2, Security	Kostas Katsaros (**DCAT**) Christos Tranoris (UoP) Marius Iordache (ORO) Shadi Moazzeni, Anderson Bravalheri (**UNIVBRIS**) Eric Troudt (**FhG**) Paris Flegkas (**UTH**)	2020-07-07
0.6	First Draft for internal review	Kostas Katsaros (**DCAT**) Charles Turyagyenda (**DCAT**) Shadi Moazzeni (**UNIVBRIS**)	2020-07-15
0.7	Addressing internal review comments	Kostas Katsaros (**DCAT**) Marius Iordache (ORO) Anna Tzanakaki (**UNIVBRIS**)	2020-07-24
0.8	Overall revision	Jesús Gutiérrez (**IHP**)	2020-07-28
1.0	Submission of the document to the EC	Jesús Gutiérrez (**IHP**)	2020-07-31

Table of Contents

List of Figures

List of Tables

Executive Summary

5G-VICTORI is built on the concept of a common inter-facility orchestration platform that is able to broker network services across multiple domains and facilities that comprise the 5G-VICTORI Platform. This document sets the basis of that ambition in 5G-VICTORI – to define the architecture of the cross-domain orchestration platform that can enable the offering of cross-facility vertical services. This is referred to as the 5G-VICTORI Operation System (5G-VIOS).

This deliverable identifies the key requirements (functional and non-functional) of such a platform and investigates the state-of-the-art of solutions from other previously running 5G-PPP projects or open source solutions tackling similar multi-domain orchestration challenges. Some of these projects set the starting point for 5G-VIOS, including 5GUK Exchange and 5G-PICTURE. As most of the facilities are using ETSI Open Source MANO (OSM) as their Network Function Virtualization Orchestrator (NFVO), interfacing and expanding on its capability would be key for the project. This work represents a starting point for the technical design with an initial 5G-VIOS specification including a high-level design architecture and component description. Preliminary analysis of the processes and the components that support them, security and workflows are included but are expected to be enhanced at later stage once implementation is complete.

A second version of this deliverable (D2.6) will be produced later in 5G-VICTORI's timeline, which will reflect the updated design as the implementation of the end-to-end (E2E) reference architecture of the 5G VICTORI Platform progresses (D2.4) and implementation of 5G-VIOS components are finalised.

1 Introduction

5G-VICTORI will provide a vertically-optimized common platform supporting an attractive, cost and energy efficient way to address the requirements and business needs of vertical industries using 5G. This platform will provide the flexibility and agility needed to support the rapid changing business environment and its associated services. To achieve this, the 5G-VICTORI Platform will be implemented over the 5G facilities developed in the framework of ICT-17 in 5G-VINNI (Patras, Greece) , 5GENESIS (Berlin, Germany) , 5G-EVE (France/Romania) and extend to the 5G UK facility (Bristol, UK) as depicted in Figure 1-1.

To enable the offering of a single E2E platform across its multiple facility sites, 5G-VICTORI will provide interconnection and interworking creating a common infrastructure of integrated network and compute/storage resources. This is a key aspect of the 5G vision as these resources will be able to be brokered to and accessed on demand by any service or application, enhancing resource utilization efficiency and providing measurable benefits for the vertical industries in terms of cost, scalability, sustainability and management simplification.

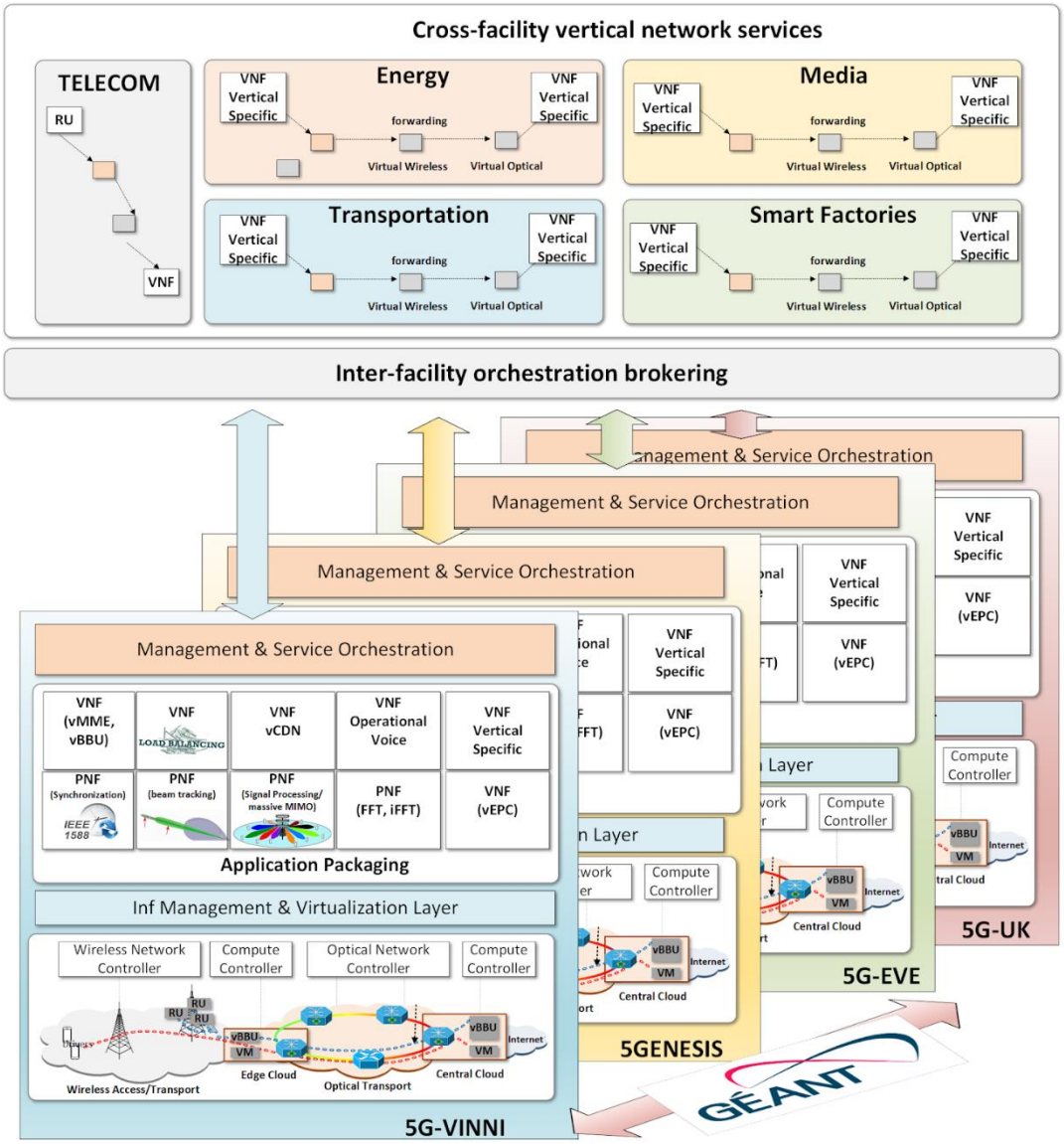

Figure 1-1 5G VICTORI use case overview

Managing such a diverse environment requires a multi-domain orchestration framework. Each domain can be interpreted in two ways:

- **As a technology domain**, which means orchestrating resources across multiple technology domains, e.g., Radio Access Network (RAN), Core Network (CN), Multi-access Edge Compute (MEC), Wireless & Optical transport network, etc., in a single geographical domain and operator. As there are multiple resource management and orchestration frameworks for resource virtualisation and automated provisioning of resources and services, it is necessary to unify management and orchestration of different technology domains to realise E2E software defined infrastructures suitable to host 5G services.
- **As an operator domain**, which means orchestrating resources and/or services according to operator policies using domain orchestrators belonging to multiple administrative domains. In order to realise E2E orchestration, interaction between multiple infrastructure providers must be addressed at different levels, including resource management and orchestration, service management and orchestration and inter-operator Service Level Agreement (SLA) fulfilment.

Flexible service provisioning over cross-platform slices will rely on combining and orchestrating a set of network functions through Service Chaining (SC) over the integrated programmable infrastructure. Cross-domain infrastructure slicing, SC and orchestration will be facilitated by merging together the SDN reference architecture and the ETSI NFV standard and leveraging existing developments of orchestration platforms from the Phase-2 5G-PPP project 5G-PICTURE[1] and 5GUK Exchange [1] [2]. A key aspect that will be also considered is the support of interoperability with legacy software (SW) and hardware (HW) technologies and architectures, which dominate the current vertical industries ecosystem.

To facilitate inter-domain orchestration and interconnection, a thin inter-domain orchestration brokering solution will be developed, namely **5G-VICTORI Infrastructure Operating System (5G-VIOS)**. The 5G-VIOS is a common platform that enables management of slices, resources and orchestration of services across different sites. The 5G-VIOS will provide network service deployment across different sites, dynamic layer-2 (L2) or layer-3 (L3) cross-site service interconnections, inter-site service composing and on-boarding, E2E slice monitoring and management for the deployed E2E services. The design of 5G-VIOS will consider the current status of the Management and Orchestration (MANO) platform in each facility and will work closely with each one to reflect the facility extensions to the common multi-site orchestration platform. This will build on top of the orchestration solutions of each facility, to provide E2E services across the different sites. The cross-domain orchestrator will implement suitable drivers to communicate with the Northbound Interfaces (NBIs) of the site orchestrators, while also provisioning and orchestrating the necessary Layer 3 (L3) or Layer 2 (L2) dynamic connectivity across the data plane of the sites. This solution will be used to extend and/or combine trials to be demonstrated in each site.

This document identifies the key functional and non-functional requirements of the 5G-VIOS platform, driven by the use case (UC) requirements captured in deliverable D2.1 [1]. It also provides a thorough analysis of the state of the art of MANO systems and platforms from previous projects and open source solutions, identifying gaps that such systems have in order to fulfil the requirements of a multi-domain orchestration brokering platform that 5G-VIOS is aiming to fulfil. It then goes to define the initial specification of 5G-VIOS, describing its architecture, components and interfaces. This is the initial definition of 5G-VIOS and it is expected that more details will be included in the subsequent version in deliverable D2.6, particularly focusing on the implementation and evaluation of the platform.

Organisation of the document

This document comprises two main sections. Following the Executive Summary and Introduction sections:

Section 2 describes the 5G-VIOS functional and non-functional requirements, followed by a state-of-the-art and gap analysis of MANO solutions.

[1] 5G-PICTURE Project, https://www.5g-picture-project.eu/

Section 3 describes the architecture, components and interfaces of 5G-VIOS, including security and preliminary workflow examples for processes that 5G-VIOS will be responsible for, such as the network service (NS) Life-Cycle-Management for establishing and migrating a NS.

Finally, Section 4 concludes this deliverable

2 Requirements and Related work analysis

5G is considered to be the technology that will accommodate the development and management of innovative services with stringent and diverse requirements from end users, calling for new business models from the industry, such as those demonstrated in 5G VICTORI. In this context, the development and efficient management of NSs serving specific vertical industries and spanning across multiple administrative domains and heterogeneous infrastructures is challenging.

The main challenges concern efficient provision of NSs considering the Quality of Service (QoS) requirements per vertical industry application and for each instance of those applications, along with the optimal usage of the allocated resources. In addition, and particular to 5G-VICTORI, mobility management for seamless and un-interrupted service provisioning is sought. The functional and non-functional requirements are reported in sections 2.1 and 2.2 respectively

Towards addressing these challenges, several solutions are being developed or have been developed by other 5G-PPP projects, open-source communities and vendors. These innovative approaches have been developed for managing and orchestrating such NSs, include OSM, SONATA, Open Networking Automation Platform (ONAP), and have been adopted and enhanced through projects such as 5G-PICTURE, 5GTANGO, 5G-TRANSFORMER, the ICT-17 platforms (5G-VINNI, 5GENESIS, 5G-EVE) and others. A brief description of each one is included in section 2.3, and a gap analysis where 5G-VICTORI aims to innovate is included in section 2.4.

2.1 Functional Requirements for 5G-VIOS

The 5G-VICTORI reference architecture is following ETSI NFV standards. Each site/domain has its own ETSI NFV MANO compliant system, comprising of three main functional blocks providing the following services and illustrated in Figure 2-1 [3]:

- NFV Orchestrator:
  - on-boarding of new Network Service (NS), VNF-FG and VNF Packages;
  - NS lifecycle management (including instantiation, scale-out/in, performance measurements, event correlation, termination);
  - resource management, validation and authorization of NFVI resource requests;
  - policy management for NS instances.
- VNF Manager:
  - lifecycle management of VNF instances;
  - overall coordination and adaptation role for configuration and even reporting between NFVI and the E/NMS.
- Virtualised Infrastructure Manager (VIM):
  - controlling and managing the NFVI compute, storage and network resources, within one operator's infrastructure sub-domain;
  - collection and forwarding of performance measurements and events.

For each domain, the functional requirements for the MANO system are reported in ETSI GS NFV-IFA 010 [4] and 3GPP TR 28.801 [5], TS 28.533 [6], whereby "**IFA**" refers to functional requirements for **Interfaces and Architecture**, "**TST**" refers to **testing**, and "**SOL**" refers to **solutions** for protocols and data models.

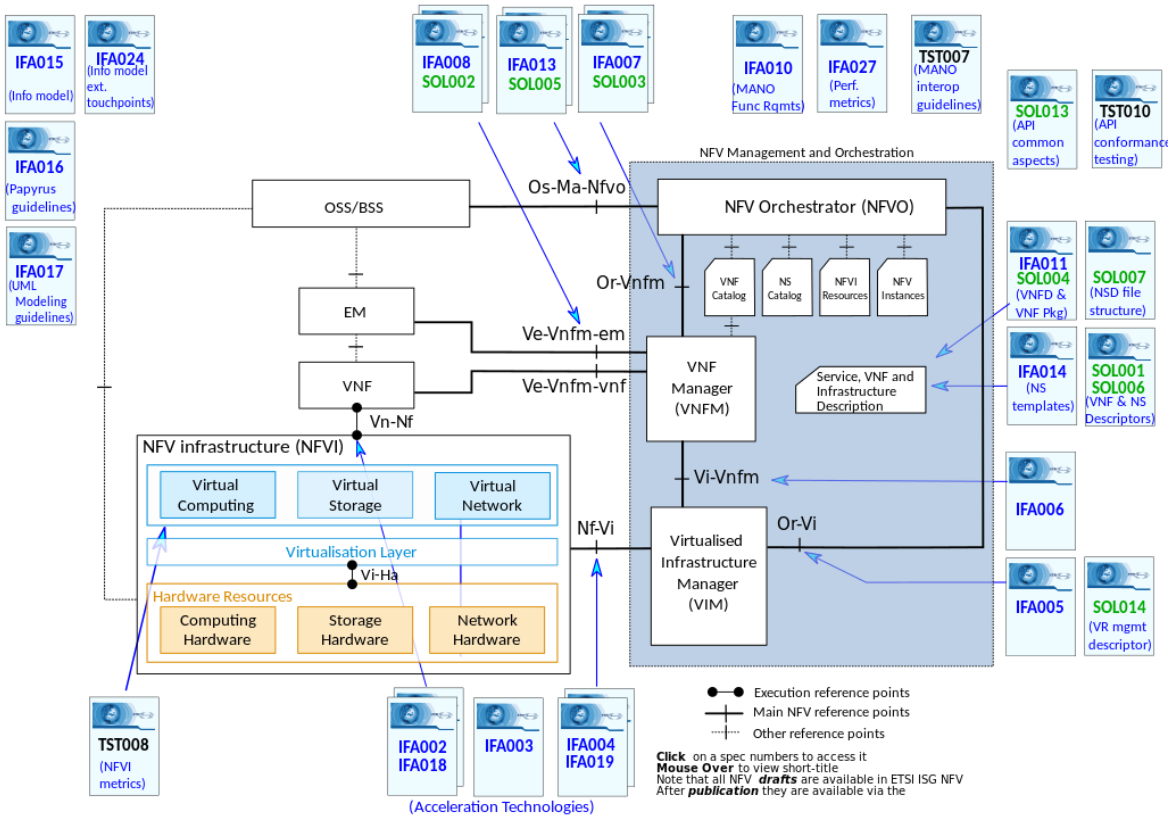

Figure 2-1 ETSI NFV Reference Architecture

5G-VIOS in particular, as a brokering platform aiming at facilitating inter-domain orchestration and inter-connection, would need to interface each domain platforms through the Os-Ma-Nfvo interface on the North Bound Interface (NBI) of the NFVO. The high level functionalities that are expected by 5G-VIOS are listed below and will drive the functional requirements:

- User Access support
  - Management & authentication of users;
  - Graphic user interface.
- Repositories
  - Catalogues of VNF and NS descriptors;
  - Repository of images.
- Service Management
  - Service composition & life-cycle management (LCM);
  - Mobility Management and service migration.
- Network Management
  - Inter-site connectivity management;
  - End-to-End slice management.
- Monitoring & Profiling
- Policy & Security management
- Service Automation

2.1.1 User Access Support

5G-VICTORI stakeholders and users span from vertical end-users, to service and infrastructure providers as well as operators as reported in deliverable D4.1 [7]. Each of these would require different access to the 5G-VIOS platform through a Graphical User Interface (GUI) or dedicated APIs. These requirements are captured in Table 2-1.

Table 2-1 User Access Requirements

ID	Requirement
VIOS-req-01	5G-VIOS shall provide a GUI, such as a portal, by which the users can interact with the platform and the facilities, allowing them to instantiate new network services and manage existing network services.
VIOS-req-02	5G-VIOS shall perform translation and mapping of the user service request(s) to network slice(s) implementing such service.
VIOS-req-03	5G-VIOS shall allow management and authentication of users providing them or their applications (VNFs) with access to the portal and to the 5G-VIOS services through corresponding programmable interfaces (APIs). This should include mechanisms to add new users, modify the access privileges of existing users and revoke/delete existing users.
VIOS-req-04	5G-VIOS shall expose appropriate interfaces to external users, depending on their roles and access rights, to allow them to consume services offered by the 5G-VICTORI system.
VIOS-req-05	5G-VIOS shall provide an interface for managing the lifecycle of all entities: Users, Groups, Tokens, Permissions;
VIOS-req-06	5G-VIOS shall provide secure access to the Administrative users;
VIOS-req-07	5G-VIOS shall provide a Multi Factor Authentication capability;
VIOS-req-08	5G-VIOS shall provide a secure interface in an industry accepted form, such as a REST API to other components that it needs to interact with;
VIOS-req-09	5G-VIOS shall provide user registration services to other components and applications of 5G-VIOS;
VIOS-req-10	5G-VIOS shall provide monitoring and audit resources
VIOS-req-11	5G-VIOS shall differentiate between users with Administrative and non-administrative privileges and provide granular monitoring and control capabilities for the activities performed by the Administrative users.

2.1.2 Repositories

A repository service provides an easy access to catalogues of network function descriptors and corresponding images in order to allow users to compose their services. The repositories requirements are shown in Table 2-2.

Table 2-2 Repositories Requirements

ID	Requirement
VIOS-req-12	5G-VIOS shall provide catalogues of available VNF Descriptors (VNFDs), Physical Network Function Descriptors (PNFDs), Containerised Function Descriptors (CNFDs), Network Service Descriptors (NSDs), NFV instances repository and NFVI resources repository from each connected facility, following the templates and formats described in IFA 014, IFA 011, SOL 001, SOL 004, SOL 006, SOL 007. in order to facilitate the automated provisioning of services.
VIOS-req-13	The 5G-VIOS should be able to update the catalogue entries as informed by the corresponding domain orchestrators.
VIOS-req-14	5G-VIOS shall allow users to add, delete, update, query and copy corresponding images for the VNFs and CNFs. There shall also be a version control mechanism to keep track of the images. These images may be saved centrally or on individual facilities with 5G-VIOS providing the links for those

2.1.3 Service Management

Service Management includes tasks such as service composition, service discovery, service fulfilment, service migration (for mobility) and at the end service decommissioning. The service management requirements are captured in Table 2-3.

Table 2-3 Service Management Requirements

ID	Requirement
VIOS-req-15	5G-VIOS shall allow users to compose network services utilising components and resources (VNFs, PNFs, CNFs) from one or multiple domains/sites. 5G-VIOS should allow both manual and pre-defined (template/blueprint) service composition. 5G-VIOS would then send those detailed requests including information regarding the location of resources, the location of service points, QoS to corresponding domain orchestrations.
VIOS-req-16	5G-VIOS shall provide the users with the ability to request a service from the catalogue along with the expected SLA.
VIOS-req-17	5G-VIOS shall be able to translate the service request to one or more service descriptors reflecting the agreed service levels.
VIOS-req-18	5G-VIOS shall keep association among the network slices generated as result of a single service request.
VIOS-req-19	5G-VIOS shall provide a mechanism to set-up, and terminate one or multiple services instances.
VIOS-req-20	5G-VIOS shall provide the users with the ability to request a service migration from one VNFI (i.e. an edge instance) to another.
VIOS-req-21	5G-VIOS shall be able to migrate network services across VNFIs upon request from user applications or other 5G-VIOS components, while maintaining the required service level and minimum service interruption.
VIOS-req-22	5G-VIOS shall inform the requester (user or component) of completed successful migration of network services.
VIOS-req-23	5G-VIOS should have means for receiving acknowledgement of releasing resources.
VIOS-req-24	5G-VIOS should be able to notify the vertical customer about service instance termination.

2.1.4 Network Management

Following service management, the corresponding inter-networking configurations for the network service and for the inter-domain data paths should be configured, as included in Table 2-4.

Table 2-4 Network Management Requirements

ID	Requirement
VIOS-req-25	5G-VIOS shall be able to request E2E network slice(s) across one or multiple domains based on the network service descriptors.
VIOS-req-26	5G-VIOS shall enable the 5G-VICTORI platform to operate different network slices in parallel with isolation that for example, prevents one slice's data communication to negatively impact services in other slices, offers the required QoS, security and privacy guarantees.
VIOS-req-27	5G-VIOS shall support network slice elasticity in terms of capacity with no impact on the services of this slice or other slice.
VIOS-req-28	5G-VIOS should be able to identify the slice(s) to be decommissioned as a result of a service instance termination and notify the corresponding domain orchestrators for releasing the resources.
VIOS-req-29	5G-VIOS shall be able to setup secure, dynamic inter-connections across different domain/sites using L2 or L3 connections.

2.1.5 Monitoring & Profiling

5G-VIOS will pull monitoring information from the underlying NFVO and applications directly, which would be offered through dashboards to the users and consumed internally for generation of performance profiles. The requirements are captured in Table 2-5.

Table 2-5 Monitoring & Profiling Requirements

ID	Requirement
VIOS-req-30	5G-VIOS must provide the users with tools to monitor the KPIs attained for the requested service, using a common interface and model for configuring what to monitor shall be implemented.
VIOS-req-31	5G-VIOS shall allow to dynamically set-up a traffic monitoring service in any given network slice.
VIOS-req-32	5G-VIOS should be able to map the associated SLA into KPIs to be monitored during the slice execution lifecycle.
VIOS-req-33	5G-VIOS should be able to identify the monitoring mechanisms to be de-activated as a result of a service instance termination.
VIOS-req-34	5G-VIOS shall provide the mechanisms to generate user and service performance profiles based on gathered monitoring KPIs.

2.1.6 Policy & Security Management

Security and privacy are important for any 5G platform. This is liked with the user access management to develop appropriate mechanisms associated with these. The policy and security requirements are summarized in Table 2-6.

Table 2-6 Policy and Security Requirements

ID	Requirement
VIOS-req-35	5G-VIOS shall allow negotiation and monitoring of SLAs, with appropriate granularity according to the final service characteristics
VIOS-req-36	5G-VIOS should detect and mitigate DoS attacks from a malicious user behaviour.
VIOS-req-37	5G-VIOS should provide isolation among users' workflows and network slices catering for multi-tenancy.
VIOS-req-38	5G-VIOS shall have the capability to conform to service-specific security assurance requirements for each network slice across all the utilised domains.

2.1.7 Service Automation

Service automation, which is also known as DevOps, allows for simplified and automated process for running experiments, onboarding new services and network functions. This is linked with the user access requirements with respect to the APIs offered to allow automation and with the repositories that enable such simplified access to service descriptors. The automation requirement is captured in Table 2-7.

Table 2-7 Automation Requirements

ID	Requirement
VIOS-req-39	5G-VIOS shall allow automated service delivery via repeatable, simplified and auditable processes

2.2 Non-Functional Requirements for 5G-VIOS

In addition to the functional requirements in the previous section, non-functional requirements are important (shown in Table 2-8) and relate to availability and reliability of 5G-VIOS. Finally,

interoperability through the adoption of open standards API and best practices would ensure the usability of the platform.

Table 2-8 Non-Functional Requirements

ID	Requirement
VIOS-req-40	5G-VIOS shall adhere to industry multi-tenancy requirements including isolation, scalability, elasticity and security, where security is meant to provide protection to prevent attacks, denial of service or information leaking.
VIOS-req-41	5G-VIOS should be reliable with a mean-time-to-failure (MTTF) of 100.000 hours.
VIOS-req-42	5G-VIOS should be available (as carrier class component providing 5 nines availability).
VIOS-req-43	5G-VIOS should keep responsiveness for vertical user requests. The 5G-VIOS should provide response times as an interactive system.
VIOS-req-44	5G-VIOS system should follow a modular, programmable, micro-service based architecture, that supports control of multiple technologically diverse domains: cloud, multi-layer WAN, NFV, IP/MPLS, and more.

