Deliverable D1.6

Operation, Management and Orchestration of Network Slices

Editor:	Jorge Carapinha, Altice Labs
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31/12/2020
Actual delivery date:	18/01/2021
Suggested readers:	ICT-19 projects, industry verticals seeking to conduct Testing on 5G systems
Version:	1.0
Total number of pages:	75
Keywords:	5G, Network Slicing, Management, Orchestration

Abstract

Network slicing is one of the key enablers of 5G and a fundamental feature of the 5G-VINNI facility. In particular, operation, management and orchestration of 5G network slices has been a key work area for 5G-VINNI and 5G-VINNI facility sites. This work draws upon pre-existing work by standardization bodies, as well as industry and open source initiatives. This document updates 5G-VINNI deliverable D1.3 "Design for systems and interfaces for slice operation v1". Previously published 5G-VINNI deliverables have been taken into consideration, especially D1.4 "Design of infrastructure architecture and subsystems v1" and D1.5 "Design of network slicing and supporting systems v1". This document is the final deliverable of 5G-VINNI Task T1.3, "Design of a system to support management and orchestration of slices, and slice operations (by a slice instance owner)".

[End of abstract]

Disclaimer

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

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

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

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

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

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

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP1 Architecture and Design of 5G-VINNI End-to-End Platform
Number and title of task(s)	T1.6 Design of a system to support management and orchestration of slices, and slice operations (by a slice instance owner)
Document title	Operation, Management and Orchestration of Network Slices
Editor: Name, company	Jorge Carapinha, Altice Labs
Work-package leader: Name, company	Dan Warren, Samsung

Copyright notice

Executive summary

The 5G-VINNI project is aimed at developing an advanced 5G E2E facility that is able to support the execution of vertical use case trials, demonstrating the value of 5G solutions and ultimately fostering the widespread adoption of 5G technologies. Network slicing is the key to build multiple logical networks addressing different requirements on top of a common and shared infrastructure. This enables the creation of specific use cases and services, tailored to individual customers or vertical industries.

This document is mainly focused on operation, management and orchestration of 5G network slices. It updates 5G-VINNI deliverable D1.3 "Design for systems and interfaces for slice operation v1" and draws upon previously published 5G-VINNI deliverables, especially D1.4 "Design of infrastructure architecture and subsystems v1" and D1.5 "Design of network slicing and supporting systems v1". This is the final deliverable of 5G-VINNI Task T1.3 "Design of a system to support management and orchestration of slices, and slice operations (by a slice instance owner)". T1.3 is part of WP1, aimed at laying the foundations of the 5G-VINNI E2E facility design and the relevant 5G components.

A key target of this document are the internal 5G-VINNI activities to be conducted with a view to materializing and operationalizing 5G network slicing at the facility sites. Nonetheless, the information contained in this document should also be of interest to network architects and engineers aiming at deploying a 5G experimentation facility for the purposes of technological evaluation and validation.

To a large extent, 5G-VINNI builds on existing solutions, components and architectural frameworks produced by open source communities, standardization bodies and industry initiatives. An important part of this document is devoted to update the state of the art and the identification of relevant results related to 5G network slicing that had been provided in D1.3. In addition, the insightful lessons and practical experience gained from the implementation, deployment and operation of the 5G-VINNI facility sites have also been considered in this document which creates a significant added value.

List of authors

Author	Company / Affiliation
Jorge Carapinha	Altice Labs
José Bonnet	Altice Labs
Paul Muschamp	BT
Marius Corici	Fraunhofer
Santosh Kumar Rajaguru	Fraunhofer
Mohamed Gharba	HWDU
Osama Abboud	HWDU
Vasileios Theodorou	ICOM
Konstantinos Chartsias	ICOM
Dimitrios Kritharidis	ICOM
Alexios Lekidis	ICOM
Andrea F. Cattoni	Keysight Technologies
João Rodrigues	Nokia
Konstantinos Liolis	SES
Christos Politis	SES
Foivos Michelinakis	Simula Metropolitan
Andres J. Gonzalez	Telenor
Pål Grønsund	Telenor
Kashif Mahmood	Telenor
Jose Ordonez-Lucena	TID
Adrián Gallego Sánchez	UC3M
Carmen Guerrero	UC3M
David Badia	UC3M
Christos Tranoris	University of Patras
Spyros Denazis	University of Patras

Table of Contents

List of Figures

List of Tables

Abbreviations

3GPP	3rd Generation Partnership Project
5G	Fifth Generation (mobile/cellular networks)
5G PPP	5G Infrastructure Public Private Partnership
5GAA	5G Automotive Association
5G-ACIA	5G Alliance for Connected Industries and Automation
5GC	5G Core
ACTN	Abstraction and Control of Traffic-Engineered Networks
AI	Artificial Intelligence
ALTO	Application-Layer Traffic Optimization
AMF	Access and Mobility Management Function
API	Application Programming Interface
APN	Access Point Name
AR	Augmented Reality
B5G	Beyond 5G
BBF	Broadband Forum
BSS	Business Support System
CaaS	Container as a Service
CFS	Customer Facing Service
CISM	Container Infrastructure Service Management
CNF	Container Network Function
CP	Control Plane
CPU	Central Processing Unit
C-RAN	Cloud-RAN (also referred to as Centralized-RAN)
CRUD	Create, Read, Update, Delete
CSP	Communication Service Provider
DECOR	Dedicated Core Network
E2E	End-to-End
eMBB	enhanced Mobile Broadband
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FTE	Flexible Traffic Engineering
gNB	next generation NodeB
GSMA	GSM Alliance

GST	General Service Template
GST	Generic network Slice Template
HNF	Hybrid Network Function
HSS	Home Subscriber Service
HTTP	Hypertext Transfer Protocol
ICM	Infrastructure Control and Management
IETF	Internet Engineering Task Force
IM	Information Model
IMSI	International Mobile Subscriber Identifier
IOC	Information Object Class
IoT	Internet of Things
IP	Internet Protocol
K8s	Kubernetes
KNF	Kubernetes Network Function
KPI	Key Performance Indicator
L2SM	Layer Two Virtual Private Network Service Model
L3SM	Layer Three Virtual Private Network Service Model
LCM	Lifecycle Management
LPWA	Low-Power Wide Area
LSO	Lifecycle Services Orchestration
LTE	Long Term Evolution
M&O	Management & Orchestration
MaaS	