The standards compliance and interoperability requirements are shown in Table 2-9.

Table 2-9 Standards & Interoperability Requirements

ID	Requirement
VIOS-req-45	5G-VIOS should interface with end users and with the facilities utilising open standards APIs.
VIOS-req-46	5G-VIOS should follow the ETSI NFV standards where possible.
VIOS-req-47	5G-VIOS should employ TOSCA-based templates that enable rapid network services programmability and self-service in operating the network.

2.3 State of the art

5G management and orchestration of the programmable infrastructure allows operators to dynamically control the resources, slices and services, implementing software network services, uncoupling the hardware equipment's from software NFVs/VNFs components, enabling for orchestration network functions running on traditional IT servers.

The 5G industry is driven by the network and application transformation, enabling technologies such as NFV and cloud-native developments, one of the most challenging for network operators being the management and network automation. This is seen as autonomous management for future zero-touch automated networks and service management and orchestration in multivendor scenario.

One of the ambitions for 5G is to build a single platform to perform E2E management of network services and infrastructure resources within the various heterogeneous network segments of an administrative domain, e.g., core, metro, access. Standardization bodies such as IETF and ETSI have created standards and guidelines and open source communities have been formed such as OSM, Open Baton, SONATA and ONAP. However, a 5G platform is also expected to support multi-domain services that, up to now, experience best effort connectivity and interconnections that do not take into account the service requirements. A 5G platform that federates different administrative domains in order for services to get the required E2E QoS and are deployed based on their requirements. Collaboration and coordination among administrative domains are needed so E2E services can be quickly deployed and orchestrated in an automatic way. Having multiple 5G network domains combining diverse 5G technologies and services, a feature-rich environment is created that supports innovative E2E 5G services. However, there is no standard to define a common interface among MANO systems that could facilitate the design of a multi-domain architecture and the introduction of multiple 5G networks.

2.3.1 5G-PICTURE

5G-PICTURE is a 5G-PPP Phase 2 project focusing on next generation converged infrastructures integrating a variety of wireless access network technologies through novel wireless, optical and packet network solutions. This infrastructure interconnects a large number of "disaggregated" compute/storage and network elements, forming a number of different domains and deploys the concepts of hardware programmability and network softwarisation to facilitate the Dis-Aggregated RAN approach. This enables the provisioning of any service across the infrastructure by flexibly and efficiently mixing-and-matching network, compute and storage resources. All this programmability is controlled and orchestrated by the 5G Operating System (5G OS) [8] designed in 5G-PICTURE. 5G OS addresses a set of major challenges related to the management, orchestration and control of resources, such as the virtualisation support and the creation of a layer of abstraction.

As every operating system, 5G OS is also responsible for:

1. providing control over heterogeneous resources,
2. seamlessly recovering from resource failures, and
3. being efficient even if the pool of resources is extended.

The existence of multiple instances for the Orchestrator, Controller and NFV Management and Orchestration (MANO) components in Figure 2-2, depicts the necessity to address efficiently the three aforementioned points. At first, the support of multiple domains as group of resources, which may be providing the same functionality using different technologies, enables the cross-concept handling of heterogeneous resources. The abstract layer built on top of these domains facilitates unified management, orchestration and control of the underlying resources. The cooperation of multiple domains is supported by the coexistence of multiple interconnected components (Controller, NFV MANO), one for each domain. Their strong interoperability and efficient utilisation are challenges that were addressed in 5G-PICTURE, with the aim to provide a failure resilient and scalable 5G OS.

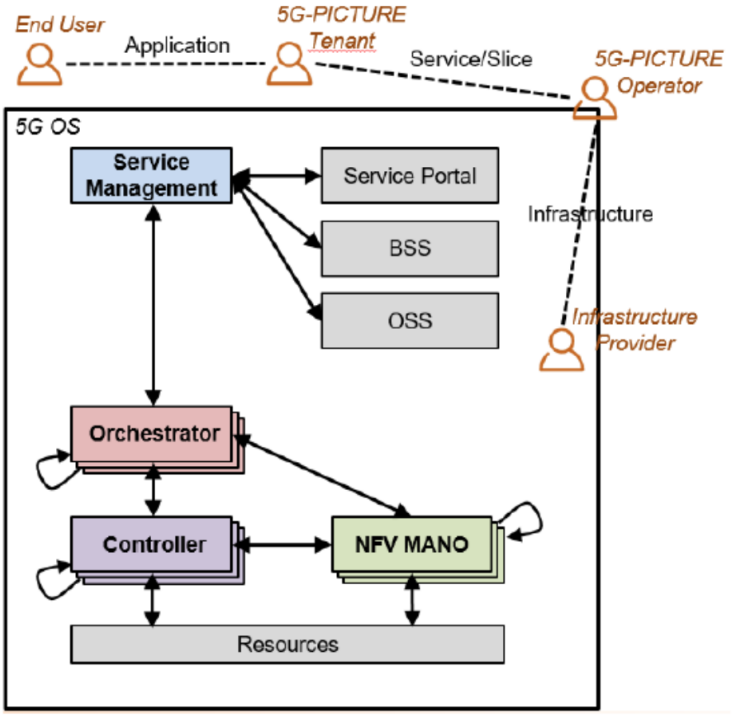

Figure 2-2 5G OS high-level architecture

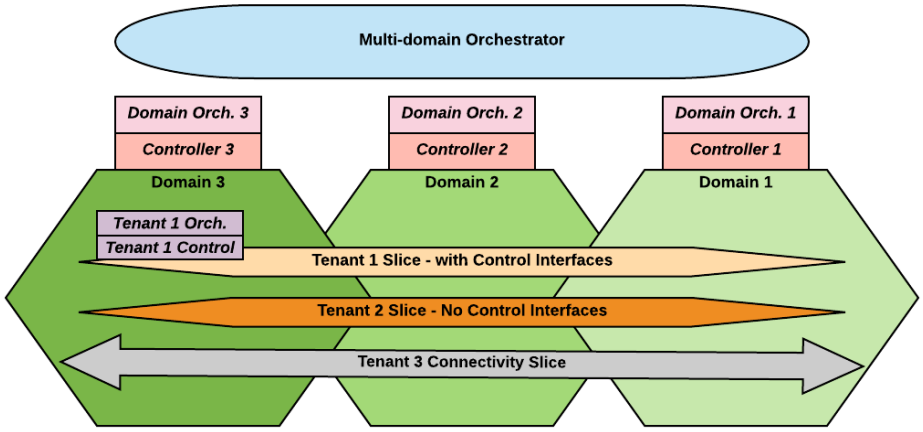

Figure 2-3 5G OS Architecture used to provide multi-tenant slices across different administrative domains

A 5G OS integrated demonstrator [9] was built by 5G-PICTURE with the aim to validate how the 5GOS can be used to orchestrate the automated provisioning of E2E NSs over a distributed and multi-domain compute and network infrastructure.

In addition to the design of the 5G OS architecture, the key component enabling orchestration of end-to-end services the Multi-Domain Orchestrator (MDO) interacts with the entire network and compute domains involved in the service creation.

The 5G OS developed by 5G-PICTURE is based on three main pillars:

1. Presence of Administrative Domains that own and provide resources such as connectivity, compute and network devices.
2. MDO that can provision E2E services using resources provided by different Administrative Domains.
3. Interfaces between Orchestrators owned by different Administrative Domains (i.e. a Domain Orchestrator) and the MDO.

Figure 2-3 shows how these concepts can provide different types of end-to-end slices that map to one or more services with clear business value. Three Administrative Domains can be seen: Domain 1, 2 and 3. Each of these has a single Domain Orchestrator and Domain Controller which provides some kind of resource interface to the MDO. The key responsibility of a Domain Orchestrator is to provide an accurate view of available resources to the MDO and on request reserve/configure these for use by a service. The MDO is responsible for taking the provided resources and integrating them within a network slice to provide isolation, abstraction and other benefits. It should also be able to maintain the slice in case one or more resources that are being used are no longer available.

Three types of slices (or services) are shown in Figure 2-3, which has three tenants being supported. Tenant 1 has requested a slice with full control interfaces that allow deployment of a Tenant Orchestrator and Control (e.g. a MVNO UC). Tenant 2 has requested a slice without any control interfaces. In this case all functions are externally managed or are unmanaged. Tenant 3 has requested a connectivity slice; this is the simplest possible slice.

These domains are also recursive. In the sense that a Domain may take one or more resources from different providers repeating the same architecture at a smaller scale.

2.3.2 5GUKExchange

The 5GUK Exchange (5GUKex) is a novel hierarchical architecture that enables E2E service orchestration while also allowing operators to maintain full control of their infrastructure. 5GUK

Exchange builds a multi-operator API based on ETSI standards, allowing operators to use their existing MANO systems (for instance OSM) for the single domain orchestration. Particularly, the operators can hide any confidential infrastructure information and provide flexibility in selecting any underlying SDN and NFV technologies. So, the 5GUKex, as a lightweight hierarchical inter-domain orchestration platform performs multi-operator coordination and service interconnection [2].

The architecture of the 5GUKex is illustrated in Figure 2-4. The 5GUKex interfaces with the Local island orchestrators to enable E2E service provisioning and interconnection across autonomously orchestrated testbed islands. The local island orchestrators implement on top of them an Island Proxy component that complies with the API that the 5GUKex exposes to automate the processes of exposing the island service capabilities, reserving service resources and service provisioning and termination. Furthermore, the Inter-domain connectivity manager (IDCM) component allows experimenters to plug-in various underlying network technologies to create an E2E network. Figure 2-5 shows the control flow and the interaction between main components of 5GUKex. These components are:

- **Island Proxy**: This component runs on top of the local Island orchestrator and acts as an intermediary between each island Orchestrator and the Network Service Broker (NSB) component of 5GUKEx. For security purposes, it can act as an isolation layer for the island's policies. It is responsible for forwarding requests from the NSB to the Island orchestrator and the responses (opposite direction). These requests include verifying the available resources on the island and requests to instantiate and terminate network services on the local orchestrator. It is also responsible for registering the island with the 5GUKEx by exposing the available network services (Catalogues) on the local islands to the 5GUKex. The Catalogues are in the form of ETSI MANO NSDs or VNFDs. In addition, during the deployment of NSs it is responsible for creating the local island network using the local SDN controller and sharing the network VLAN information with the NSB to deploy the end to end service.
- **Network Service Broker (NSB)**: This component is located at the southbound part of the 5GUK Exchange architecture (Figure 2-4) and interacts with the Network Service Manager (NSM) and the island orchestrators, particularly through the Island Proxy component. The NSB is responsible for receiving island registration messages and Catalogue updates from the Island Proxies. The NSM invokes the NSB in the event of a request for Network Service Instantiation, Deployment or Termination. The NSB then contacts the relevant Island Proxies to serve the request and check the available resources. Once an NS is deployed, it sends the network endpoint information (VLAN ID, switch port and island ID) to the Inter-Domain connectivity manager (IDCM) to create the network slice between the two islands.
- **Inter-domain Connectivity Manager (IDCM):** This component is responsible for establishing the connectivity among the island testbeds for a desired inter-domain network service. So, it sets up the control plane of the respective islands and connects them to the 5GUK Exchange. It also creates the data path between the islands when an NS is deployed. It utilises an SDN controller to create the E2E layer 2 network and when an NS is terminated it will terminate the connections.
- **Network Service Manager (NSM):** This component is responsible for life cycle management of NSs. It requests the NSB to deploy/terminate an NS and receives island responses and network endpoints used by the running services. In addition, it provides information about the running NSs on the islands and their relevant monitoring data.
- **Network Service Composer (NSC):** This component enables the users to create inter-domain network services (iNS) by combining and selecting different network services or VNFs available on multiple islands. Using the Dashboard, the user can choose network services from multiple islands and sends the request to NSC. Then, the NSC combines the requested Network Service Descriptors (NSDs) and endpoints to create an inter-domain network service descriptor (iNSD).

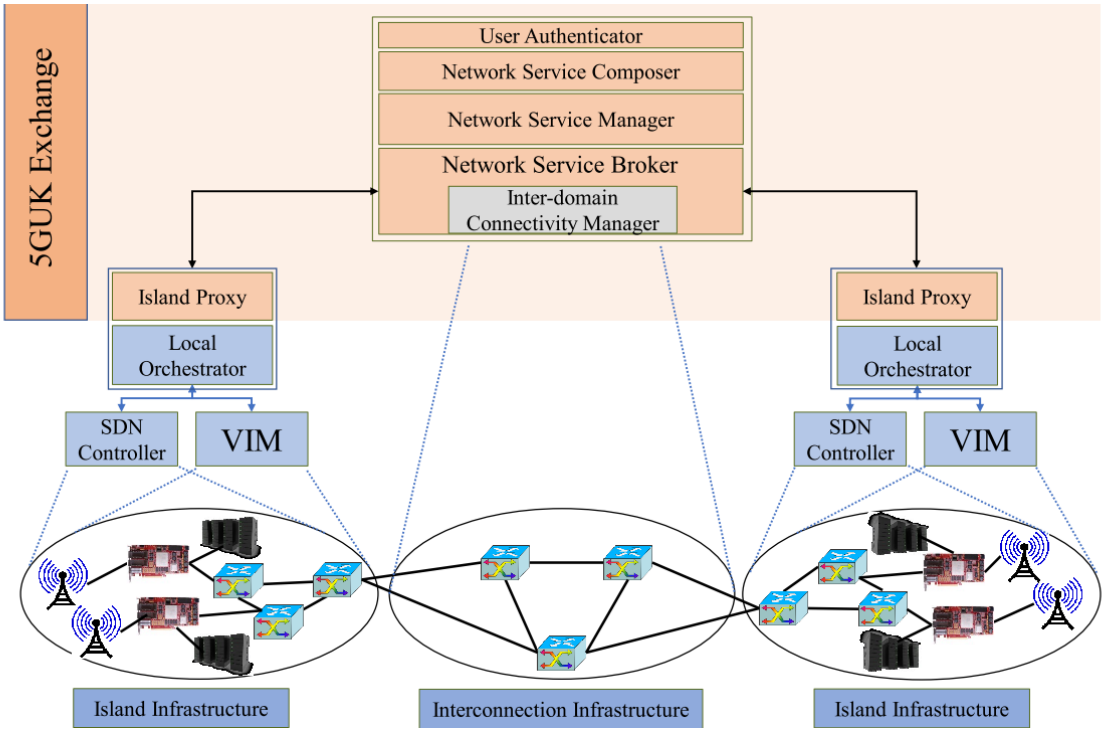

Figure 2-4 The 5GUK Exchange architecture

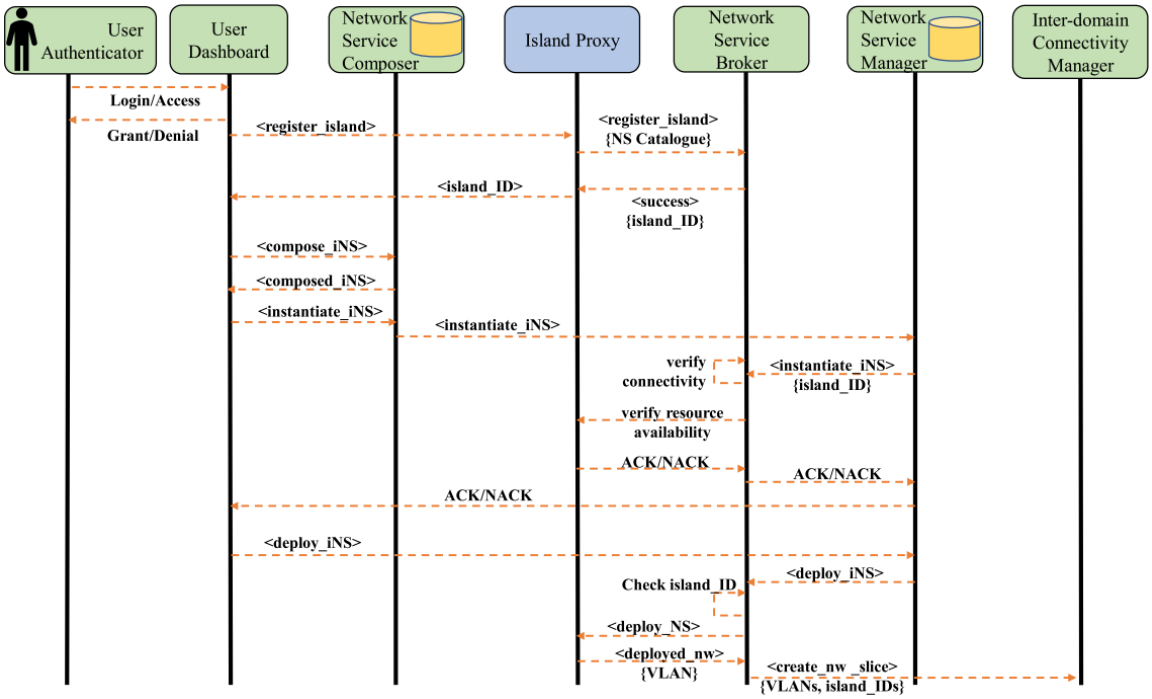

Figure 2-5 The 5GUK Exchange control flow and interaction between the main components

2.3.3 5G VINNI

The 5G-VINNI E2E facility is an ICT-17 platform that aims at interconnecting eight 5G domains across Europe, enabling the onboarding and validation of a wide range of industry UCs. For each facility site, there are different technologies and reference architectures that affect all the affected 5G domains (Radio, Core, NFVI, MANO, and Transport), SW components for service orchestration and interfaces for interworking between facility sites. As far as the implementation of CN functions are concerned those

are built according to the reference architecture, while the 5G RAN implementation is aligned with the 3GPP standards to ensure standards compatibility.

With respect to the service orchestration technologies that are being captured in [10], 5G VINNI represents all major trends that are enlisted in here. In summary, all MANO and NFVI technology choices per facility are described in [10] (OpenStack related technologies). Also service orchestration options and architecture recommendations are a mixture of commercial proprietary and open source solutions including to Nokia Flowone (Norway, UK) capabilities, OSM adoption (Greece, Spain, Germany-Berlin), SONATA in Portugal, and Huawei in Germany (Munich site).

In 5G-VICTORI we will be exploiting the Patras facility from 5G-VINNI. Details of this are presented below.

2.3.3.1 Patras 5G facility

The Patras 5G facility being part of the 5G-VICTORI infrastructure is an exemplary Open Source 5G and IoT facility. This means that most of the installed components are offered as Open Source but there are also dedicated components and services to support 5G and IoT scenarios. Numerous partners have deployed their technologies in the Patras 5G /Greece facility, thus creating a unique 5G playground for KPI validation and support on future verticals. For more information please check (http://wiki.patras5g.eu/).

The Patras Facility uses currently OSM version FIVE with OSM FIVE VNFM support. OSM is aligned to NFV ISG information models while providing first-hand feedback based on its implementation experience.

The approach considers the following high-level requirements and ambitions:

- Adopt and being interoperable with current Cloud/SDN/NFV/MEC standards.
- Use open standards and integrate technologically mature, and widespread open source toolsets.
- Put effort to enable experimentation and make it effortless for experimenters to deploy experimentation scenarios.
- Being interoperable with standards and other 5G facilities.
- Adopt Open Source technologies that are embraced also by industry, like Open Source MANO.
- Consider the multi domain nature of the Facility.
- Scheduling of various Network Service orchestrations.
- Automation of all processes that pertain the 5GinFIRE operations.
- Consider the Long-term evolution of component. Vertical applications can access the Patras 5G Service Catalogue through the Patras Facility site portal: https://patras5g.eu. Vertical applications can self-manage and onboard their artifacts through our portal or access programmatically available services.

Various artifacts can be managed through the facility portal via standardized TMForum OpenAPIs: Service Catalog, Service Order and Service Inventory, Partner Management and Users, Service Orchestration, VNFs/NSDs catalogue, NFVO endpoints via OSM NBI, Service and NFV Deployment requests.

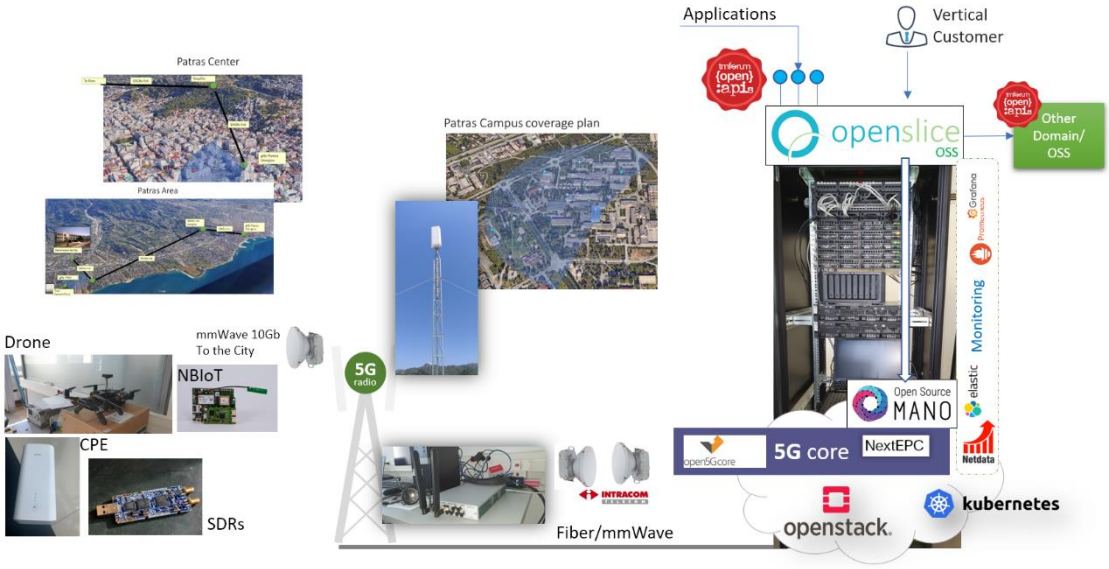

Figure 2-6 5GVINNI (Patras cluster) Summary of capabilities

2.3.4 5GENESIS

5GENESIS is another ICT-17-2018 project that is developing a multi-site facility. 5GENESIS defines a 5G experimentation blueprint [11], which will serve as a common architectural reference. This blueprint will also include an openness framework, with APIs for exposing the facility to verticals for experimentation. The Berlin facility from 5GENESIS will be part of the 5G-VICTORI Platform, supporting rail and media vertical applications.

The Management and Orchestration Layer of the 5GENESIS architecture [12], shown in Figure 2-7 in green, comprises three main components, namely:

- NFV MANO,
- Network Management System (NMS),
- Slice Manager.

5GENESIS uses OSM to realize the NFV MANO functionality. The VIM is provided by OpenStack, which is conformant to ETSI NFV specification.

The NMS is a platform-specific network management system with direct access to physical resources as well as configuration interfaces. In the Berlin platform, the NMS will provide the overview of the physical resources and an interface to manage them.

The Slice Manager is common to all 5GENESIS platforms and provides management service for network slicing for different domains by communicating with South Bound components. This provides the much-needed link between the orchestration and the Coordination Layer. Due to the modular architecture of Slice Manager, it provides flexibility and scalability in building and maintaining the applications. It exposes APIs that would be used by Experiment Lifecycle Manager (ELCM) [13] to perform CRUD operations on NSIs, NFVOs.

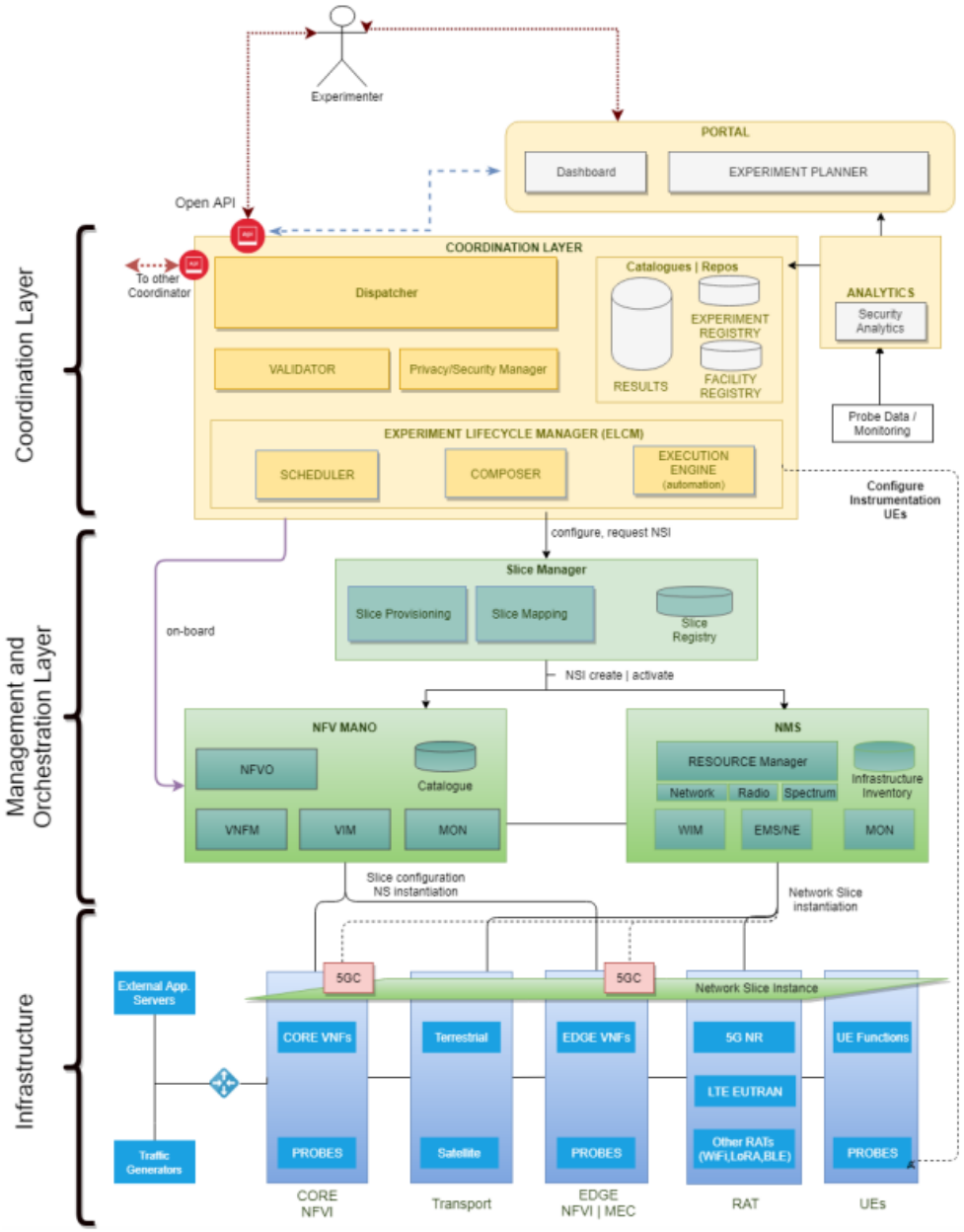

Figure 2-7 5GENESIS architecture overview with MANO components indicated with green

2.3.5 5G EVE

5G EVE is an ICT-17-2018 project, an E2E European 5G platform for validation of the extensive 5G vertical's UCs and trials, with the goal of implementing and testing advanced 5G infrastructures, interconnecting sites from France, Spain, Italy and Greece.

5G-EVE facility is enabling several experiments with (a) heterogeneous NR access, using licensed/unlicensed spectrum, (b) MEC, backhaul and core services and technologies, (c) site interworking for multi-site, multi-domain, multi-technology orchestration, in the first project phase 3GPP Rel. 15 compliant and second project phase 3GPP Rel. 16 compliant. 5G-EVE creates and make available a technical open framework using open APIs – enabling several verticals to access and use the project 5G facilities, by abstracting the infrastructure complexity. The projects also provides a methodology across the 5G E2E facilities for performing tests, KPIs evaluation, technology-benchmarking and performance evaluation and diagnosis, delivering also support for analysing the results and proposing performance improvements for the trials/experiments.

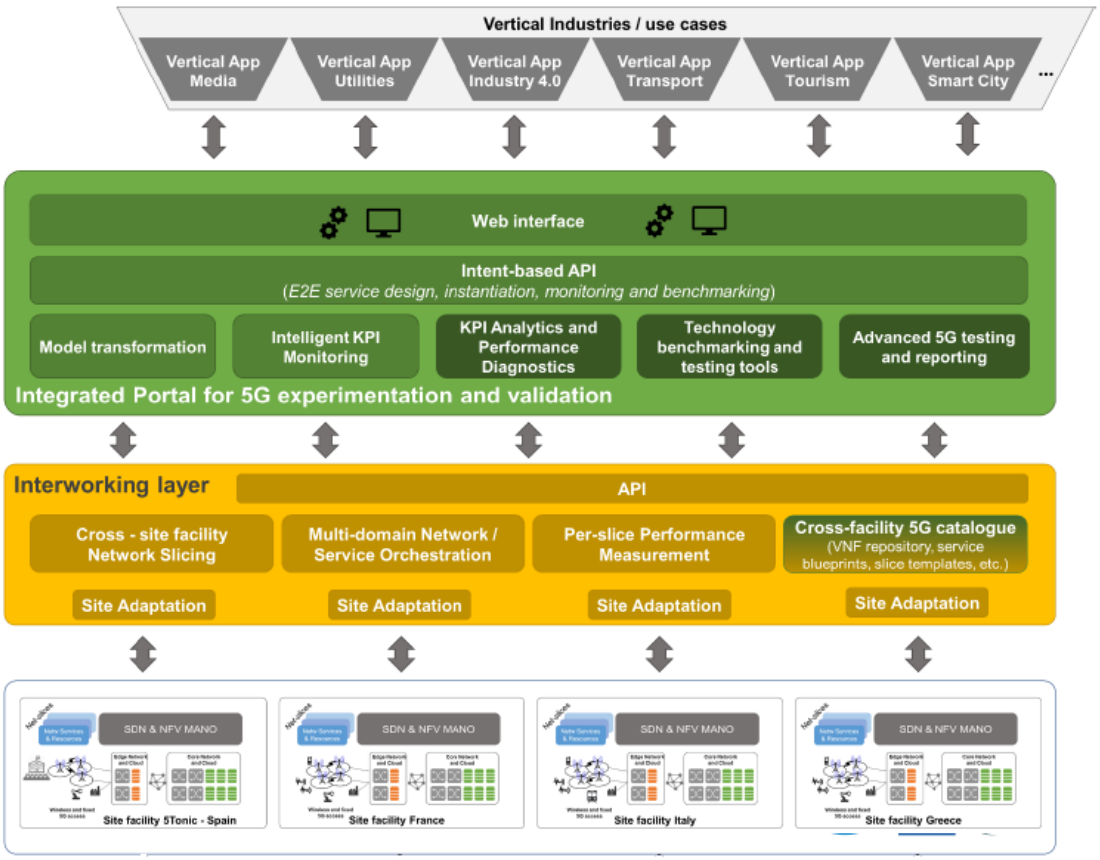

Figure 2-8 5G-EVE overall architecture

The overall 5G-EVE architecture shown in Figure 2-8 consists of: (1) vertical industries UCs that interface with (2) the portal, serving as a main point of contact with the 5G-EVE facility and comprises tools for testing, benchmarking, validation and experimentation, as verticals may use APIs to automate processes and (3) the 5G experimentation and validation platform that interacts with different underlying resources offered by the sites facility, as described in Figure 2-8.

The platform is able to create network slices, resource scheduling and resource allocation for experimentation, multi-x slice implementation, for both user plane and control plane, providing 5G service capabilities. The main 5G network components and capabilities are 5G RAN and Core, MEC, orchestration tools as open ones like ONAP, OSM, Juju or commercial like Ericsson Orchestrator, open source tools like OAI, OpenStack, OpenDayLight.

5G-EVE is vertical oriented open framework and is facilitating to the different vertical's industry, including the ICT-19 EU funded projects, including 5G-VICTORI, the 5G experimentation tools and experimentation and demonstration environment,

5G-VICTORI France/Romania (FR/RO) cluster is supported by ICT-17 5G-EVE cluster located in the France, Paris and Nice facilities, making use of OpenAirInterface (OAI) (5G software alliance), Mosaic-5G (first ecosystem of 5G R&D open source platforms ranging from the centralized network control to the mobile edge network deployment), ONAP and capable virtualized infrastructure. The 5G-EVE cluster, depicted in Figure 2-9, providing several 5G key concepts as network slicing capabilities, 5G RAN and Core implementations, service's resources allocation and isolation, network capacity to support simultaneous services and users.

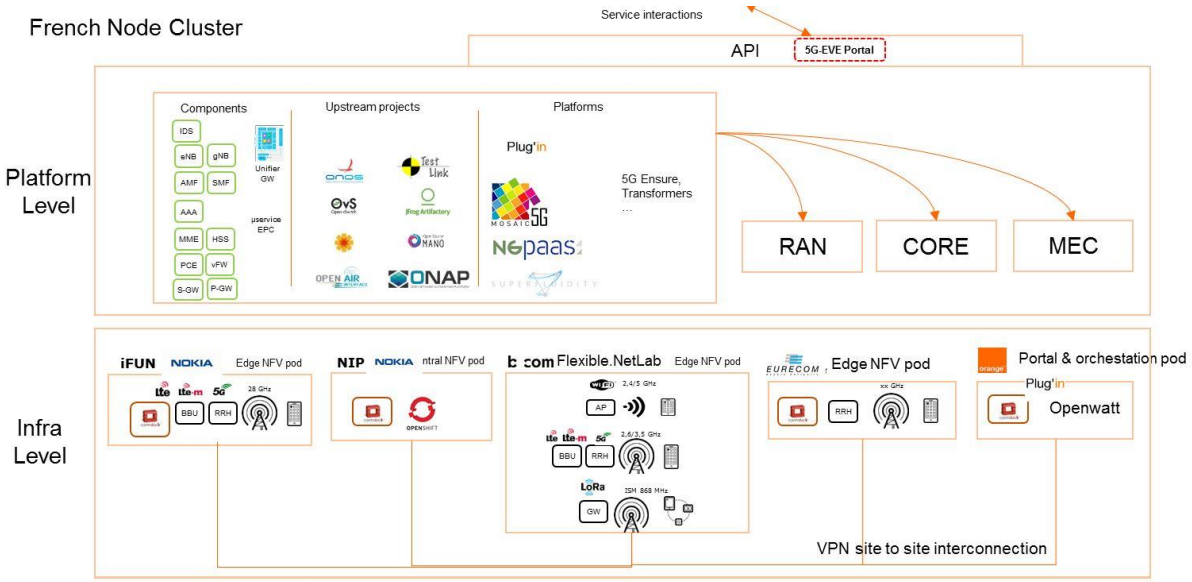

Figure 2-9 5G-EVE French cluster overview

The 5G-EVE cluster functionality is extended to the Romanian cluster implementation, following the already existing infrastructure and components deployed by Orange in the SliceNet and Matilda projects. Examples are the Orange IaaS testbed, integrating also the 5G-EVE tools and components in terms of service instantiation (web portal), orchestration (ONAP), OAI 5G network elements (RAN and Core), 5G infrastructure for experimentation and validation for Romanian's UCs with 5G-VICTORI components.

2.3.6 SONATA

SONATA has developed an NFV framework that provides a programming model and development toolchain for virtualized services, fully integrated with a DevOps-enabled service platform and orchestration system. The SONATA NFV platform is a flexible and integrated platform able to allow creation of a versatile and modular ecosystem that serves service developers and testers, telecom operators or vertical industries, managing the full lifecycle of network services. The SONATA NFV platform encompasses everything from development to network services testing and then their deployment and orchestration in a 5G infrastructure.

It is composed of three main modules: (a) a service development kit (SDK), (b) the service platform and (c) a validation and verification (V&V) platform.

- The SDK is a set of programming models, processes and tools oriented to assist software developers in the development and test of Virtual Network Functions (VNFs) and NSs. The SDK allows developers to define complex services consisting of multiple VNFs. Further the SDK provides a multi-PoP emulation platform for rapid prototyping of VNFs and services as well as primitive validation and profiling solutions.
- The SONATA Service Platform is a modular and highly customizable solution able to manage the full lifecycle of Network Functions (VNF, CNF, PNF, and HNF), NSs and Network Slices. It provides a MANO (Management and Orchestration) framework (Figure 2-10) able to orchestrate the network resources in an efficient way. Due to the fully customisable and modular design of its MANO framework (Figure 2-11), the service platform offers customisation opportunities on two levels. First, the service platform operator can modify the platform, e.g., to support a desired business model, by replacing components of the loosely coupled MANO framework (MANO plugins). Second, service developers can in influence the orchestration and management functionalities of the platform pertaining to their own services, by bundling small management programs, so-called function- and service-specific managers (FSMs/SSMs), with their services.

- The V&V Platform performs automatic verification and validation of services and manages the test results to form a continuous improvement feedback loop. It allows the qualification of services across multiple different orchestration platforms, testing both the functional and non-functional aspects of their deployments. This process is fully automated to ensure that can take place with little or no human intervention.

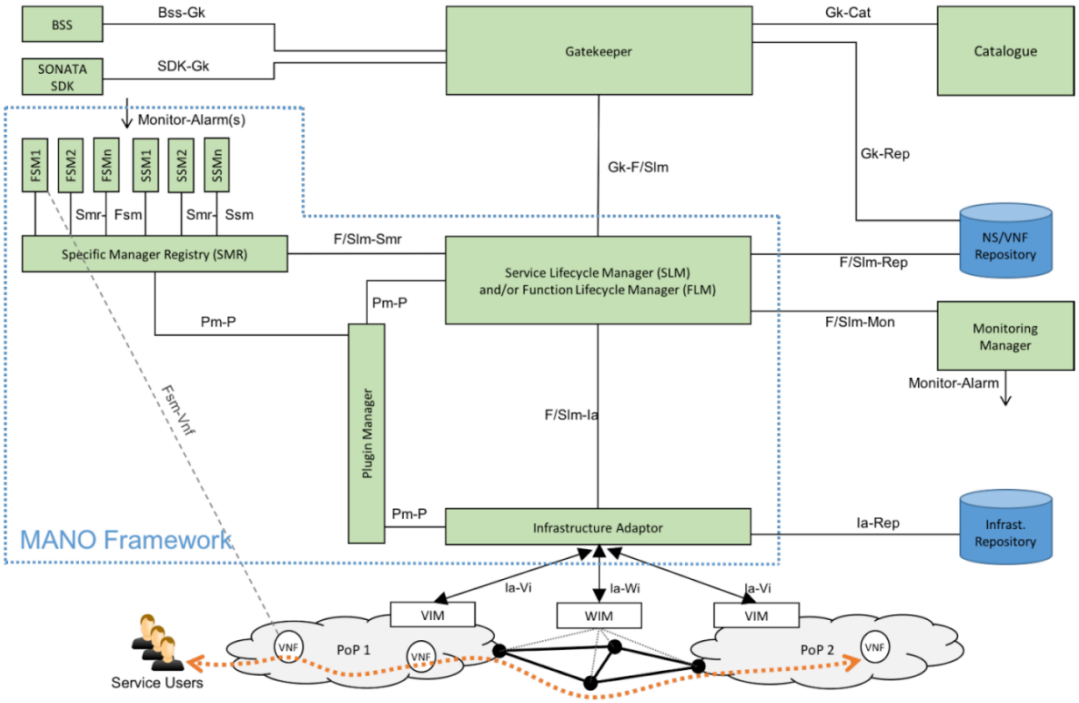

Figure 2-10 Logical view on the SONATA service platform architecture and its interfaces

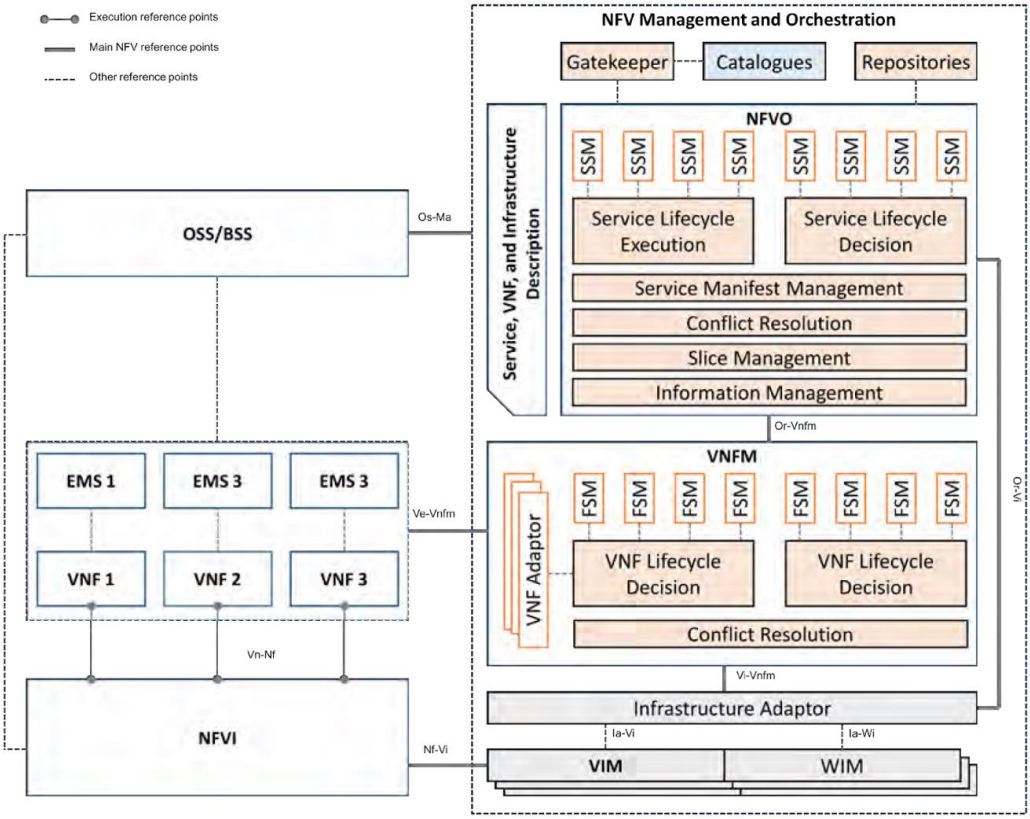

Figure 2-11 SONATA's high-level architecture mapped to the ETSI NFV reference architecture

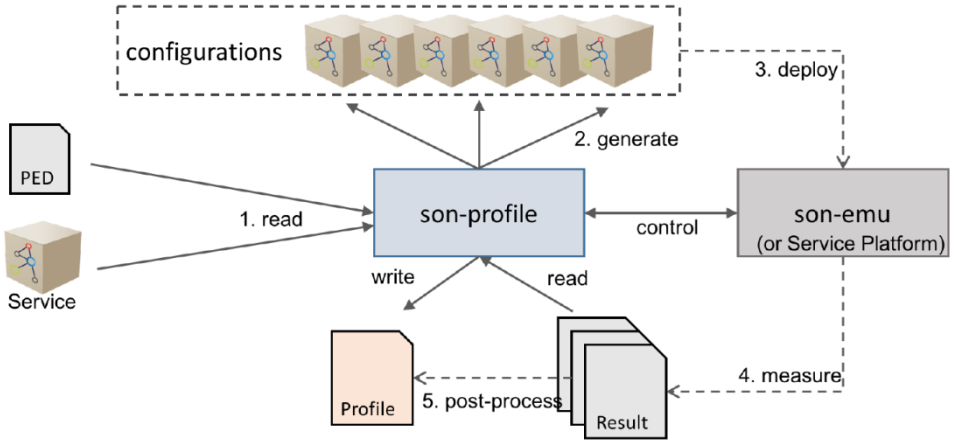

Figure 2-12 SONATA's profiling toolchain

There are two types of profiling approaches that can be applied to NFV UCs. The first one utilizes cloud testbeds to execute VNFs in realistic environments under different resource configurations. To do so, a VNF is executed as a VM with a predefined resource configuration and its performance is measured. After this, the VM is destroyed and a new one with another resource configuration is started. Based on this, performance values for different resource configurations can be measured, which creates a mapping from available resources to resulting performance profile. The second approach executes a single VNF and sends varying amounts of workload to it. During this, its resource consumption, like CPU and memory, is measured so that the results reflect a mapping from workload to resource usage. SONATA follows a hybrid approach and allows developers to specify both the resource configurations to be tested during a profiling run as well as the used traffic generators and their parameters. It utilizes the emulation platform with its container-based VNF execution that allows to control the resource configurations, such as CPU cores, available CPU time, memory and block I/O, in a very fine-grained way for each of the VNFs in a profiled service. SONATA profiling toolchain is provided as a part of the SDK with the main component called son-profile and illustrated in Figure 2-12. ETSI recommendations and methods for pre-deployment testing of the functional components of a NFV environment can be found in [14]. The modular implementation of SONATA is followed also in the profiling functionality. This enables not only the deployment but also validation of the performance, reliability and scaling capabilities of NSs.

2.3.7 5GTANGO

The 5GTANGO project is based on the concepts and developments of SONATA, enabling the flexible programmability of 5G networks with: a) an NFV-enabled SDK; b) a Store platform with advanced validation and verification mechanisms for VNFs/Network Services qualification (including 3rd party contributions); and, c) a modular Service Platform with an innovative orchestrator in order to bridge the gap between business needs and network operational management systems. The 5GTANGO architecture is based on a number of design principles including simplicity, loose coupling allowing a modular development based on micro-services and support of multiple service platforms.

The 5GTANGO's platform is split into three large subsystems, which are modelled according to three phases. These phases represent stages in the lifecycle of a service: development, verification & validation, and operation. Although these phases may overlap, they typically happen at different timescales and are performed by different actors:

1. Phase 1: Development of a NS, and publishing of that network service in a catalogue.
2. Phase 2: Testing, validation, and verification of a NS with a V&V platform, and publishing of the results in a catalogue.
3. Phase 3: Selecting a NS from a catalogue, deployment and operation of that NS using the service platform orchestration capabilities such as placement, monitoring, scaling, etc.

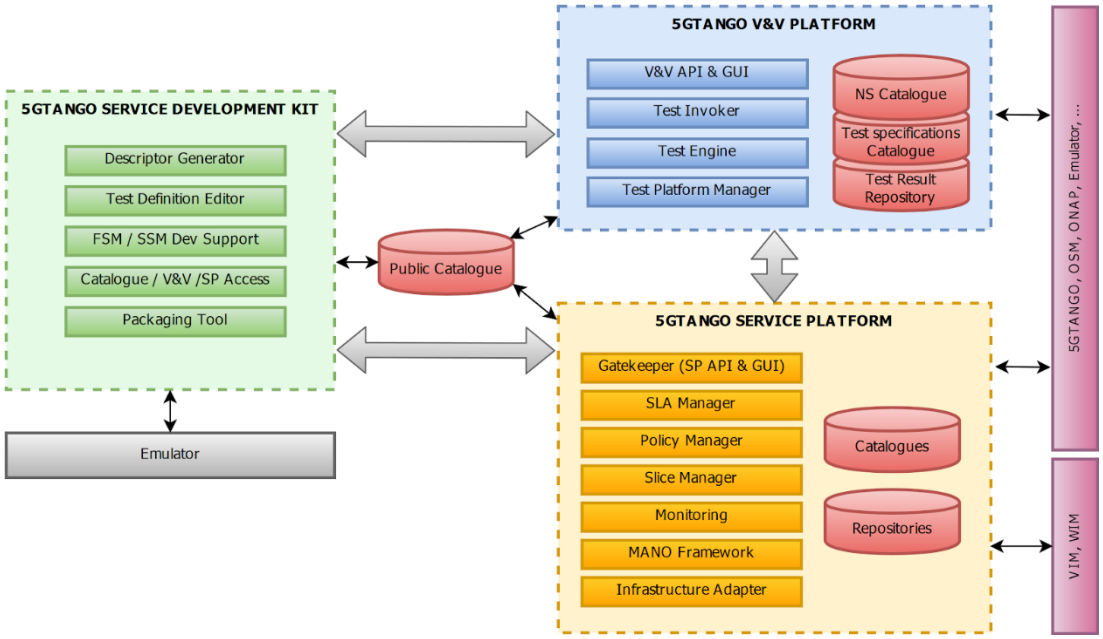

Figure 2-13 Overall 5GTANGO Architecture

2.3.8 5G-TRANSFORMER

Figure 2-14 presents the 5G-TRANSFORMER system architecture constituting three major building blocks, namely: vertical slicer (5GT-VS), service orchestrator (5GT-SO) and mobile transport and computing platform (5GT-MTP). The 5GT-VS is the entry point for a vertical requesting a service and it handles the association of these services with slices as well as network slice management. The 5GT-SO is responsible for E2E orchestration of services across multiple domains and for aggregating local and federated (i.e., from peer domains) resources and services and exposing them to the 5GT-VS in a unified way. Finally, the 5GT-MTP provides and manages the virtual and physical compute and network resources on which service components are eventually deployed. It also decides on the abstraction level offered to the 5GT-SO.

2.3.8.1 Vertical Slicer (5GT-VS)

The 5GT-VS is the common entry point for all verticals into the 5G-TRANSFORMER system. It is part of the operating and business support systems (OSS/BSS). Vertical services are offered through a high-level interface at the 5GT-VS northbound, which is designed to allow verticals to focus on the service logic and requirements, without caring about how their services are eventually deployed at the resource level. The 5GT-VS offers a catalogue of vertical service blueprints (VSB), based on which the vertical service requests are generated by the vertical. The role of the 5GT-VS is to trigger the actions allowing the 5G-TRANSFORMER system to fulfil the requirements of a given incoming service request. After appropriate translation between service requirements and slice-related requirements by the VSD/NSD Translator, a decision is taken on whether the service can be provided through an already existing slice or a new one needs to be created. The vertical slicer is the component of the system that is aware of the business needs of the vertical, their SLA requirements, and how they are satisfied by mapping them to given slices.

The 5GT-VS maps vertical service descriptions and instantiation parameters at the vertical application (VA) level into an NFV-NS (existing or new) implementing the network slice. In turn, such NFV-NS will be updated or created through a network service descriptor (NSD), which is a service graph composed of a set of VNFs chained with each other, and the corresponding fine-grained instantiation parameters (e.g., deployment flavour) that are sent to the 5GT-SO.

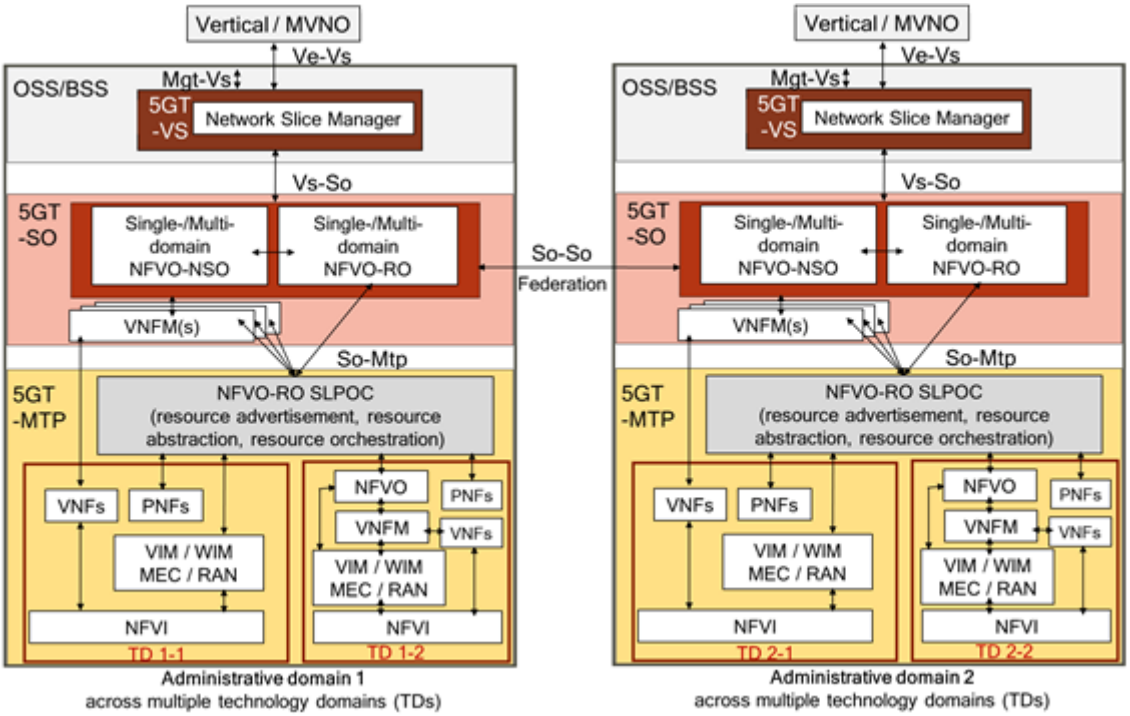

Figure 2-14 5G-TRANSFORMER System Architecture

2.3.8.2 Service Orchestrator (5GT-SO)

The NFV-NS from the 5GT-VS is received by the 5GT-SO through the Vs-So interface. The main duty of the 5GT-SO is to provide end-to-end orchestration of the NFV-NS across multiple administrative domains by interacting with the local 5GT-MTP (So-Mtp reference point) and with the 5GT-SOs of other administrative domains (So-So reference point). If needed (e.g., in case of not enough local resources), the 5GT-SO interacts with peer 5GT-SOs located in other administrative domains (federation) to take decisions on the E2E (de)composition of virtual services and their most suitable execution environment. Even if a service is deployed across several administrative domains, e.g., if roaming is required, a vertical still uses one single 5GT-VS to access the system. The 5GT-SO hides this federation from the 5GT-VS and thus from the verticals. The 5GT-SO embeds the network service orchestrator (NFVO-NSO) and the resource orchestrator (NFVO-RO) with functionalities equivalent to those of a regular NFV orchestrator and it may be used for single and multi-domains. The 5GT-SO also includes the Virtual Network Function Managers (VNFM) to manage the lifecycle of the VNFs composing the NFV-NS. ETSI GS NFV-IFA 006-based interfaces are used to allow the VNFM interacting with the NFVO-RO Single Logical Point of Contact (SLPOC) entity in the 5GT-MTP, as well as peer 5GT-SOs for resource allocation requests involving the VNFs under its control.

2.3.8.3 Mobile Transport and Computing Platform (5GT-MTP)

The 5GT-MTP is responsible for orchestration of resources and the instantiation of VNFs over the infrastructure under its control, as well as managing the underlying physical mobile transport network, computing and storage infrastructure. There are multiple technology domains (TD) inside a 5GT-MTP – e.g., data centres, mobile network, wide area network (WAN). The 5GT-MTP NFVO-RO acts as E2E resource orchestrator across the various technology domains of the 5GT-MTP. The computing and storage infrastructure may be deployed in central data centres as well as distributed ones placed closer to the network edge, as in MEC. Therefore, the 5GT-MTP is in charge of managing the virtual resources on which the NFV-NSs are deployed. The NFVO-RO acts as single entry point, i.e., single logical point of contact (SLPOC) in ETSI GS NFV-IFA 028 terminology, for any resource allocation request coming from the 5GT-SO. The So-Mtp interface is based on ETSI GS NFV-IFA 005 and ETSI GS NFV-IFA 006. The former allows the NFVO-RO of the 5GT-SO to request resource allocations to the NFVO-RO of the 5GT-MTP, whilst the latter allows the VNFM of the 5GT-SO to request resource allocations for the VNFs under its control. In terms of managing VNF instances, the So-Mtp interface also consists of ETSI GS

NFV-IFA 008-based interfaces (i.e., the Ve-Vnfm-vnf reference point) to allow the VNFM of the 5GT-SO to directly configure the VNF instances running in the 5GT-MTP. The 5GT-MTP NFVO-RO has full visibility of the resources under the control of the VIM managing each technology domain with ETSI GS NFV-IFA 005-based interfaces deployed between the 5GT-MTP NFVO-RO and the 5GT-MTP VIMs. Therefore, when receiving a resource allocation request from the 5GT-SO, the 5GT-MTP NFVO-RO generates the corresponding request to the relevant entities – e.g., VIM or WAN Infrastructure Manager (WIM) – each of them providing part of the virtual resources needed to deploy the VNFs, interconnect them, and/or configure the relevant parameters of the PNFs that form the NFV-NS.

2.3.9 MATILDA

MATILDA is a holistic and innovative 5G framework for the lifecycle of design, development, and orchestration of 5G-ready applications and 5G network services over programmable infrastructure. In MATILDA, 5G-ready applications are based on cloud-native/microservice development principles and preserve the separation of concerns among the orchestration of the developed applications and the required network services that support them. However, since the 5G ecosystem introduces several challenges that are not addressed by cloud-native applications and their orchestration means (e.g. **zero-delay tolerance** and **excessive mobility**), MATILDA introduces a specification for the management of network and computing slices (developed in partnership with the SliceNet project) that are application-aware and can lead to optimal application execution.

Using the framework, software developers can create vertical applications with simple and conventional microservices – where each component is independently orchestratable. Based on the definition of metamodels (application component and graph metamodels), metadata information and are attached to these microservices (in the form of descriptors) and exploited during the deployment and operation to drive the (re)configuration of the programmable infrastructure. MATILDA uses VNFs and NSs to materialize the network slices and based on the defined intents and policies is able to ensure vertical applications can benefit from the 5G infrastructure to its full potential. The framework also encourages new business and wide collaboration by providing a Marketplace where application, components, VNFs and NSs can be published.

To fulfil its vision MATILDA reference architecture is divided in three distinct layers: a Development Environment (DE), a Vertical Application Orchestration Layer (VAO) and a Telecom Layer Platform (TLP), as presented in Figure 2-15.

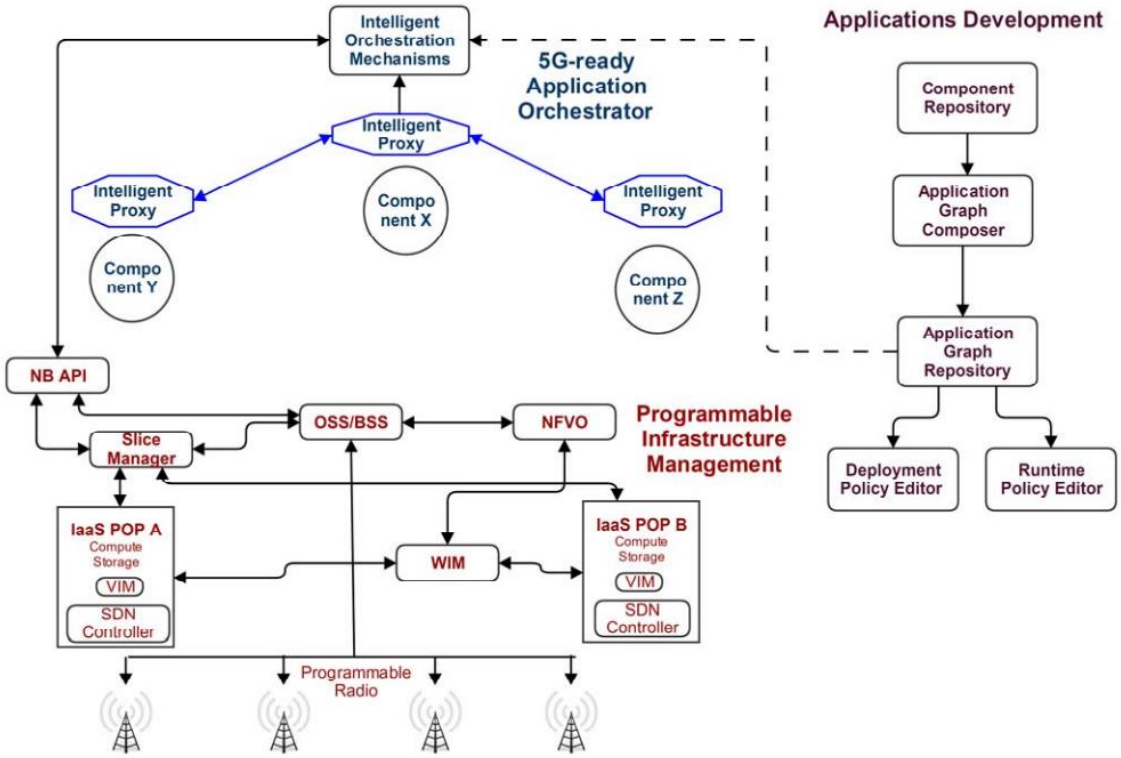

Figure 2-15 MATILDA high level architecture. The top left of the figure (in blue) represents the VAO, the bottom (in red) represents the TLP and the right side (in purple) represents the DE.

The DE is used offline to assist programmers and engineers in the creation new applications or pieces of the virtualized infrastructure. Using the DE tools, components can be packaged and distributed via Repositories (and Marketplaces) or integrated into an application graph using the graph composer (which in turn can also be shared). Still in the same environment it is possible to annotate the generated graph models with policies and intents, prior to the deployment, making them 5G-enabled.

The VAO is responsible for deploying and managing the overall orchestration of 5G-ready applications over network slices realized by the TLP. Inside the VAO, a set of intelligent orchestration mechanisms tackle optimal deployment, runtime policies enforcement, provision of control and data plane functionalities, real time monitoring and big data analysis aspects. The application graph is materialized by a service mesh, where each cloud-native component is interacting with other components through a proxy. The proxies create a data plane and undertake several tasks, such as dynamic service discovery, load balancing, TLS termination, circuit breaking, health checking, traffic shaping, publication of metrics, etc. The VAO includes an execution manager, stream aggregators and data fusion mechanisms, an analytics module and policy and optimization engines.

The TLP is responsible for providing a pool of coordinated (and possibly distributed) network and computational resources managed by a telecom operator based on the deployment and runtime requirements provided by the VAO (forming a layered orchestration and management hierarchy). As indicated by Figure 2-16. The main interface between the VAO and the TLP is an extended OSS. This OSS provides a northbound interface that is consumed by the VAO and the telecom provider, while leveraging southbound interfaces to the different NFV MANO architectural components to provide 5G capabilities of the network to the application graph via the deployment of network services.

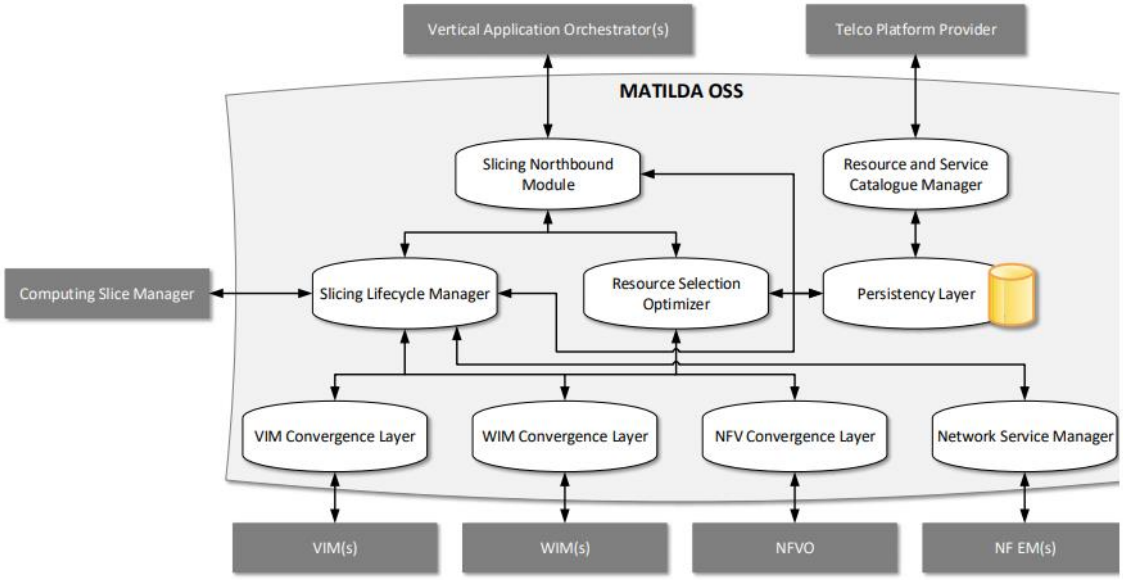

Figure 2-16: Architecture and main building blocks of the MATILDA extended OSS within their reference points towards external components

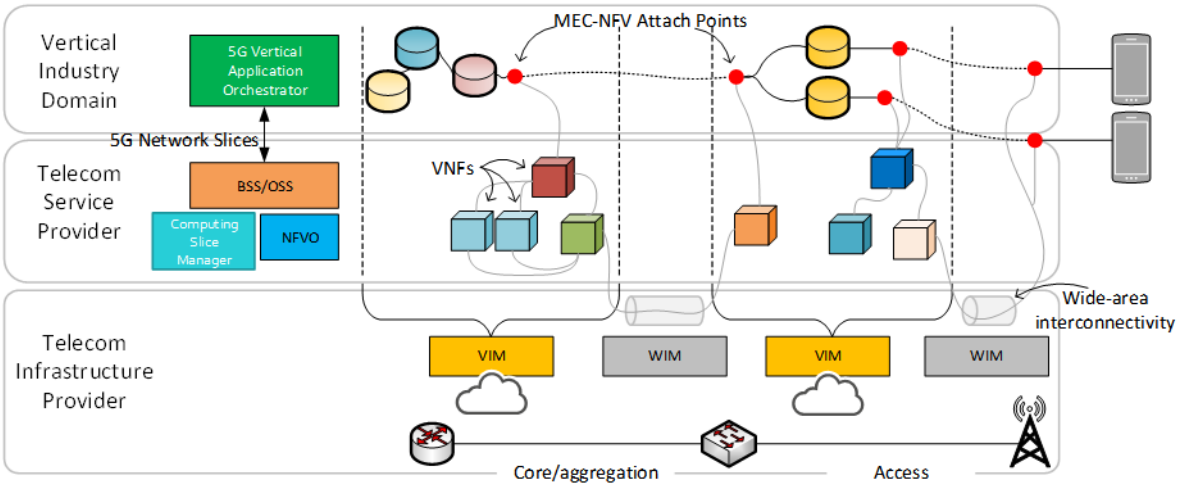

Figure 2-17: Key architectural building blocks and main stakeholders involved in the deployment of a vertical application onto a 5G infrastructure.

Due to this comprehensive architecture, MATILDA positions itself as an E2E framework for 5G and focus on delivering tools that can be used by network providers to support applications from vertical industries to explore the 5G infrastructure and create new, innovative, service offerings.

Monitoring in MATILDA is based on the collection of time series data in a Prometheus monitoring engine. Such data is made available through the UI provided by Prometheus, while custom reporting is also made available through Grafana. In both cases, the objective is to provide a live view of the basic resource usage and application-specific metrics to system administrators.

Moving one step further, profiling mechanisms have been developed aiming at supporting the realization of analysis over the collected time series data. The MATILDA Profiler is implemented based on the adoption of the OpenCPU framework that permits the detaching of the design and implementation of an analysis process from the execution of an analysis over selected time series data. This way, data scientists can easily design and implement the envisaged analysis scripts and onboard them into the Profiler. Based on the supported set of analysis scripts, the end user is able to select an algorithm, the set of monitoring metrics to be included in the analysis and the start and end time of the analysis.

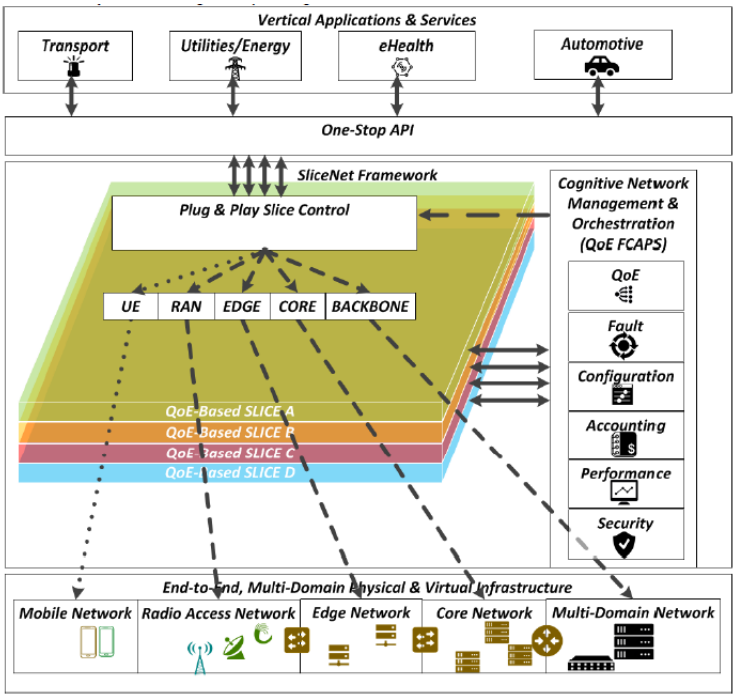

Figure 2-18 SliceNet Overall Architecture

2.3.10 SliceNet

SliceNet is the ICT-07-2017 project that is creating and demonstrating the tools and mechanism for achieving the ambition of creating E2E cognitive network slicing and slice management framework in virtualized multi-domain and multi-tenant networks. SliceNet project is designing, prototyping and demonstrating an innovative, verticals oriented and Quality of Experience (QoE) driven 5G network slicing framework, using different control and orchestration tools for provisioning and network operation. SliceNet is enabling for 5G verticals industries an innovative one-stop shop solution for demanding service requirements and for 5G service providers is providing guaranteed service quality by agile QoE-optimization of service creation and service delivery, presenting also an integrated FCAPS framework for E2E management, control and orchestration of network slices.

SliceNet project is enabling the innovative FCAPS management across multiple planes and operator domains, as fault management will detect and resolve slices' operational problem, configuration management will coordinate changes in the operation of slices, accounting management will optimise the resource distribution for slices, performance management will assure the overall performance of the whole system running numerous slices, in collaboration with the QoE management for individual slices. SliceNet is enabling also the slicing-friendly infrastructure, provisioning and control of user-definable slices, enabling fully compliant ETSI MEC paradigm for challenging requirements and network slicing across multiple administrative domains for different vertical's UCs. SliceNet is demonstrating the efficiency and the support for delivering slice-based network services for three different UCs, (1) 5G eHealth UC, (2) 5G Smart Grid UC and (3) Smart City UC. The 3rd UC is dealing with a Smart Lighting IoT solution implemented and demonstrated by Orange Romania, the infrastructure developed and deployed in SliceNet being one of the key pillar in terms of virtualization capabilities that will be further exploit and use for 5G-VICTORI UCs demonstration and experimentation.

2.3.11 5GCity

The 5GCity project focuses on the design, development and deployment of a distributed cloud and radio architecture for municipalities and infrastructure owners acting as 5G Neutral Hosts. In this way, it enables the realization of the Smart Cities vision, in which digital services are offered to citizens and Vertical industries leveraging on various 5G technologies. The approach followed in developing the 5GCity architecture was based on a three tier topology, as illustrated in Figure 2-19, in which tiers

correspond to different geographical areas of the city where the actual infrastructure elements for digital services are deployed.

Driven by the goal of empowering the city infrastructure and transforming it into a hyper-connected, distributed 5G-enabled edge infrastructure, 5GCity developed the Neutral Host Slicing and Orchestration Platform that is depicted in Figure 2-20 with a demonstration in three different cities (Barcelona, Bristol and Lucca) to validate the neutral host model in dense deployment environments such as cities. In essence, this platform allows tenants to develop slices using a set of virtualized network and computing resources in RAN, edge/far-edge and core network to support rapid, dynamic and customizable deployment of virtualized network services. In particular, the Orchestration & Control Layer acts as the logical core of the 5GCity platform [15] and it is composed by multiple functional blocks for control, management, and orchestration across its three-tier architecture.

As one of the central components of the platform, the Slice Manager provides the required logic for the dynamic creation and management of slices, which are defined as a collection of compute, network and radio chunks combined together with the orchestration of network services deployed on top of them [16]. Apart of managing the registration of infrastructure resources and the life-cycle of slices, the Slice Manager also performs several actions to seamlessly automate operational tasks related to slice provisioning and day-1 (i.e. configuration) and day-2 (i.e. optimization) operations over deployed virtual functions.

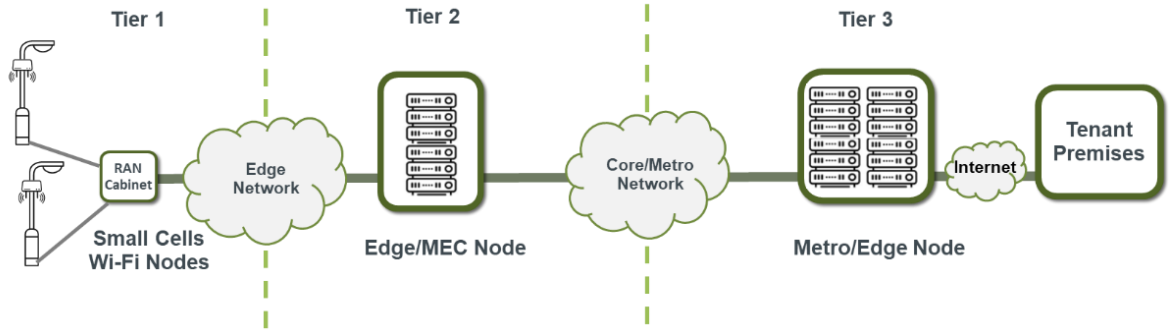

Figure 2-19 5GCity Three-Tier Topological Architecture

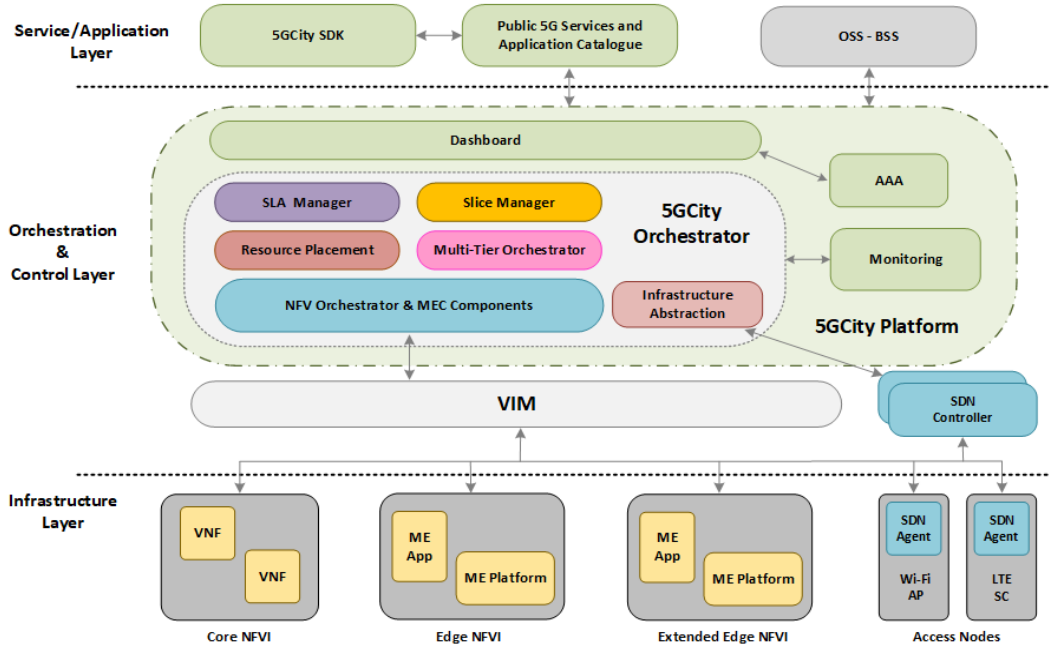

Figure 2-20 5GCity Neutral Host Slicing and Orchestration Platform

In addition to integrating the Cloud and Edge computing paradigms into a single E2E infrastructure, this platform also enables virtual RAN slicing and RAN function virtualization based on the use of SDN-based RAN controllers to manage the radio components and enforce many of the required actions. In overall, the resulting slicing and orchestration platform allows infrastructure operators and Vertical actors to virtualize and slice infrastructure resources made available in the cities, enabling them to be logically segmented, configured and reused in order to meet multi-tenant demands.

2.3.12 OSM

Open Source MANO (OSM) is an ETSI-hosted open source project and community aiming to develop an open-source NFV Management and Orchestration (MANO) software stack for telco services, aligned with ETSI NFV information models. It constitutes a layered approach to create composite services of growing complexity and offers a production-quality MANO stack that meets operators' requirements for commercial NFV deployments. It is capable of consuming openly published information models, available to everyone, suitable for a broad ecosystem of virtual network functions (VNF) vendors, operationally significant and VIM-independent. The architectural principles it relies on are layering, modularity, abstraction and simplicity. It is suitable for both greenfield and brownfield deployments and is built on top of existing popular, well tested technologies (e.g. Cloud Management, SDN Controllers, Visualization and Reporting Frameworks, etc.).

OSM's approach aims to minimize integration efforts thanks to four key aspects:

1. A well-known **Information Model (IM),** aligned with ETSI NFV, which is capable of modeling and automating the full lifecycle of Network Functions (virtual, physical or hybrid), Network Services (NS), and Network Slices (NSI), from their initial deployment to their daily operation and monitoring. OSM's IM uses MANO descriptor files for VNFs, network services and network slices, which are configuration templates that define the main properties of managed objects in a network in a standard way. This enables the formation of a unified VNF catalogue for maintainability, reusability and automation. Specifically, a Virtual Network Function Descriptor (VNFD) is a template for a VNF description regarding its deployment, operation, connectivity and virtualized resource requirements. A Network Service Descriptor (NSD) contains the complete information about the Network Service (NS), i.e., the set of network functions and the connection between them. It refers to VNF or PNF Descriptors (VNFDs/PNFDs) for VNFs/PNFs that are part of this NS, or to other NSDs in case of a nested NS. Finally, a Network Slice Template (NST) contains the complete information about a network slice instance, i.e. the NSs and the connections between them. Actually, OSM's IM is completely infrastructure-agnostic, so that the same model can be used to instantiate a given element (e.g. VNF) in a large variety of VIM types and transport technologies, enabling an ecosystem of VNF models ready for their deployment everywhere.
2. OSM provides a **unified northbound interface (NBI),** which enables the full operation of the system and the Network Services and Network Slices under its control. In fact, OSM's NBI offers the service of managing the lifecycle of NSs and Network Slices Instances (NSI), providing as a service all the necessary abstractions to allow the complete control, operation and supervision of the NS/NSI lifecycle by client systems, avoiding the exposure of unnecessary details of its constituent elements.
3. The extended concept of **"Network Service"** in OSM, so that an NS can span across the different domains identified —virtual, physical and transport—, and therefore control the full lifecycle of an NS interacting with VNFs, PNFs and HNFs in a unified manner along with on demand transport connections among different sites.
4. In addition, OSM can also manage the **lifecycle of Network Slices**, assuming if required the role of Slice Manager, extending it also to support an integrated operation.

The management and configuration actions that can be performed with OSM include examples such as health/performance monitoring & alarms, auto-scaling, fault-management, reaction to custom

events, or Virtual Deployment Unit (VDU)-level actions such as applications installation and lifecycle, leadership election and metrics collection. The available actions can be categorized as follows:

- **Day-0 configurations**: This is the basic instantiation phase. It concerns the pre-initialization configurations for VNFs and the management access establishment, e.g. create user, set hostnames, passwords, network configuration.
- **Day-1 configurations**: This is the service initialization phase. It concerns the initial configuration primitives invoked during instantiation time. Examples include packages installation, configuration files edit and commands execution.
- **Day-2 configurations**: This is the runtime operations phase. It concerns the configuration primitives invoked during runtime at Operator's demand, for behaviour modification and services monitoring e.g. on-demand actions for logs, routes installation, backups creation and restoration.

Day-1 and Day-2 configurations are implemented through Charms, which consist of actions and hooks modeled and grouped into layers, for repeatable and reliable management. Actions are parameterizable programs containing logic to install, configure, and scale, whereas hooks are events/signals that trigger actions.

The high-level architecture of OSM is depicted in Figure 2-21 and consists of the following components:

- **MON**: A monitoring module providing the mechanisms to leverage external monitoring tools. Monitoring information, sent to the unified message bus, can be consumed by fault and performance management solutions.
- **POL**: Policy Manager for OSM.
- **RO**: A resource orchestrator interfacing with the infrastructure layer. It coordinates the allocation and configuration of computing, storage and network resources, while supporting and managing different Virtualized Infrastructure Managers (VIMs) and Software Defined Networking (SDN) controllers.
- **VCA:** A VNF configuration and abstraction component. It is aligned with the VNF Manager defined by ETSI (day-1 and day-2 configuration of VNFs) and responsible for the execution of Juju charms specified within VNF packages.
- **LCM:** A lifecycle management component, supporting the lifecycle management of network services. It enables functionalities such as management of VNF/NS descriptors and packages. It interfaces with the RO for resource orchestration and the VCA modules for configuration.
- **NBI**: A unified northbound interface which enables full operation of the system. It is a restful server aligned with ETSI NFV standard SOL005.
- **light-ui**: A Web user interface, communicating with NBI.

Each component/micro-service is deployed as a container and all communicate asynchronously via a common Kafka message bus. The Information Models are stored in a common MongoDB database.

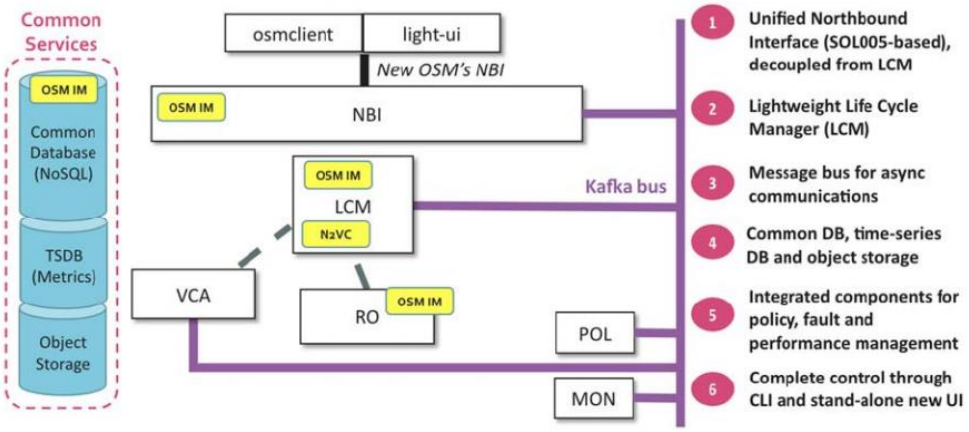

Figure 2-21 OSM high level architecture

2.3.13 ONAP

Open Network Automation Platform (ONAP) is an open source project hosted by Linux Foundation[2], officially launched in 2017, as an open source platform enabling telco networks to move to become more and more autonomous. ONAP is the platform capable to provide real time, policy-driven service orchestration and automation, enabling telco operators and application developers to instantiate and configure network functions. ONAP is the platform addressing also future 5G challenges, covering multi-site and multi-vendor automation capabilities, service and resources deployment, providing cloud network elements and services instantiation in a dynamic, real time and closed-loop for several majors telco activities as design, deployment and operating services, within two main ONAP framework, Design-time and Run-time.

ONAP intends to be the common automation platform for services management, vendor agnostic, able for policy-driven service design and analytics, providing orchestration and service configuration capabilities for virtual and network functions, enhanced with a number of features from release to release:

Table 2-10 ONAP version releases

Release name	Release version	Release date
El Alto	5.0.1	October 2019
Dublin	4.0.0	July 2019
Casablanca	3.0.2 3.0.1 3.0.0	January 2019 November 2018 April 2019
Beijing	2.0.0	June 2018
Amsterdam	1.0.0	November 2017

[2] Open Network Automation Platform , https://docs.onap.org/en/elalto/index.html#

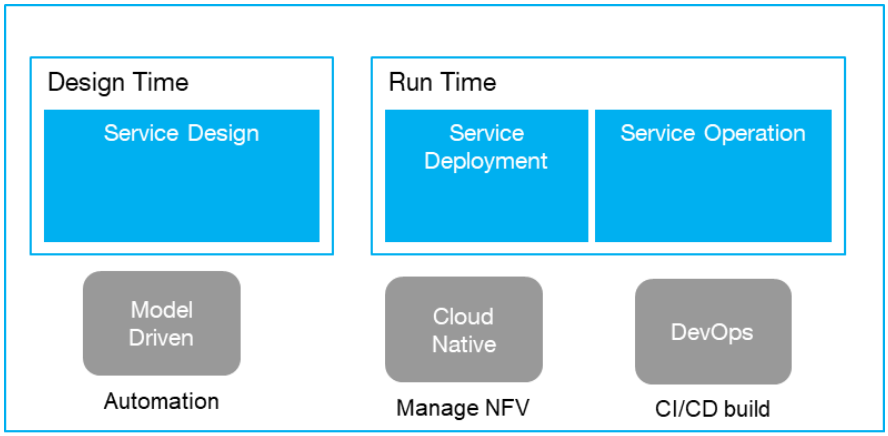

Figure 2-22 ONAP activities and principles

ONAP is bringing the DevOPS practices and the agile deployment methods for telco operators, with main focus on (1) Automation of VNFs and service lifecycle management and (2) automation of management operation and service assurance.

Service design activities framework is modelling the resources and is defining the service policies rules, using TOSCA approach for VNFs planning on-boarding, resource creation and service composition and service distribution.

Service deployment and orchestration provides automated instantiation and service management, defining the VNFs used for the service, orchestration steps, cloud zone selection and the service orchestration.

Service Operation monitors the service behavior and through the analytics control framework provides resource scaling and healing, by collecting and evaluating monitored events data and closed loop design and deployment.

ONAP platform allows rapid and dynamic network function and services instantiation, enabling several features supported by ONAP architecture, described in Figure 2-23[3] :

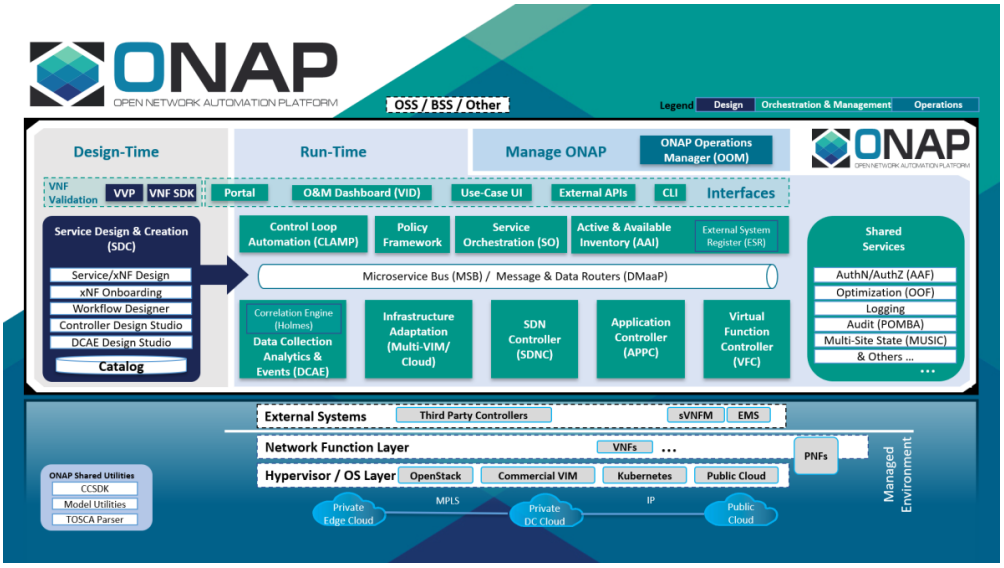

Figure 2-23 ONAP high level architecture

[3] ONAP architecture with its microservices-based platform components, https://docs.onap.org/en/elalto/guides/onap-developer/architecture/onap-architecture.html

ONAP functional architecture is highlighted by several key components for design and onboarding services, APIs north-bound interoperability, information model and framework utilities, workflows and policy models covering a number of SDOs as ETSI NFV MANO, IETF, MEF, OASIS TOSCA.

Latest ONAP release supports several models, as we highlight several of them further:

- VNF Descriptor Information Model based on ETSI NFV IFA011
- PNF Descriptor Information Model based on ETSI NFV IFA014
- PNF Descriptor TOSCA based Data Model based on IM and ETSI NFV SOL001
- VNF Package format leveraging the ETSI NFV SOL004 specification
- Network Service Descriptor (NSD) realized by the VFC

ONAP main key components, modules and functions description:

- **ONAP portal** is the single user design and runtime environment, based on the users roles managed by the portal, ONAP is providing access to design, analytics and operational functions, including an SDK to enable multiple development access and a CLI required by operators.
- **ONAP Design Framework** is the comprehensive development environment tools, techniques and repositories for describing resources, services and products and their management and control functions.
- **ONAP Runtime Framework** is the environment that executed rules and policies, distribution of the models of the design and creation environment and policies among different ONAP modules.
- **ONAP Service Orchestration** is the component that executes the specific process of automating activities, tasks, rules and policies needed for services on-demand life cycle management, providing E2E orchestration for infrastructure, network and applications, including VNFs, CNFs and PNFs.
- **ONAP Multi-Cloud Adaptation** is the infrastructure adaptation layer for VIMs/Clouds capabilities, exposing hardware platform awareness.
- **ONAP Optimization Framework** is providing the policy and model driven framework for applications and services optimization, enabling the service placement on multi-site multi-cloud (ONAP multi-VIM/Cloud) environment with respect of a variety of policies as location, platform capabilities, capacity and resource availability.
- **ONAP Active and Available Inventories**(A&AI) is the module providing real-time views of the system's resources, services and products, acting not only as a registry but also is maintaining un updated view of the inventories items, as it is metadata driven allowing also new inventories to be added dynamically.
- **ONAP Closed Loop Automation**(CLAMP) is the design-time and run-time phase collaboration, Data collectors from DCAE, orchestrators implementing CLAMP actions, providing also FCAPS functionality.

As a conclusion, ONAP is the open tool platform that provides real-time, policy-driven orchestration and automation for virtual (VMs or container based network service function) and physical network functions, enabling IT software approach for telco cloud environments, enabling cloud-native application developments and CI/CD for operators. It is requested that various edge cloud architecture (MEC) to be plugged into ONAP architecture for service orchestration, as various edge clouds may impact ONAP components in terms of data collection, processing, policy and resources management and CLAMP.

ONAP NBI architecture for El Alto release brings to ONAP a set of API that can be used by external systems as BSS/OSS as interaction points such as ONAP to Partner's Orchestration(NBI) and exposure of TMF open APIs

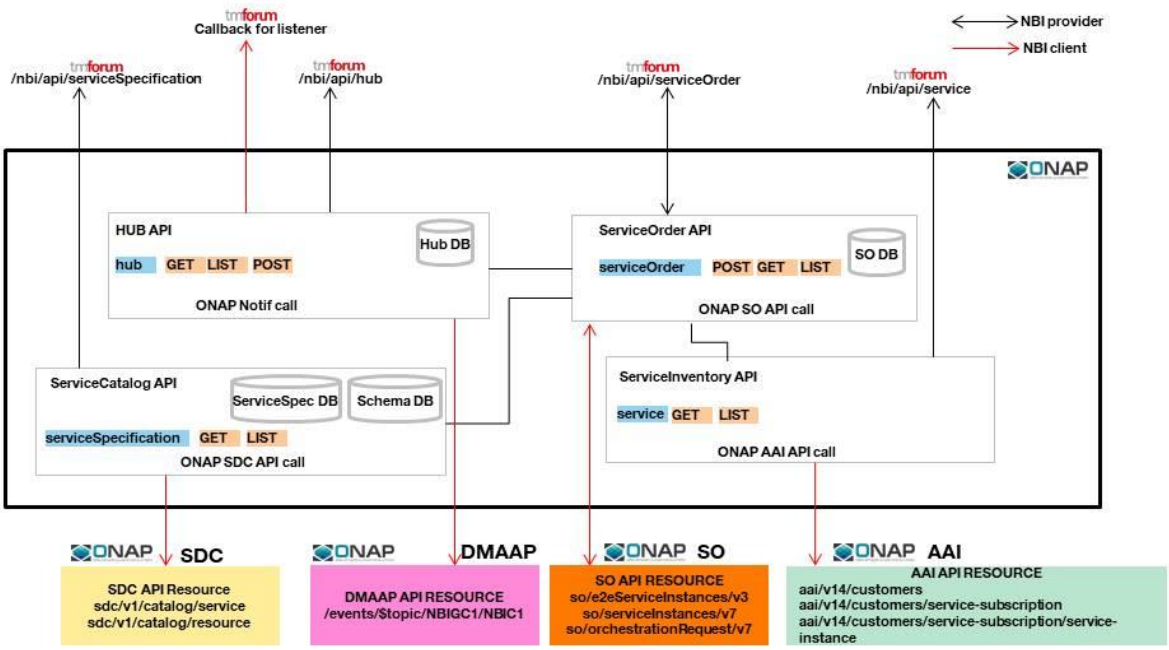

Figure 2-24 ONAP NBI architecture

2.3.14 Openslice

Openslice is a prototype open source, operations support system. It supports VNF/NSD onboarding to OpenSourceMANO (OSM) and NSD deployment management. It also supports TMFORUM OpenAPIs regarding Service Catalog Management, Ordering, Resource, etc.

Openslice allows Vertical Customers to browse the available offered service specifications and also allows NFV developers to onboard and manage VNF and Network Service artifacts. Figure 2-25 displays the usage of Openslice.

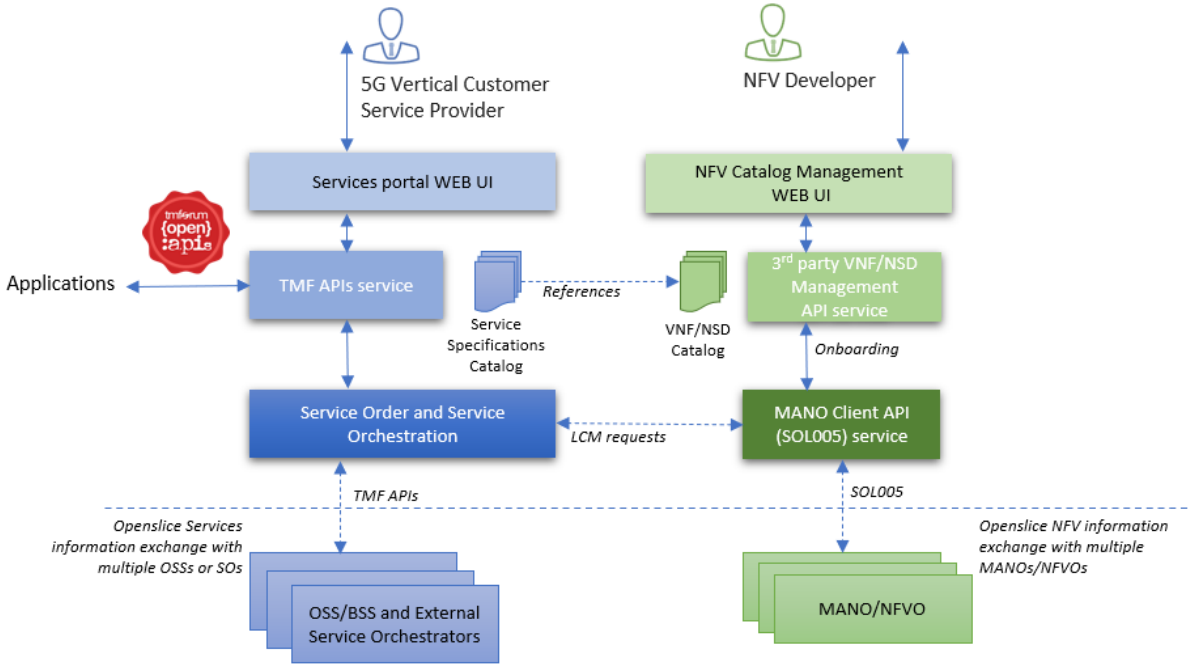

Figure 2-25 Openslice Architecture

Some of the key functionality of Openslice include:

- Service Catalog Management: A Communication Service Provider (CSP) has the ability to manage the Service Catalog Items, their attributes, organize in categories and decide what to make available to Customers.
- Services Specifications: A CSP is able to manage Service Specifications.
- Service Catalog Exposure: A CSP is able to expose catalog to customers and related parties.
- Service Catalog to Service Catalog: Openslice is able to consume and provide Service Catalog items to other catalogs.
- Service Order: The Communication Service Customer is able to place a Service Order.
- Service Inventory: The CSP is able to view deployed Services status.
- Openslice supports both APIs for programmable access to the infrastructure as well as a web portal for user friendly access.

2.3.15 Cloudify

Cloudify is an open source cloud orchestration framework. Cloudify enables the modelling of applications and services and automation of their entire life cycle, including deployment on any cloud or data centre environment. In addition, it offers monitoring of all aspects of a deployed application, detecting issues and failure, manually or automatically remediating such issues, and performing ongoing maintenance tasks. Cloudify's platform consists of a core engine responsible for the lifecycle management of applications and network services, and a set of plugins providing integration points for all needed components from cloud infrastructure automation (Compute, Storage, Network) to logging and monitoring.

Description of an application, with all of its resources (infrastructure, middleware, application code, scripts, tool configuration, metrics, and logs), in a generic, descriptive manner is performed by the Application modelling module. Cloudify has been developed to describe any application or network service in a generic, intuitive, human-readable modeling language based on TOSCA standard. Cloudify uses blueprints for the modelling stage of the applications. The blueprints enable the description of the complete service, including topology, lifecycle management, policies, and resources. Cloudify comes with pre-integrated network function blueprints, allowing users to avoid the integration processes needed for automation. Cloudify uses a visual editor and design tool that radically simplifies the creation of blueprints, called Composer. It allows a drag-and-drop of nodes, resources, and other custom items into a graph and quickly connect them to create a TOSCA-based blueprint.

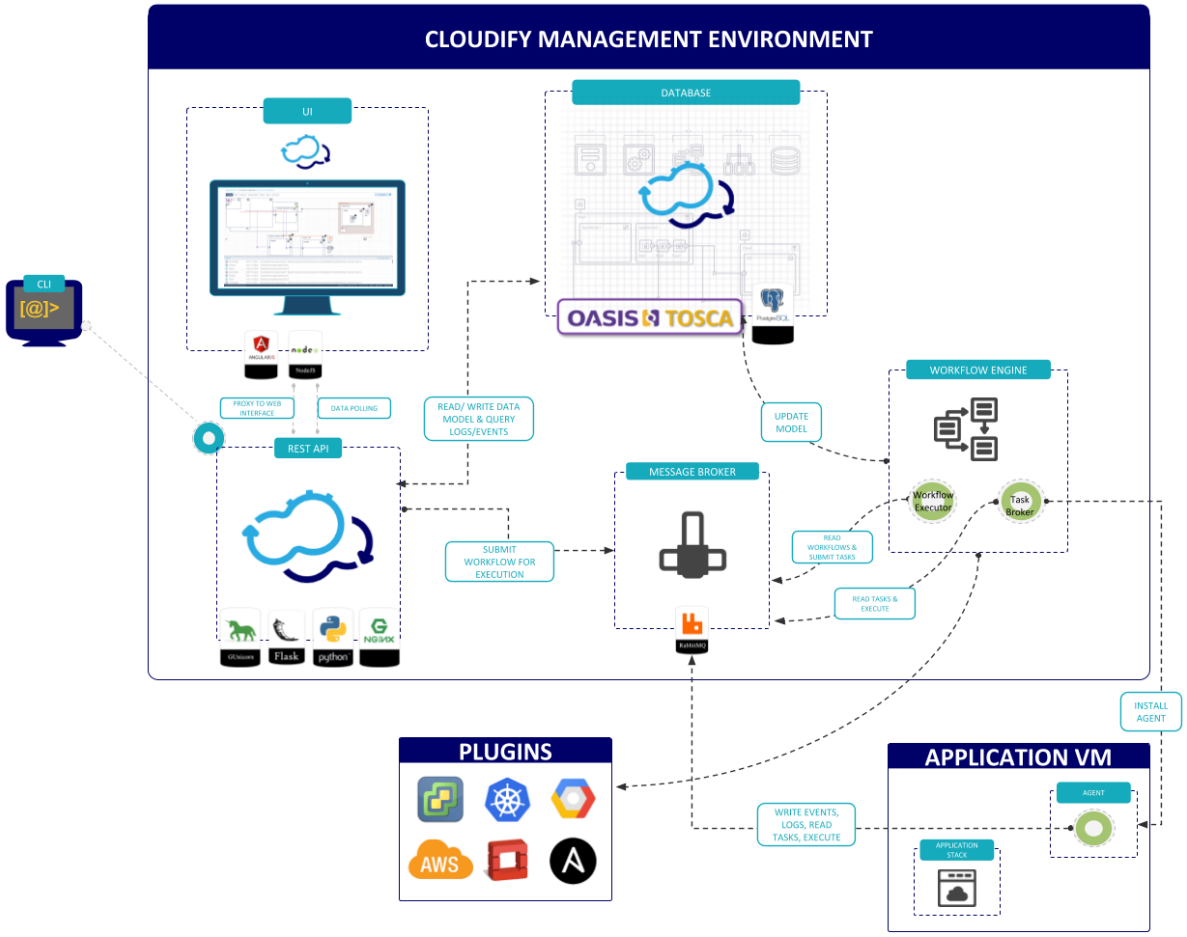

Figure 2-26 Cloudify architecture

Cloudify runs off an intelligent and declarative engine that automates the entire lifecycle management of application or network services - and can use any cloud or IT Infrastructure.

The orchestration engine provides complete lifecycle management of any application or network service including on-boarding, instantiation, day 2 change/ updates, and termination. The workflow engine is policy-driven with several built-in workflows (Heal, Scale, Update, Install…) automating service lifecycle events, designed for fault tolerance and remediation. While the Cloudify plugins extend functionality to any 'endpoint' with a programmable interface (API, NETCONF) and to non-programmable devices that are CLI or SSH based. The plugins abstract underlying infrastructure implementation by using TOSCA as the modelling language.

Other features of Cloudify include:

- an intuitive UX and dashboards allowing complete visibility and control over usage, behaviour, faults and events of application/ network services.
- Policies that enable users to analyze a stream of events that correspond to a group of nodes. The analysis process occurs in real time and enables actions to be triggered, based on outcome.
- Closed Loop Automation capabilities are built-in offering an 'agentless' monitoring including metrics collection and policy management enabling Cloudify to react to different scenarios based on changes to the state of the service. Administrators can also plug-in their own external metrics and policy engines.
- The Cloudify Console offers role-based access control, and an intuitive, customizable UI. It enables teams to monitor all deployments from a single access point, managing resources to take actions based on different events.

2.4 Gap analysis

The previous section presented a representative list of projects and solutions targeting multi-domain orchestration, most of which are based on OSM or ONAP as the underlying NFV MANO. A summary of these is presented in Table 2-11, where the key features of MANO systems are listed. For example, 5GUK Exchange [2] leverages the ETSI NFV MANO NS descriptors to develop an inter-domain orchestration brokering solution on top of the OSM NFV MANO system and also provides integrated dynamic service-based L2 cross-site connectivity capabilities. This limits its applicability because it would difficult to have full L2 control across two public domains in real scenarios.

There are a few other works that focus on the 5G multi-domain orchestration. For example, the 5GEx [17] relies on a peer-to-peer interaction of multiple MDOs, each one administered by an operator, to deploy services E2E. Each MDO further interacts with domain orchestrators which consist of SDN or NFV technologies that are responsible for the orchestration of a network segment within an operator domain. X-MANO [18] leverages the ETSI NFV MANO NS descriptors to build a cross-domain orchestration solution that can work either in a peer-to-peer or hierarchical fashion introducing different layers on top of the Open Baton NFV MANO system. The X-MANO architecture introduces Federation Agents (FAs) which provide resource availability in a domain to the Federation Managers (FMs). The FMs can in return work in a peer-to-peer manner with other FMs if needed to orchestrate the network services across multiple FMs. X-MANO does not describe the implementation details and experimental results, which makes it is difficult to compare the performance of the solutions. Furthermore, the authors do not focus on the inter-domain connectivity solutions, which we believe is one of the most important aspect of the MDOs.

Multi-domain orchestration may be implemented according to two primary concepts, either as a federation, whereby each NFVO talks with a peer NFVO to orchestrate the resources under a shared pool. Examples of this include 5G-TRANSFORMER, 5G-VINNI, and Openslice. On the other hand, in hierarchical brokering there is a central point that brokers the services across the different domains according to the requirements from the service and availability of the resources. It is also responsible for setting up any required inter-domain connectivity policies. 5GUK Exchange and 5G-PICTURE are following this principle. Having a trusted third party to interface and broker the services across multiple domains increases the scalability of the solution as each NFVO only needs to interface with a single entity (the brokering platform) and not with multiple systems. The broker does not have full control over the underlying infrastructure, which still remains under the control of the individual NFVO, but can have a full view of the exposed services across all connected domains.

Most of the solutions offer monitoring of specific KPIs either from network or compute resources. These are accessible through integrated dashboards and provide a visual overview of the real-time and historical use of the underlying infrastructure. However, only a few of the solutions investigated make actual use of these KPI to analyse and extract information in the form of performance profiles. These include MATILDA, SONATA and 5G TANGO. SONATA's profiling solution allows developers to specify both the resource configurations to be tested during a profiling run as well as the used traffic generators and their parameters. This is also followed by 5GTANGO which is based on SONATA. MATILDA on the other hand is implemented based on the adoption of the OpenCPU framework that permits the detaching of the design and implementation of an analysis process from the execution of an analysis over selected time series data. 5G-VIOS will use machine learning (ML) techniques to enable the compute of optimal configuration of available resources to meet the performance goals and SLAs for each vertical service.

Finally, none of the solutions investigated support service mobility as it is evident in Table 2-11. There are some aspects of initial service placement introduced in OSM Release 7 according to a cost function that take into considerations service requirements and available resources, but this is static and does not allow for on-line seamless service migration. 5G-VIOS will aim to offer service continuity as users move and their corresponding edge service needs to migrate and follow them.

Table 2-11 Gap Analysis of MANO systems

Projects / Features	5G PICTURE	5GUK Exchange	5G-VINNI	5GENESIS	5G-EVE	SONATA	5GTANGO	5G-TRANSFORMER
End-to-End Slice Life Cycle Management	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Service Discovery Support	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Multi-Domain Orchestration (Federation or Brokering)	Yes, hierarchical brokering	Yes, brokering	Yes, federation	No	Yes	Yes	Yes	Yes, federation
Mobility Support	No	No	No	No	No	No	No	No
Business Automation / DevOps	Limited REST APIs	Limited, provides REST APIs	Yes	No	Yes	Yes	Yes	Yes
Monitoring	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Profiling	No	No	No	No	No	Yes	Yes	No
Edge Support	No	No	Yes	No	Yes	No		Yes
Policy/SLA Management	Yes	Yes, through the Island Proxy	Yes	Yes	Yes	Yes	Yes	Yes
Standards Compliance (ETSI NFV MANO)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cloud Native Support	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Service Based Architecture	No	No	Yes	Yes	No	No, but modular	Yes	No

Projects / Features	MATILDA	SLICENET	5G-CITY	OSM	ONAP	OpenSlice	Cloudify
End-to-End Slice Life Cycle Management	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Service Discovery Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multi-Domain Orchestration (Federation or Brokering)	yes	No	No	Yes	Yes	Yes, federation	No
Mobility Support	No	No	No	No	No	No	No
Business Automation / DevOps	Yes	No	No	Yes	Yes	Yes	Yes
Monitoring	Yes	No	Yes	Yes	No	Yes	Yes
Profiling	Yes	No	No	No	No	No	No
Edge Support	Yes	No	Yes	No	Yes	Yes	No
Policy/SLA Management	Yes	Yes	Yes	Yes	Yes	No	No
Standards Compliance (ETSI NFV MANO)	Yes	Yes	Yes	Yes	Yes	Yes	No
Cloud Native Support	Yes	No	No	Yes	Yes	Yes	Yes
Service Based Architecture	Yes	No	No	Yes	Yes	Yes	Yes

3 5G-VICTORI Operation System

This section introduces the 5G-VICTORI Operation System (5G-VIOS), which is derived from previous works of 5G PICTURE with the introduction of the 5G Operating System (5G OS) and the 5GUK Exchange project with the functionality of the 5GUK Exchange service broker as an enabler of E2E orchestration with minimum overhead in complexity and performance while allowing operators to maintain full control of their infrastructure.

To abstract the complexities of the underlying 5G infrastructure and provide the common functionalities required for efficient and flexible service and slice management and orchestration, we propose the 5G-VIOS, following the concept of 5G OS [8] [19]. In short, 5G-VIOS will abstract and manage resources of a 5G system – in complete analogy to an ordinary OS. 5G-VIOS is the core part of the 5G-VICTORI architecture that enables multi-domain and multi-edge service delivery and orchestration.

Following the gap analysis in section 2.4, 5G-VIOS will be offering key functionalities including E2E slice life cycle management, service discovery and multi-domain orchestration similar to most of the solutions reviewed. However, the key innovations identified as gaps from the literature are focused on mobility management and profiling, in addition to extending inter-domain orchestration with L3 connections. 5G-VIOS will aim to offer service continuity across multiple edges while users move. The second innovation point is profiling which will make the connection between resource configurations, service demands, and performance targets defining a service profile for individual VNFs. Finally, expanding from L2 to L3 inter-connectivity with principles such as SD-WAN, will offer more flexible and wider inter-domain orchestration capabilities.

3.1 Proposed High Level Design

Contrary to 5G-PICTURE, 5GUK Exchange and the majority of the solutions that are built in a monolithic, manner, propagating service and slice management requests between components towards the actual infrastructure through function calls, 5G-VIOS is opting for a modular, cloud native architecture, following a Service Based Architecture (SBA). This captures non-functional requirements as detailed in Table 2-8. Individual components will be developed as micro-services and will be interconnected through a common bus over which they can communicate using either Pub/Sub protocols such as Kafka, RabbitMQ, particularly for extending similar buses from underlying NFVO (OSM) serving components such as monitoring and profiling, or use a REST API Gateway with HTTP/2 to connect with individual edge domains ensuring security through Role-based access control (RBAC) policies. Such architecture has several benefits including cost savings, better agility in managing the underlying infrastructure and increased velocity with respect to system releases. Figure 3-1 illustrates a high level architecture of 5G-VIOS, whereby key components are identified as new or modified using as basis previous works in 5GUK Exchange and others. Individual edges could reflect different technology or administrative domains, each with their respective NFVO and interfacing with 5G-VIOS through a proxy service.

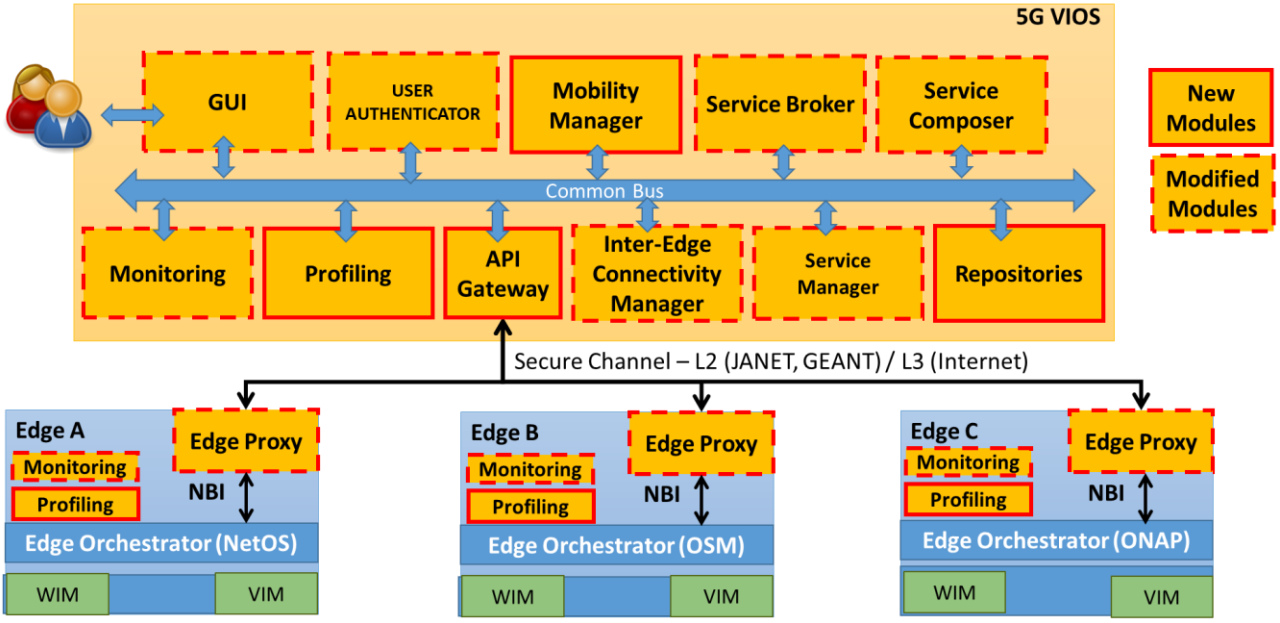

Figure 3-1 5G-VIOS High Level Architecture

3.2 Component Description

This section describes each one of the proposed components, their functionality and implementation options including existing solutions that could be used and expanded.

3.2.1 GUI

3.2.1.1 Functionality

This component in a form of a web-portal, allows end-users to interact with the 5G-VICTORI infrastructure and services. For instance, the experimental users can first register with the GUI and then access the GUI using the required credentials to login and utilize the services exposed by the 5G-VIOS. On the other hand, the edges/clusters can be registered to the 5G-VIOS through the GUI and their available network services (Catalogues, in the form of VNFD/NSDs) and the images can be onboarded to the repositories. The users can also see the list of registered islands and compose the services for an inter-edge network service. Afterwards, the users can instantiate and later deploy this inter-edge network service. The GUI shall also be utilised to visualise the network service performance profiles, monitored KPIs and performance metrics, as a dashboard utilizing existing monitoring facilities that are exported from individual edges as gathered by the Monitoring component. This functionality responds to the requirements in Table 2-1 and specifically Req.#1-#4 for accessing the services of 5G-VIOS through a GUI or and API.

3.2.1.2 Implementation Options

There are a few implementations of portals, each with varying capabilities depending on the scenarios that were investigated. For example, the 5GInFIRE project has developed a portal (https://portal.5ginfire.eu/) that allows users to deploy 5G experiments in the 5GInFIRE infrastructure. This was developed by University of Patras (UoP) and was also adopted in the 5G-VINNI project. The actual 5G-VIOS GUI implementation is left for deliverable D2.6, getting inspiration and aligning with existing implementations.

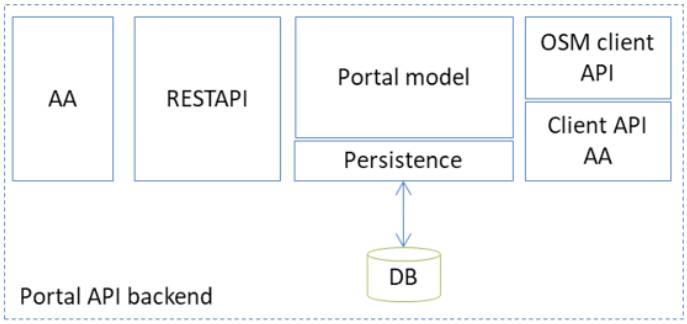

Figure 3-2 5GInFIRE web-portal architecture

Figure 3-2 displays the architecture of the portal API. It consists of the following components:

- Portal model: contains the model of entities, their definitions and associations of the portal entities like users, VxF/experiment metadata, categories, etc.
- Persistence and DB: a persistence layer to keep entities permanently available through a database system. Implementations is based on OpenJPA to keep entities permanently available through the database system based on MySQL.
- AA: Authentication and authorization mechanism(s) to allow access to the portal API. The implementation is based on OpenStack Keystone service.
- RESTAPI: implementation of the portal API server is based on Apache CFX (https://cxf.apache.org/).
- OSM client API: implementation of a client that communicates with OSM via the OSM API. Each OSM release has its own connectors.
- Client API AA: implementation of a client that is capable of communicating with another AA service(s) to authenticate/authorize users through OAuth 2.0.

The web frontend architecture consists of the following components:

- The Angular framework which supports the web implementation.
- The UI web pages facing the end users.
- The controllers for each page.
- The services that correspond to model entities (VxF, User, Experiment, etc.) and provide communication means with the API backend.

A different implementation option is the portal developed in the 5GENESIS project [20], where users can define new experiments, examine the results and logs of previous executions or manage deployed VNFs, among other features. Furthermore, experimenters can view information about their latest performed actions and access to system notices. The organization of this portal is illustrated in Figure 3-3. This portal has been implemented using the Flask framework for Python. For the frontend, the Bootstrap framework is used. The database model is based on SQLAlchemy, which allows every 5GENESIS Platform to decide between multiple backends.

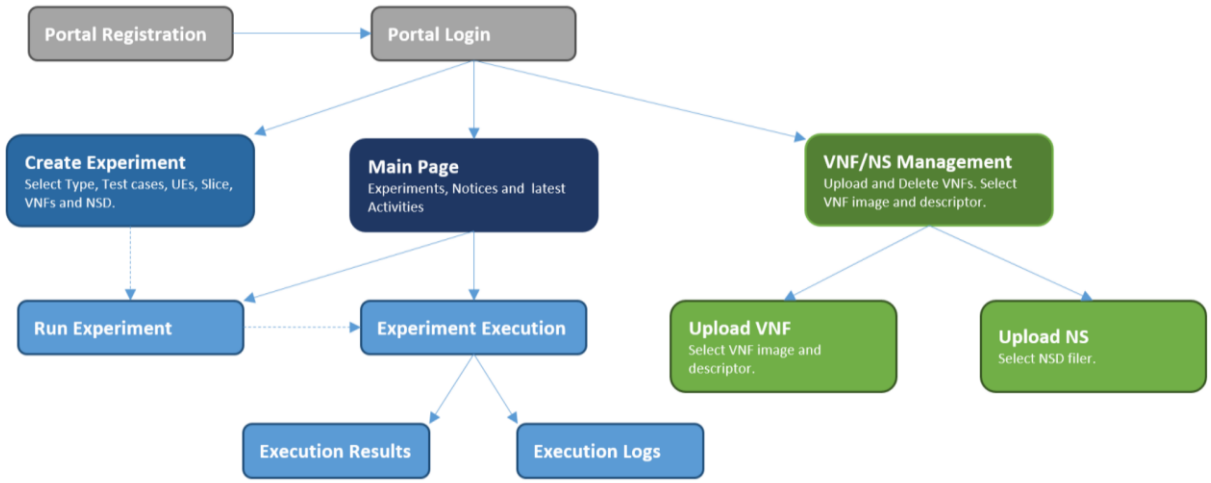

Figure 3-3 5GENESIS Portal overview [20]

3.2.2 User Authentication

3.2.2.1 Functionality

Securing Access of Users to the platform and 5G-VIOS's interfaces is paramount in achieving improved integrity, availability and confidentiality of all the data stored, transported and otherwise processed in 5G-VIOS. This corresponds to requirements in Table 2-1, particularly Req.#5-#11 and (Req.#35-#38).

To this respect, a Zero Trust Security model shall be implemented on all user identity, user access and user management controls and processes. A Zero-Trust Security model provides granular visibility into unauthorized workload access or anomalous user behaviour and further simplifies security controls.

Further expanding on this approach, a Zero Trust Model for User Access means that no user is presumed to be 'trusted' by 5G-VIOS. 5G-VIOS will leverage Multi-Factor Authentication for all its users and implement IAM – Identity Access Management Platform.

In 5G-VIOS implementations, an IAM –is used to manage the lifecycle of all User accounts that need access to 5G-VIOS. This involves the creation, modification, deletion and audit of all User accounts, both internal and external, comprehensive identity governance, privilege access management. Administrative access to the IAM platform should allow for the integration of Multi-Factor Authentication (MFA) capabilities. At the same time, an IAM shall have the capabilities to generate, store and securely delete any and all authentication artefacts such as Security Tokens, n-Th Factor Authentication Tokens for each and any user's identity.

The Role-Based Access Control component of 5G-VIOS shall have the capabilities to integrate with an existing IAM platform and securely expose an API to the 5G-VIOS components and applications that require Secure User Access.

Role-Based Access Control component should have the capabilities to:

- provide an interface for managing and provisioning User Roles, identifying user to a specific category or group;
- provide secure interfaces to other components of 5G-VIOS and applications;
- provide monitoring and audit resources;

In addition to these capabilities, all the IAM and RBAC components **shall** have comprehensively documented APIs, controls and processes.

3.2.2.2 Implementation Options

A contextual implementation of Secure User Access to the platform and its components can be referenced from 5GUK Exchange's Architecture, comprising access to several "Island Infrastructures"

by means of an Island Proxy. 5GUKEx is a lightweight hierarchical inter-domain orchestration platform and, in this deployment, the Island Proxy serves as an intermediary between the island orchestrator and 5GUKEx and as an isolation layer for security and policy purposes, as show in Figure 2-5.

3.2.3 Service Composer

3.2.3.1 Functionality

The Composer interacts with the GUI to take the input from vertical users and combine functionalities exposed from the domains/site/edges that are provided in the Repositories and plan an experiment (i.e., to request an inter-edge network service comprising of selected network functions belonging to different domains/site/edges). In addition, it allows experimenters to control the lifecycle of their service composing requests and also receive information about their status. The request is relayed to Service Manager for further operations. This component can store the composed templates of inter-edge network services in a shared/unshared database. Later, if being requested by the experimenters, this information can be utilised for deploying the same inter-edge network service. All the composed inter-edge network services shall be observed by the users through the user's dashboard. This functionality responds to the requirements of service management in Table 2-3, particularly Req. #15-#19.

3.2.3.2 Implementation Options

A service is composed of a combination of VNFs, PNFs, or CNFs and networks connecting these functions. The service is described in a file via a modelling language which can be in one of several formats. One of the common formats to describe a service is to use the TOSCA language, as reported also in Req.#45-#46.

The connection between applications in a service is often described by a VNF Forwarding Graph (VNFFG) as defined by ETSI in IFA 014 [21].

The Service Composer shall be developed in the form of container, written in Python3 programming language with the ability of providing REST APIs to interact with the other components, specifically the GUI and the Service Manager. The basis of this component is the implementation of NSC of 5GUKExchange, which would be refactored and re-packaged in a container to meet the SBA and update the corresponding APIs.

The service composer also holds a database with the corresponding VNF-FGs and also interacts with the repository. One of the options for implementing the database could be a type of non-relational database (NoSQL) such as MongoDB. As the MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time. It seems an option to address the requirements of Service composer.

3.2.4 Service Broker

3.2.4.1 Functionality

The Broker acts as an intermediary between the Edge Proxy and 5G-VIOS. It receives the Edge registration requests, which results in creating the required credentials and sending them back to the Edge Proxy. In addition, it receives the experimenter requests, check to which testbed each experimental service belongs and constructs relevant requests to forward to the corresponding edge orchestrators. During the inter-edge NS *instantiation* phase, the Broker verifies the testbed capabilities and the possibility of deploying and running the requested network services on respective Edges. So, at first, it invokes the 5G-VIOS profiling component to compute the *optimum* required resources for each service and then requests the Edge Proxies to get the resource availability status which in turn, it receives the status of the network services and forwards it to the Service Manager to update its data. If the respective Edges are capable of running the requested inter-edge network services, the Service Manager forwards a successful status to the experimenter, which in turn the experimenter can request the Service Manager to deploy the inter-edge network service. Once the NS is deployed, the Service Broker can request the Inter-Edge Connectivity Manager to create the network slice between respective

edges. This functionality responds to the requirements of service management in Table 2-3, particularly Req. #15-#19.

3.2.4.2 Implementation Options

The Service Broker shall be written in Python3 programming language with the ability of providing REST APIs to interact with the other components. The basis of this component is the implementation of NSB of 5GUKExchange, which would be refactored and re-packaged in a container to meet the SBA and update the corresponding APIs.

3.2.5 Service Manager

3.2.5.1 Functionality

Service Manager is responsible for the life cycle managements of inter-edge network services. It holds information about the capabilities of each testbed (i.e., the network function and network service catalogues). It also keeps information about the network services that have been instantiated by an experimenter to allow control of their lifecycle. The Manager is also responsible for receiving the experimenter requests which contain information on the requested inter-edge network services for an experiment and stores the requested Network services in a shared/unshared database and then forwards them to the Service Broker to deploy the network services on respective edges which in turn, the Inter-Edge connectivity manager will be utilised to interconnect the network services on respective edges. During the inter-edge network service *running* phase, it invokes the monitoring component and does the life cycle management of the network service and update the profiling component if needed. Finally, it is responsible for terminating a running inter-edge network service. In specific circumstances, such as the migration, the Mobility Manager can interact with the Manager and then the Inter-Edge Connectivity Manager to migrate a network service from one edge to the other. This functionality responds to the requirements of service management in Table 2-3, particularly Req. #15-#24.

3.2.5.2 Implementation Options

The Service Manager shall be written in Python3 programming language with the ability of providing REST APIs to interact with the other components. The basis of this component is the implementation of NSM of 5GUKExchange, which would be refactored and re-packaged in a container to meet the SBA and update the corresponding APIs. The database shared with the broker is of non-relational database (NoSQL) type such as MongoDB.

3.2.6 Repositories

3.2.6.1 Functionality

Repositories comprise descriptors of programmable HW (PNFs) and SW (VNFs, CNFs) components as well as vertical industry specific network functions. Some network functions that are necessary for the operation of vertical and cross vertical industries (i.e. synchronization, positioning, signalling, voice, etc.) as well as the Network Service Catalogues, i.e., available network services on each edge/cluster in the form of NSDs/VNFDs including the required images will be part of these repositories. Each Edge/cluster will expose a set of available vertical specific virtual and physical network functions that will be packaged and exposed through function Repositories. Then, the experimenter can invoke the Service Composer to request an inter-edge network service comprising of selected network functions belonging to different domains/site/edges provided in the Repositories. Also, if being allowed by the edge policies, if any update occurred in the VNFs/ NSs in an Edge/Cluster, the Repositories shall also be updated. This functionality responds to the requirements of repositories in Table 2-2, particularly Req. #12-#14 and with the service automation in Req. #39.

3.2.6.2 Implementation Options

As of OSM Release 7, a new functionality has been introduced. The OSM Repository is a set of NSs & VNFs descriptors related artefacts and metadata, organised with a predefined structure. It brings several improvements on the management and consumption of NSs. It has a standardized model format

to implement a consumption mechanism for a remote NS repository, which could be queried and managed by OSM abstracting from the actual storage mechanism; the interface will be exposed by HTTP requests. This approach allows other NS developers for publishing and onboard their services, pushing local artefacts and dependencies of the VNFs to the remote repository.

OSM Repositories are divided in two elements, the server side which is in principle a local or remote server with a predefined structure and version control for VNFDs and NSs. The second element is the client side, which is responsible to manage the usage of the repositories operating with simple commands to associate repositories, list the artefacts available in them and onboard packages from the repository.

As mentioned above, the repositories can be hosted either locally at each edge or centrally in a common database. In either way, 5G-VIOS would list all the available repositories and related metadata following a similar to the OSM structure and potentially adopt to capture differences in ONAP if required, as this would be the NFVO in one of the 5G-VICTORI platforms.

3.2.7 Monitoring

3.2.7.1 Functionality

An important part of the orchestrator's role is to monitor the service status and make sure it is functioning properly. In case a service or a component of the service is not functioning well or suffering from a performance bottleneck, there may be a need to trigger a healing or scaling action to overcome this situation. The Monitoring component on each Edge may utilise different Monitoring tools such as Prometheus and Zabbix to monitor running network services on Edges. By providing appropriate APIs, it allows the network operators to check the status of the deployed NSs as well as provisioning the utilisation of available computing and network resources. The Monitoring component of the 5G-VIOS can accumulate the monitoring data from all the respective edges and enables the Service Manager to do the life cycle management of inter-edge network services. In addition, the Profiling component shall use monitoring information to create models for the performance of services knows as ‘profiles’. This functionality responds to the requirements of monitoring in Table 2-5, particularly Req. #30-#34

3.2.7.2 Implementation Options

The 5G-VIOS Monitoring component could have two primary implementations depending on the tools that are used in each edge. The first is to employ polling status indicators through appropriate APIs or by subscription to notification events. These messages are actually collected by the edge proxy and passed to the Monitoring component inside 5G-VIOS.

Specifically, at the edges where the orchestrator is OSM, the OSM MON feature (‘’mon-collector’’ module) can be utilised to collect NFVI and VNF metrics whenever specified at the descriptor level. For metrics to be collected, they have to exist first at any of these two levels:

- NFVI – made available by VIM's Telemetry System (such as Gnocchi).
- VNF – made available by OSM VCA (Juju Metrics).

Figure 3-4 shows the OSM Monitoring diagram[4]. In step (1), the VIM/VNF metrics are collected from the VIM (OpenStack or VMWare) and the running VNFs. As can be seen in step (2), the Prometheus stores these metrics in its data base (TSDB) and the metrics can be retrieved through its REST APIs. Also, if needed to visualise the data, tools such as Grafana can be integrated with Prometheus (as shown in step (3).

[4] https://osm.etsi.org/wikipub/index.php/OSM_Performance_Management

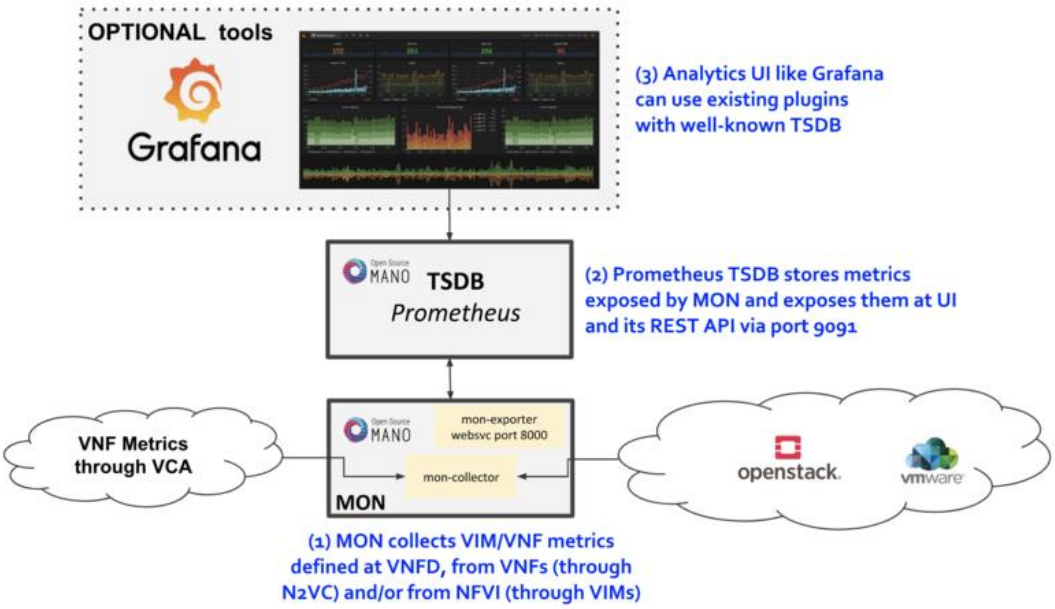

Figure 3-4 OSM (rel 5+) performance management and Monitoring diagram

Current implementation of ONAP does not offer a dedicated Monitoring component or API. However, APIs exposed from OpenStack and Prometheus can be used to capture the required KPIs. In addition, to monitor the network performance related to the network devices and Applications, Zabbix[5] can be optionally implemented. Zabbix relies on two key components for these functions: *zabbix-server* and *zabbix-agent*. Thus, it collects data on all of the network's devices (as long as the zabbix agent is installed and responding to SNMP is configured on them) and reports the data to a central controlling authority (zabbix-server).

3.2.8 Profiling

3.2.8.1 Functionality

When deploying an inter-edge network service comprising multiple VNFs hosted at edge servers, various resources such as CPU, Memory, and Network should be assigned to the involved VNFs to meet the required performance targets and SLAs specified by the UCs. The leading role of a profiling system is to make a connection between the resource configurations, service demands, and performance targets. To fulfil this capability, the Profiling component at each edge can get the monitoring metrics from the Monitoring component and create models for the performance of VNFs (Profiles) running at each edge. Then by utilising these profiles and machine learning (ML) techniques, the Profiling component at 5G-VIOS can compute the optimum configuration of available resources to meet the performance goals and SLAs for an inter-edge network service. The models used for these ML techniques will be investigated at a later stage. So, the VNFs can be deployed with an optimum amount of resource configurations and at the same time meet the KPI and performance goals. Moreover, during the life-cycle management (LCM) of the running VNFs, the profiler can monitor the utilisation of the resources at VIMs and VNFs and based on the achieved performance metrics can update the Service Manager to possibly derive LCM decisions such as scaling or migration.

The Profiling component shall provide APIs to the users or other components to provision the network service performance profiles, monitored KPIs and performance metrics.

This functionality responds to the requirements of profiling in Table 2-5, particularly Req. #34.

[5] https://www.zabbix.com/network_monitoring

3.2.8.2 Implementation Options

The Profiling component shall be written in Python3 programming language with the ability of providing REST APIs to interact with the other components as well as the users. At some edges where the orchestrator is OSM (release 5+), the 'mon-evaluator' module included in MON component can be called. In the 'mon-evaluator'. Whenever a threshold is crossed and an alarm is triggered, the notification is generated by MON and put in the Kafka bus. The OSM KPI evaluation steps is shown in Figure 3-5 [22]. As can be seen in Figure 3-5, if the OSM is installed along with the ELK stack, the following tools shall be utilised:

- Elastisearch - scalable search engine and event database.
- Filebeat & Metricbeat - part of Elastic 'beats', which evolve the former Logstash component to provide generic logs and metrics collection, respectively.
- Kibana - Graphical tool for exploring all the collected events and generating customized views and dashboards.

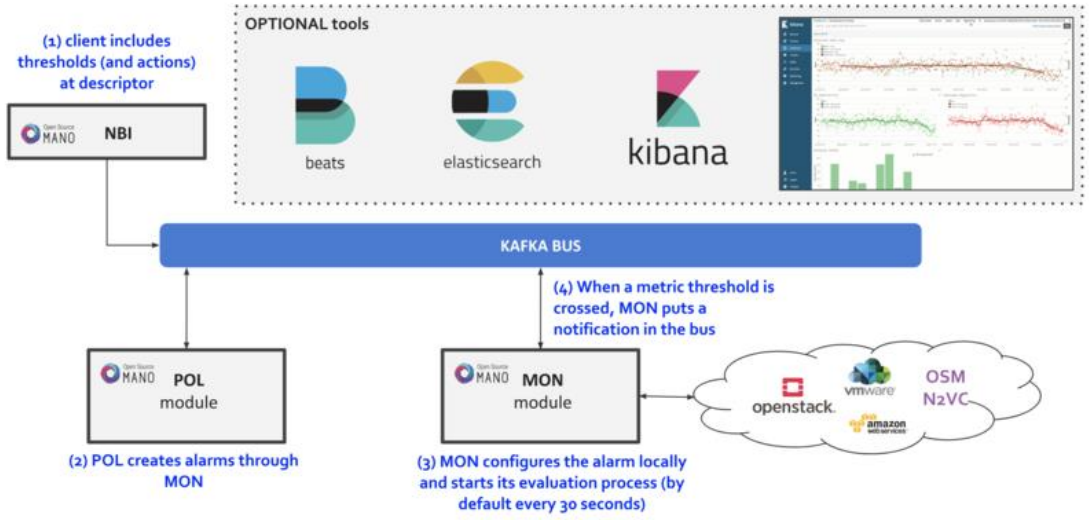

Figure 3-5 Evaluating the KPI thresholds in OSM

Therefore, if the Profiling component can access the Kafka bus and OSM is installed along with the ELK stack and the 'Mon-Evaluator' module is enabled, it can utilise some OSM capabilities in detecting the KPI thresholds. In some cases, such as (the edges where the orchestrator is not OSM or the results may not be accurate or sufficient enough, the modules to check the performance metrics and evaluating the KPI thresholds of network services (and configuring the Alarms) shall be written in Python3 programming. Moreover, the Profiling component may store the generated "VNF Performance Dataset" Records utilising the *Elasticsearch*, Logstash, and Kibana (the Elastic Stack) data repository[6]. It is noteworthy that employing the Elasticstack gives developers the advantage of a 'store first, examine later' philosophy of monitored measurements.

3.2.9 Inter-edge Connectivity Manager

3.2.9.1 Functionality

The Inter-edge Connectivity Manager (ICM) component performs the dynamic interconnection among the domains/sites that is needed to enable (or disable) an end-to-end network services by configuring the interconnection infrastructure appropriately. It has two main responsibilities: (a) serving as a bootstrapping point by setting up the control plane of the edge and connecting it to the 5G-VIOS and secondly, (b) being responsible for creating the datapath between the edges when a service is

[6] https://www.elastic.co/elasticsearch/

deployed. The ICM will interface with the edge proxy to form a secure link between 5G-VIOS and each edge using the underlying interconnection facility. This functionality responds to the requirements of network management in Table 2-4, particularly Req. #25-#29.

3.2.9.2 Implementation Options

As described above, services can be composed of a series of VNFs, PNFs and CNFs interconnected through different (virtual) networks. Controlling these (virtual) networks is the role of an SDN controller that configures the networks according to the defined parameters and required QoS. The SDN controller can be an open source one (e.g. ONOS, OpenDayLight) or a commercial one (e.g. NetOS). The ICM employs the SDN controller to create the E2E sliced Layer-2/3 network and at the same time has a database to store the endpoint and connections to monitor and terminate the connections when the service is terminated.

This module will be an extension of the Inter-domain Connectivity Manager (IDCM) introduced in 5GUKExchange, extending it to L3 connectivity solutions. It shall be written in Python3 programming language with the ability of providing REST APIs to interact with the other components. Using open source solution like VyOS (https://www.vyos.io/products/#vyos-platform), we will able to control functionalities such as firewall, IPSec, and routing required to provide the inter-edge connectivity. A database to keep track of the endpoints and connections would be of NoSQL type such as MongoDB.

3.2.10 Mobility Manager

3.2.10.1 Functionality

One of the key innovations that 5G-VIOS aspires to achieve is to introduce mobility management within the Life-Cycle Management of a service, so that it would be able to offer seamless service continuity to vertical users while they move. Connectivity continuity is a well investigated subject and there are solutions that offer horizontal and vertical handover coupled with multi-homing protocols allowing sessions to continue. However, service continuity is challenging and depends if the application is stateful or stateless, as the application needs to follow the user to a different edge instance. This is an open challenge as several issues have to be tackled such as the change of the IP address of the user, the communication between edge hosts when using Network Address Translation (NAT) and more. The combined smooth integration of network and application mechanisms is needed in order to guarantee session and service continuity during inter-edge handover. Mobility Manager will interact with the Service Manager that takes care of LCM, requesting the service migration. Initially that would be initiated by individual applications, which monitor the mobility of the users, sending a request through the API Gateway. However, this could be further extended to automated migration using the monitoring and profiling modules whereby the migration would be triggered by performance related events. This functionality responds to the requirements of service migration in Table 2-3, particularly Req. #20-#22.

3.2.10.2 Implementation Options

Individual edge instances would be mapped to underlying geo-fenced areas that they are serving. One edge may serve multiple radio access networks, so the connectivity handover may not affect or trigger service handover. However, once a user moves to a different area served by another edge instance, service migration will be triggered. Following the rule of “make before break” in order to minimize the downtime, the application should monitor the mobility of the users and inform 5G-VIOS in advance of potential movement. Mobility Manager will have a look-up table to map the geo-fenced areas with corresponding edge instances and relay the service migration to the service manager for appropriate LCM actions including instantiation of service in new edge, traffic steering and termination of the service in the previous edge. Similar to the other components, the Mobility Manager will be implemented as a micro-service using Python3 with REST APIs to interact with the rest of the modules.

3.2.11 Edge Proxy

3.2.11.1 Functionality

The Edge Proxy acts as an isolation layer for security and policy purposes and is the main interface between the edges and the 5G-VIOS. It will act as the anchor point for the secure inter-connections between edge and 5G-VIOS.

As there will be various MANO systems such as OSM, and ONAP, it should expose a northbound interface compliant with the 5G-VIOS APIs and on the other hand a southbound interface compliant with ETSI MANO APIs.

The Edge Proxy can interact with the Monitoring component to report the metrics related to the computing and network resources on the respective edges and expose them to the network operators or experimenters.

Applications running at the edge will also use the Edge Proxy southbound APIs to send requests to 5G-VIOS, particularly for service migration and reporting KPIs to the monitoring component.

During the Edge registration, the Edge Proxy exposes the network service Catalogues (available NSs at the Edges) to the 5G-VIOS.

During the inter-edge network service *instantiation* phase, by invoking the Profiling component of 5G-VIOS, the optimum amount of computing and network resources required to meet the target KPIs can be computed which in turn, the Edge Proxy can decide if the edge is capable of running the requested network service with the required amount of resources. Then, based on the policies and SLAs this component can forward the status to the Service Broker to possibly deploy the NS.

3.2.11.2 Implementation Options

This module will be an extension of the Island Proxy introduced in 5GUKExchange, extending it to support the L3 inter-connectivity and changes in the APIs for newer OSM releases, ONAP and application specific messages. The Edge Proxy shall be written in Python3 programming language with the ability of providing APIs being compliant not only with the 5G-VIOS API gateway but also with various ETSI MANO APIs (such as OSM, and ONAP).

3.2.12 API Gateway

3.2.12.1 Functionality

The API Gateway component acts as the main entry point from the individual edge instances towards the 5G-VIOS. It takes all API calls from clients, then routes them to the appropriate microservice with request routing, composition, and protocol translation. Typically, it handles a request by invoking multiple microservices and aggregating the results, to determine the best path. It can translate between protocols that are used internally. This is also known as a dispatcher.

It is exposing a Southbound API that talks with NBI of MANO and applications/services directly through the Edge Proxy, taking care of RBAC policies,

This component allows applications and services running within individual edge domains to interact with 5G-VIOS components or specific services and vice versa through the provided APIs. For instance, the applications can request the migration of one NF from one edge to the other edge utilising the provided APIs, or report KPIs to the monitoring component.

3.2.12.2 Implementation Options

5G-VIOS components are being developed as microservices and the API Gateway will be the glue to connect all of them. An API gateway is the conductor that organizes the requests being processed by the microservices architecture to create simplified experience for the user. It is a translator, taking a client's many requests and turning them into just one, to reduce the number of round trips between the client and application. An API gateway is set up in front of the microservices and becomes the entry

point for every new request being executed by the app. It simplifies both the client implementations and the microservices app.

There are a variety of different technologies that can be used to implement a scalable API Gateway. On the Java Virtual Machine (JVM) one can use one of the NIO-based frameworks such Netty, Vertx, Spring Reactor, or JBoss Undertow. One popular non-JVM option is Node.js, which is a platform built on Chrome's JavaScript engine. Another option is to use NGINX Plus. NGINX Plus offers a mature, scalable, high-performance web server and reverse proxy that is easily deployed, configured, and programmed. Other solutions include Kong, an open source API gateway that is built on top of (NGINX.) which is a very popular open source HTTP proxy server. While basic features are had with the open-source version, certain features like the Admin UI, Security, and developer portal are available only with an enterprise license.

3.2.13 Interconnection Infrastructure

3.2.13.1 Functionality

The interconnection infrastructure represents the physical aspect of the traffic forwarding and is also referred to as the fabric. It is responsible for identifying traffic based on L2 encapsulation such as VLAN or MPLS tags and forward appropriately. The solution can logically scale to several million of active slices by stacking L2 encapsulation such as QinQ for VLAN. This will be extended also to L3 interconnection using secure tunnels such as IPSec between different edges where L2 is not possible.

3.2.13.2 Implementation Options

All four clusters can be interconnected through GÉANT, the pan-European data network for the research and education community as illustrated in Figure 1-1. It interconnects national research and education networks across Europe, enabling collaboration on projects ranging from biological science, to earth observation, to arts and culture. This will enable different clusters to interconnect and run cross-facility tests offering a L3 service.

On individual clusters, there will be a mixture of connectivity technologies, primarily fibre and millimetre wave (mmWave) backhaul links that will form the connectivity fabric. SDN enabled switches would control the connectivity between different edges of a single cluster.

3.3 Interface Description

The orchestration of NFV network services in multi-domain environments brings new implications, particularly in the definition of eastbound / westbound interfaces across NFVOs to securely exchange information and expose management capabilities across domains. Several previous projects have drawn the attention on the definition of these interfaces. Among them, the most significant progress has been captured in ETSI GS NFV-IFA 030 [23] that defines the Or-Or and Os-Ma-Nfvo reference points. This Or-Or reference point enables NFVOs from different administrative domains to communicate with each other in a federated way, while the Os-Ma-Nfvo mainly matches the role of 5G-VIOS as a northbound communication. The Or-Or reference point assumes that NSs managed by one NFVO (nested NFVO, NFVO-N) are included as constituents of the NSs managed by the other NFVO (composite NFVO, NFVO-C). The processing of nesting NSs (managed by NFVO-N) into composite NSs (managed by the NFVO-C) means triggering network service orchestration functions across both NFVOs. The communication between 5G-VIOS and individual edge orchestrators shall follow the ETSI SOL005 [24] specifications with respect to network orchestration and TM Forum API [25] for service orchestration.

Some of the key interfaces that are defined in [23] and can be passed through either the Or-Or or Os-Ma-Nfvo reference point in the form of RESTful APIs.

- **NSD management interface:** allows NFVO-N to announce NSDs towards the NFVO-C, so that the later could make use of them to build composite NSs.

- **Network Service Life-Cycle Management interface:** defines the set of NS life cycle operations that the NFVO-C can invoke over a network service instance managed by the NFVO-N, when this instance has been nested into a composite NS instance.
- **Network Service Fault Management interface:** defines the fault-related alarms and events that the NFVO-C can collect from a NS instance managed by the NFVO-N, when this instance has been nested into a composite NS instance.
- **Network Service Performance Management interface:** defines the performance information/actions that the NFVO-C can collect/trigger from/towards a NS instance managed by the NFVO-N, when this instance has been nested into a composite NS instance.
- **Network Service Life-Cycle Management Granting Operation interface:** allows the NFVO-N managing a NS instance to request a grant for authorization of a life cycle operation over that instance, when it is part of a composite NS instance. This interface enables NFVO-C to approve/reject a NS life cycle operation from the NFVO-N when this may negatively impact the behaviour of the composite NS instance.
- **Network Service Instance Usage notification interface:** allows NFVO-N managing a NS to receive notifications from the NFVO-C, indicating that this instance has been nested/detached into/from a composite NS instance.
- **VNF Package Management interface.** The design of the protocol and data model for the above interfaces is based on the information model and requirements defined in ETSI GS NFV-IFA 013.

3.4 Security

5G-VICTORI should follow design documents, standard best practice and industry recommendations for the 5G network elements, functions and systems.

5G main security activities and actions are mentioned by several 3GPP standards, describing the system architecture for the 5G system [26], the security architecture and procedures for 5G System [27] and 5G system network function repository services [28], describing security requirements for user access, service registration, discovery and authorization.

The 5G system is characterized by the service-based representation, where different network functions within the control plane may enable and authorize others network functions to access services, maintaining also the profile of available network functions. 5G-VICTORI treats several 5G security aspects of 5G system resource management, from a larger set of key security topics, as described in Table 3-1.

In 5G-VICTORI we are focusing on the secure user access description, secure interfaces recommendation and secure repositories functions implementation, providing also recommendation for security patch management, security vulnerability management and security incident management.

Table 3-1 5G Security key activities

Key activities	Outputs

Common Security:
• User Access Management
• Password Management
• System hardening(configuration and settings to reduce vulnerability)
• System Security
• Log management
• Input protection
• Data protection
• Communication security
• Backup management
• Support for conducting security testing
	• Access Management principles being used
	• System Logging level
	• Backup and Restore systems
	• System Hardened
	  ○ System Ports
	  ○ APIs & communication channels encryptions
	• Application patches
	• Cluster connectivity's

Node Specific: • Link Protection • Network Access • System configuration	• Security compliance reports

3.4.1 Secure Interfaces

Securing the 5G-VIOS platform interfaces represents a challenge that needs to be treated very carefully considering that is a cross site platform (physically separated) that is interconnecting using interfaces those infrastructures and also with respect to main functional requirements described in section 2.1.

In a security context, the following actions must be taken into account as goals of the entire system to deliver to its providers and users, secure and protected access and data.

- Anonymity is used in mobile networks to prevent an attacker from being able to identify individual users and also in preventing traffic monitoring and traffic analysis by unauthorized users.
- Confidentiality: mobile networks provide encryption on any communication to and from the UE.
- Safety: the 5G mobile network is expected to feature prominently in the safety of critical national infrastructure control systems.
- Availability: Mobile networks rely upon the network architecture being designed in such a way to ensure and improve availability across the network.

Inter-domain orchestration and inter-connection of all sites that form entire 5G-VIOS relies on interfaces for each domain and also sub-interfaces between all components of a single domain such as NBI of the NFVO or Nf-Vi of the VIM. This will introduce significant security issues.

First vulnerabilities are represented by the peer to peer connections of all devices involved and also of cross site connections where both physical and software problems may appear (e.g. no build in security of devices, physical access, shared transport of data connection, etc.).

Going further into the architecture, distributed data centers, edge computing and even the applied concept of network slices combined with the virtualization technique and software components, may introduce other serious security threats that must be addressed. 5G-VIOS requirements #37 & #38 are addressing this concerns targeting the issue of network slicing and security assurance of the entire system.

Together with the requirement #36 related to prevention of Distributed Denial of Service (DDoS) attacks and malicious behavior of users, the above requirements should stabilize the system from the security point of view and protect interfaces from issues like API vulnerabilities, core components integrations, improper access control or unauthorized access of users.

To prevent some of the above malicious actions a security oriented architecture [27] needs to be started with the following actions such as securing the interconnections through layers of IPSec using VPNs, implementing multifactor authentication and enabling a token or certificate-based authentication for users and inter-connected components.

According to [28], for secure architecture interfaces some network elements and concepts need to be also applied. Network access security, is the set of security features that enables a UE (any device) to authenticate and access services via the network securely, including the 3GPP access and non-3GPP access. Network domain security, is the set of security features that enables network nodes to securely exchange signaling and user plane data. Application domain security (IV), is the set of security features that enables applications in the user domain and in the provider domain to exchange messages securely. These include the network element registration, discovery and authorization security aspects, and also the protection for the service-based interfaces.

To protect messages that are sent over some interfaces on the system architecture, in 5G a new network element called Security Edge Protection Proxy (SEPP) is introduced, that applies application layer security for all communications on a specific interface between different network functions.

3.4.2 Secure Repositories

As described by 3GPP [26], a repository function (3GPP NRF) is intended to support several functionalities, as the service discovery functions, by receiving network function discovery requests and providing the information to the network function instance, maintaining the network function profile of available instances and the supported services. 5G core implementations support state-of-the-art security protocols such as TLS to protect communication and OAuth 2.0 framework at the application layer authorizing the granted network functions to access services offered by other functions, through management APIs. The network function service based discovery and registration should support confidentiality, integrity and replay protection [27], ensuring that the network function discovery and registration are authorized, process described in Figure 3-6.

The 5G VICTORI platform establishes E2E communication for different communication systems and therefore must use different technologies and ensure the smooth functionality throughout all connected systems. This requires the use of different templates and data sources, which shall be handled using repositories.

The sources for the data repositories shall be defined and configured in 5G-VIOS to ensure, only verified and commonly accepted sources are used [2].

A process for the verification of the sources shall be implemented, to verify, that used sources may not negatively affect the 5G-VIOS system. This includes especially substantial and documented testing of the repositories.

The network function service discovery and registration shall be able to hide the available supported network functions in different administrative domains, as the request/response procedure shall support NF mutual authentication and the repository may provide authentication and authorization of network functions to establish communication between them.

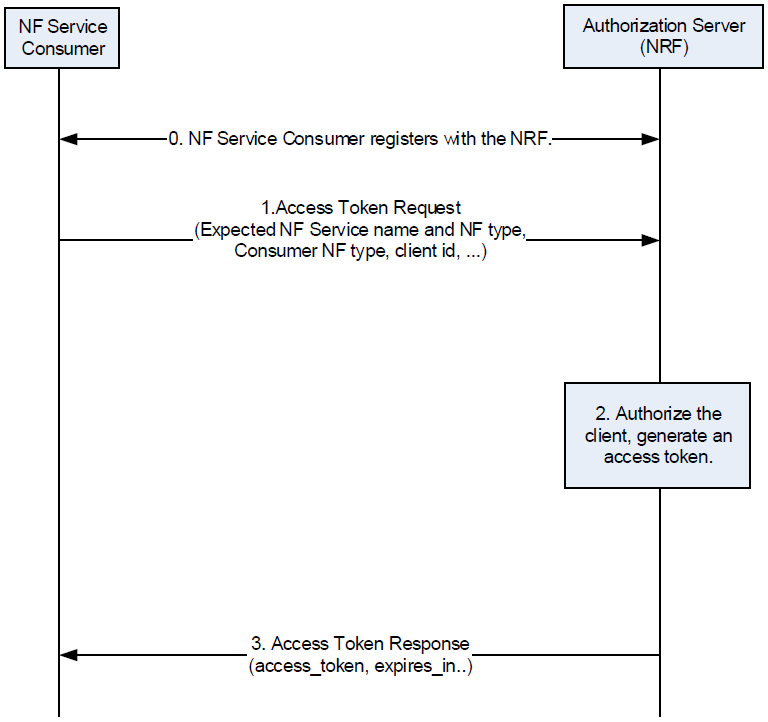

Figure 3-6 Service authorisation access to the repository functions

3.4.3 Patch Management

The 5G-VIOS system consists of many different components, such as different HW and SW. As the development of the various components speeds up over the recent years, the possibility for new vulnerabilities is significantly growing. Patches help to enhance the components and add new features, but also mitigate new vulnerabilities (e.g. DDoS, VIOS-req-47).

Changes, especially patching, may negatively affect the safety, operability or reliability of the 5G-VIOS system, if not performed correctly. For each patch, the asset owner shall gather and analyse the patch information for each device. The asset is also responsible for the extensive testing of the patch and the according documentation [29]. The identified consequences of the patch management shall be considered in the risk management process [30] [31].

For a structured process, a patch management lifecycle shall be implemented in 5G-VIOS. This starts with the availability of a patch (e.g. from a third party) and ends with the successful installation of the patch on the 5G-VIOS system. The different states, the dependencies of the states and the responsibilities, including defined state changes, shall be defined, implemented and used for every patch.

5G-VIOS is based on the Zero Trust Model, which implies, that all components should be considered as exposed and therefore the patch level should be kept adequately.

As for VIOS-req-30, the average time between the information about the availability of new patches, which are classified as “high” / “critical” or above and the successful implementation shall be measured.

The patch management process shall respect the defined SLAs to ensure the service levels for all stakeholders of the 5G-VIOS system, especially any requirements regarding the availability (VIOS-req-35).

3.4.4 Vulnerability Management

In respect to the different security layers of a 5G network, different types of vulnerabilities shall be considered. These vulnerabilities may have a negative impact on the confidentiality, integrity and especially the availability of the 5G-VICTORI system [32]. The responsibility by providers or users of the vulnerability management shall be taken into account for the following security layers [26]:

- Services, Applications and UCs
- Users and Things
- Inter-networking
- 5G Mobile Network and Virtualisation Systems
- Physical Infrastructure

The 5G-VICTORI system shall be tested regularly by a third party for new vulnerabilities. Any possible new vulnerabilities shall be evaluated and rated, using a commonly used scheme, e.g. MITRE CVE [33]. The provider shall communicate the results, including new vulnerabilities and the corresponding ratings / scores [34], to all affected customers and stakeholder [26].

The mitigation of vulnerabilities may affect only the provider, or all involved parties. The elaboration of counter measures should include all affected parties, but at a minimum the provider. The process of identifying and rating the vulnerabilities, and the definition of counter and mitigation measures should align with best practices of risk management. The output of the vulnerability management shall be used as one of the inputs for risk management, especially for risk identification [30] [31].

With respect to the requirement VIOS-req-01, a web GUI will be implemented for users to access 5G-VIOS. The required web GUI shall be checked regularly for new vulnerabilities. Especially interfaces to external parties, as describe in VIOS-req-04, shall be checked with high attention for new vulnerabilities.

At least one KPI shall be defined to measure the vulnerability lifecycle throughout the overall lifecycle of the 5G-VIOS system.

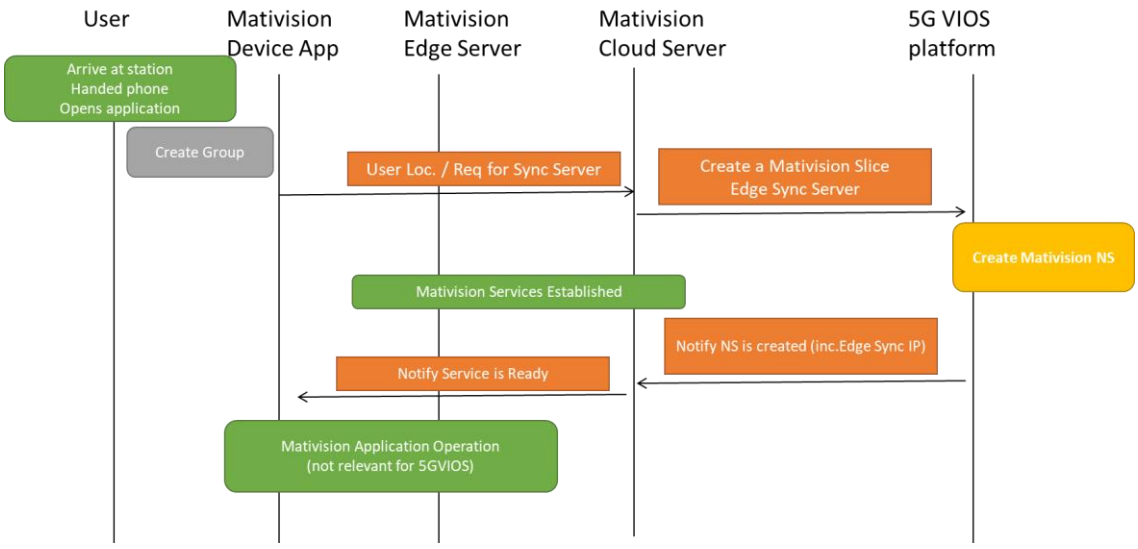

Figure 3-7 Vertical User Interaction with 5G-VIOS

3.5 Workflows

This section presents initial example workflows of how vertical users interact with 5G-VIOS and how individual components communicate internally.

The key operations that 5G-VIOS performs include:

- User Authentication.
- Network Service Composition.
- Network Service Establishment.
- Network Service Termination.
- Network Service Migration.
- Network Service / Infrastructure Monitoring.
- Network Service Profiling.
- Edge/Domain Registration/Onboarding.

As a basis we assume an application from Mativision as described in 5G-VICTORI D2.1 [1] and we investigate two primary operations of 5G-VIOS, the **service establishment** and **service migration** that support this UC. A high level user journey for this application is illustrated in Figure 3-7, where a user is handed a mobile phone with the application and once opened it triggers the instantiation of the network service at the corresponding edge. The assumption is that the vertical user has setup a cloud backend server which is running and the user application can connect. Also, all edges have registered with 5G-VIOS and published their network services. Finally, we assume that the Mativision Network Service is already composed and the required descriptors are available in 5G-VIOS. In case of failure, either due to unavailable resources or resources could not be committed E2E in a consistent atomic way that avoids conflict and over-allocation, or other error, the user would be notified that the service could not be able to be delivered. Appropriate messaging would enable the correction or modification of the NSD to support it.

The specifics and interactions among 5G-VIOS components during a Network Service Establishment, is depicted in Figure 3-8 and Figure 3-9. As mentioned above, we have assumed that the vertical user or administrator has composed the network service and an NSD is available for this particular NS defining the NFs and networks it would require. The NS may use multiple domains hence the service broker would need to send the request to each one. This would in principle pass through the Edge Proxy on each domain and be executed by the domain NFVO. Once the NFVO acknowledge the request for the NS, this is deployed on the edge domains and the inter edge connectivity manager configures the data plane connectivity among them. After the establishment of the connectivity, 5G-

VIOS notifies other components like the Monitoring to keep track of that. Finally, 5G-VIOS needs to notify the user application that the service is ready and respond with the IP information of the edge server.

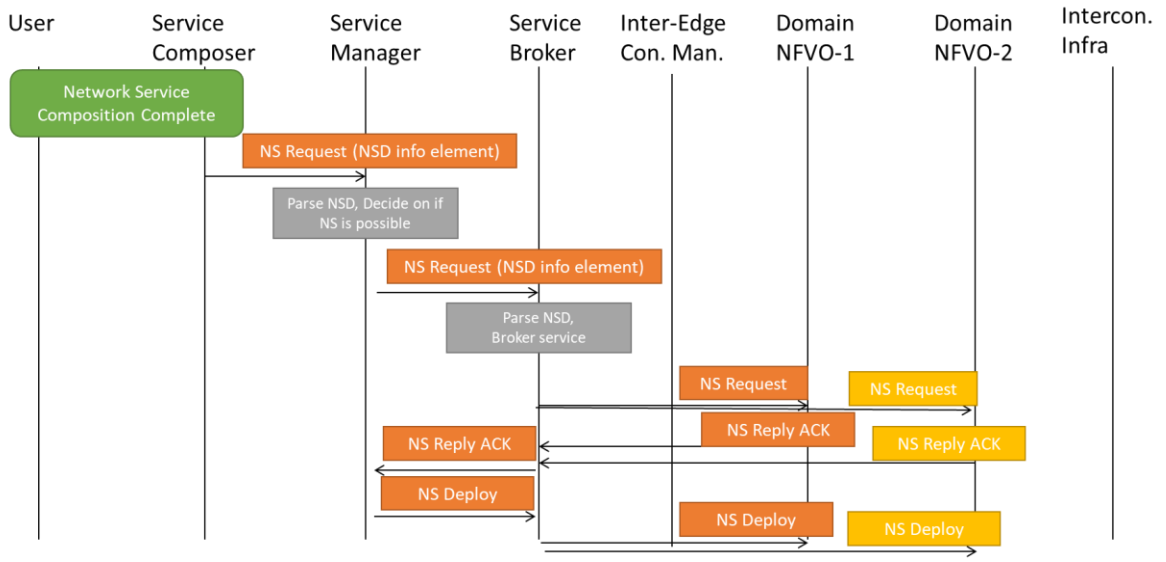

Figure 3-8 Network Service Establishment (part 1)

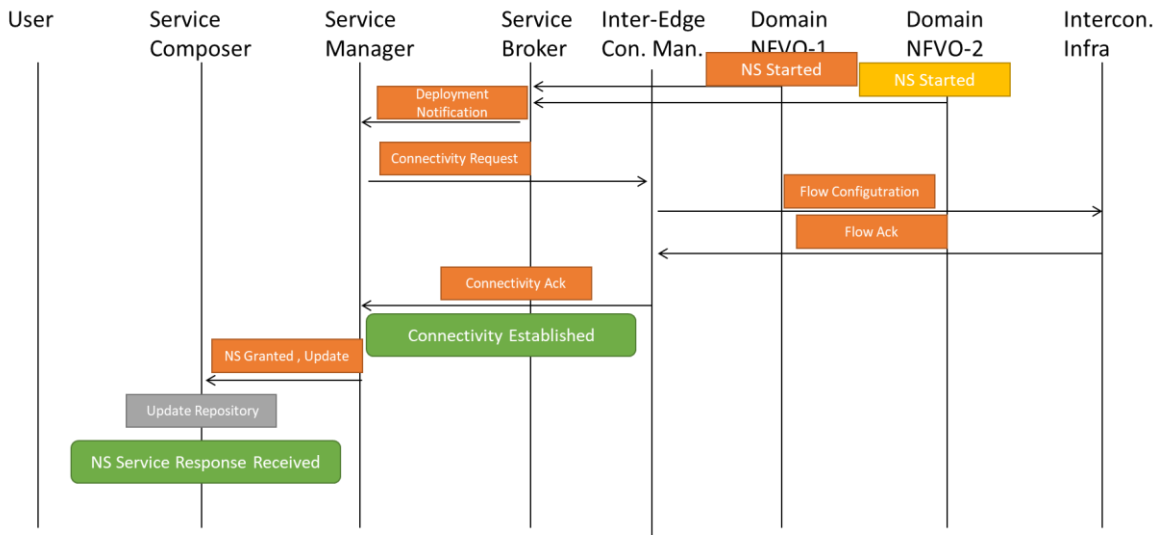

Figure 3-9 Network Service Establishment (part 2)

The second process we investigate in this section is the service migration (Figure 3-10). In this case we assume that a NS is deployed at edge location #1 and the user is moving towards location #2. The request is sent through the API Gateway to the Mobility Manager that initiates the migration process. The Mobility Manager sends a request to the service composer so that the NSD is updated to take into account the new edge infrastructure. After this is completed, the process of network service establishment described above is followed assuming that the NS is deployed in edge location #2 only. Once the NS is established in the new location, a NS termination process is initiated which would release the resources used at edge location #1 and tear down any networks not required. At the same time though, 5G-VIOS informs the application of the new NS information, e.g. the IP of the edge server.

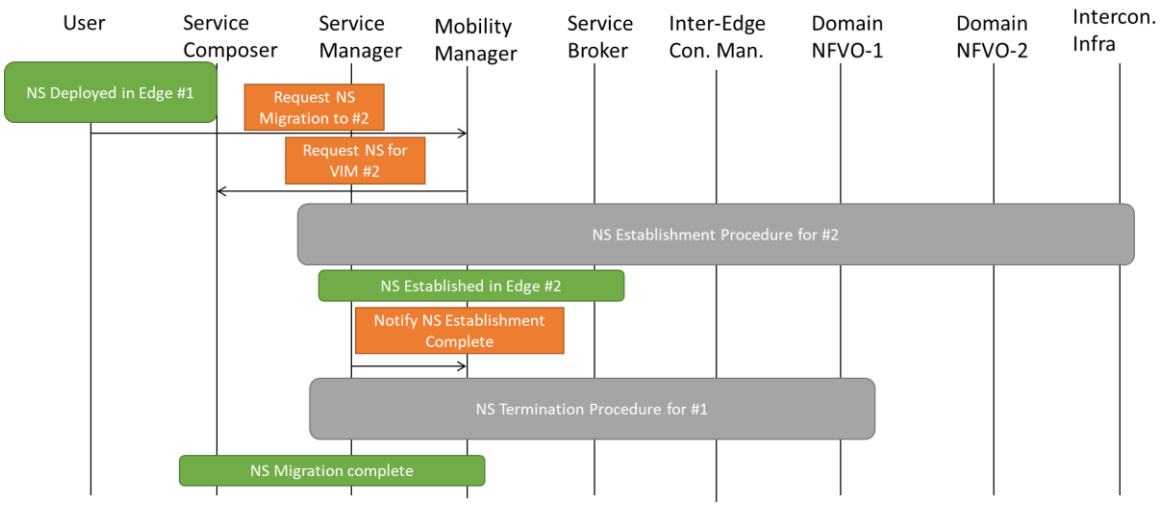

Figure 3-10 NS Migration Process

4 Conclusions

The 5G-VICTORI project is developing on the concept of a single E2E platform across multiple facilities providing interconnection and interworking, thus creating a common infrastructure of integrated network and compute/storage resources. To facilitate this, a multi-domain orchestration platform is being developed, the 5G-VIOS.

Some of the key features that this platform aims to offer include:

- E2E Slice Life Cycle Management
- Service Discovery Support
- Multi-Domain Orchestration through brokering of services
- Mobility Support for service migration
- Business Automation / DevOps
- Monitoring & Profiling
- Edge Support
- Policy/SLA Management
- Standards Compliance (ETSI NFV MANO)
- Cloud Native Support
- Service Based Architecture

A review of the state-of-the-art solutions from other previously running 5G-PPP projects or open source solutions tackling similar challenges, reveals that features such as mobility support and profiling are not adequately supported, while multi-domain orchestration is still challenging. Some of these projects set the starting point for 5G-VIOS, including 5GUK Exchange and 5G-PICTURE. As most of the facilities are using ETSI OSM as their NFVO, interfacing and expanding on its capability would be key for the project.

This work represents a starting point for the technical design with an initial 5G-VIOS specification including a high level design architecture and component description. Preliminary analysis of the processes and the components that support them, security and workflows are included but are expected to be enhanced at later stage once implementation is complete.

5 Bibliography

[1] 5G-VICTORI, “D2.1 5G-VICTORI Use case and requirements definition and reference architecture for vertical services,” 2020.

[2] N. Uniyal, A. S. Muqaddas, D. Gkounis, A. Bravalheri, S. Moazzeni, F. Sardis, M. Dohler, R. Nejabati and D. Simeonidou, “5GUK Exchange: Towards Sustainable End-to-End Multi-Domain Orchestration of Softwarized 5G Networks,” *Computer Networks,* vol. 178, 2020.

[3] ETSI, “NFV Architecture,” [Online]. Available: https://www.etsi.org/images/articles/NFV%20Architecture.svg. [Accessed 7 July 2020].

[4] ETSI GS NFV-IFA 010 V2.1.1, “Network Functions Virtualisation (NFV); Management and Orchestration; Functional requirements specification,” 2016.

[5] 3GPP TR 28.0801, “Telecommunication management;Study on management and orchestration of network slicing for next generation network,” 2016.

[6] 3GPP TS 28.533 , “Management and orchestration; Architecture framework,” 2018.

[7] 5G-VICTORI, “Deliverable D4.1 - Field Trials Methodology and Guidelines,” 2020 (to appear).

[8] 5G-PICTURE, “Deliverable D5.1 Relationships between Orchestrators, Controllers and slicing systems,” 2018.

[9] 5G-PICTURE, “Deliverable D5.4 Integrated Prototype (acress tasks and work packages),” 2020.

[10] 5G-VINNI, “Deliverable D2.1 - 5G-VINNI Solution facility sites High Level Design (HLD) - v1,” 2019.

[11] “5GENESIS deliverable D2.3 "Initial Tests and Experimentation",” 2019.

[12] “5GENESIS deliverable D3,1 "Management and Orchestration",” 2019.

[13] “Deliverable D3.15, Experiment and Lifecycle Manager (Release A),” 5GENESIS, 2019.

[14] ETSI GS NFV-TST 001, “Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services,” 2016.

[15] H. Khalili, A. Papageorgiou, S. Siddiqui, C. Golman-Meixner, G. Carrozzo, R. Nejabati and D. Simeonidou, “Network Slicing-aware NFV Orchestration for 5G Service Platforms,” in *European Conference on Networks and Communications (EuCNC)*, 2019.

[16] A. Papageorgiou, A. Fernandez-Fernandez, S. Sidduqui and G. Carrozzo, “On 5G network slice modelling: Service-, resource-, or deployment-driven?,” *Computer Commuications,* vol. 149, pp. 232-240, 2020.

[17] C. J. Bernardos, B. P. Gerö, M. D. Girolamo, A. Kern, B. Martini and I. Vaishnavi, “5GEx: realising a Europe-wide multi-domain framework for software-defined infrastructures,” *Transactions on Emerging Telecommunications Technologies,* vol. 27, no. 9, pp. 1271-1280, 2016.

[18] A. Francescon, G. Baggio, R. Fedrizzi, E. Orsini and R. Riggio, “X-MANO: An Open-Source Platform for Cross–domain Management and Orchestration,” in *IEEE International Workshop on Open-Source Software Networking (OSSN 2017)*, 2017.

[19] S. Dräxler, H. Karl, H. Razzaghi Kouchaksaraei, A. Machwe, C. Dent-Young , K. Katsalis and K. Samdanis, “5G OS: Control and Orchestration of Services on Multi-Domain Heterogeneous 5G Infrastructures,” in *European Conference on Networks and Communications (EUCNC2018)*, 2018.

[20] 5GENESIS, “Deliverable D3.7 - Open APIs, service level functions and interfaces for verticals (Release A),” 2019.

[21] ETSI GS NFV-IFA 014, “Network Function Virtualisation (NFV); Management and Orchestration' Network Service Templates Specification”.

[22] ETSI OSM, “Fualt Management,” [Online]. Available: https://osm.etsi.org/wikipub/index.php/OSM_Fault_Management. [Accessed 9 July 2020].

[23] ETSI GS NFV-IFA 030, “NFV Release 3 - Management and Orchestration - Multiple Administrative Domain Aspect Intereces Specification,” 2018.

[24] ETSI GS NFV-SOL 005, “NFV Release 2 - Protocols and Data Models - RESTful protocols specification for the Os-Ma-nfvo Reference Point,” 2020.

[25] TM Forum TMF620, “Product Catalog Management API - REST Specificiation,” 2018.

[26] DCMS Phase 1 5G Testbeds and Trials Programme, “5G Network Architecture and Secuirty,” 2018.

[27] CISCO, “Securing the 5G Core (5GC) and Evolved Packet Core (EPC) with Cisco Security White Paper,” 2019.

[28] 5G AMERICAS, “The evolution of Security in 5G,” 2018.

[29] ANSI/ISA, “ANSI/ISA-TR62443-2-3-2015, Security for industrial automation and control systems Part 2-3: Patch management in the IACS environment,” 2015.

[30] ISO, “ISO 31000:2018 - Risk Management - Guidelines,” 2018.

[31] ISO, “ISO/IEC 27005:2018 - Information Technology - Security techniques - Information security Risk Management,” 2018.

[32] European Commision, “Commision Recommendation of 26/3/2019 - Cybersecurity of 5G networks,” 2019.

[33] CVE, “Common Volnerabilities and Exposures,” [Online]. Available: https://cve.mitre.org/. [Accessed 6 July 2020].

[34] CVE, “CVE security volnerability database,” [Online]. Available: www.cvedetails.com. [Accessed 6 July 2020].

[35] 5G-PICTURE, “Deliverable D5.2 Auto-adaptive hierarchies,” 2019.

6 Acronyms

Acronym	Description
5G OS	5G Operation System
5GT-MTP	5G Transformer Mobile Transport and Computing Platform
5GT-SO	5G Transformer Service Orchestrator
5GT-VS	5G Transformer Vertical Slicer
5G-VIOS	5G-VICTORI Operation System
API	Application Programmable Interface
CN	Core Network
CNF	Containerised Network Function
CNFD	CNF Descriptor
CPU	Central Processing Unit
CSP	Communication Service Provider
DDoS	Distributed DoS
DE	Development Environment
DoS	Denial of Service
DSE	Dynamic Slicing Engine
E2E	End to End
ELK	Elasticsearch, Logstash, Kibana
ETSI	European Telecommunications Standards Institute
ETSI NFV-IFA	ETSI NFV Interfaces and Architecture
ETSI NFV-SOL	ETSI NFV Solution
FCAPS	Fault, Configuration, Accounting, Performance & Security
GUI	Graphical User Interface
HNF	Hybrid Network Function
I/O	Input/Output
IAM	Identity Access Management
ICM	Inter-edge Connectivity Manager
IDCM	Inter Domain Connectivity Manager
iNS	Inter-domain Network Service
iNSD	iNS Descriptor
KPI	Key Performance Indicator
L2	Layer-2 of OSI stack
L3	Layer-3 of OSI stack
LCM	Life-Cycle Management
MANO	Management and Orchestration
MDO	Multi Domain Orchestror
MEC	Multi-Access Edge Compute
MFA	Multi Factor Authentication
ML	Machine Learning
MON	Monitoring
MPLS	Multi Protocol Label Switching

NBI	North Bound Interface
NFV	Network Function Virtualization
NFVI	Network Function Virtualisation Infrastructure
NFVO	NFV Orchestrator
NFVO-C	NVFO Composite
NFVO-N	NFVO Nester
NMS	Network Management System
NRF	Network Repository Function
NS	Network Service
NSB	Network Service Broker
NSC	Network Service Composer
NSD	NS Descriptor
NSI	Network Slice Instance
NSM	Network Service Manager
NST	Network Service Template
OAI	Open-Air Interface
ONAP	Open Network Automation Platform
OSM	Open Source MANO
OSS/BSS	Operations Support Systems / Business Support Systems
PNF	Physical Network Function
PNFD	PNF Descriptor
POL	Policy
PoP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RBAC	Role Based Access Control
RO	Resource Orchestrator
SC	Service Chaining
SDK	Service Development Kit
SDN	Software Defined Networking
SD-WAN	Software Defined Wide Area Network
SEPP	Security Edge Protection Proxy
SLA	Service Level Agreement
SLPOC	Single Logical Point of Contact
SNMP	Simple Network Management Protocol
TLP	Telecom Layer Platform
TOSCA	Topology and Orchestration Specification for Cloud Applications
VAO	Vertical Application Orchestration
VCA	VNF Configuration and Abstraction
VDU	Virtual Development Unit
VIM	Virtualised Infrastructure Management
VNF	Virtual Network Function

VNFD	VNF Descriptor
VNF-FG	VNF Forwarding Graph
VNFM	VNF Manager
VSB	Vertical Slide Blueprint
WAN	Wide Area Network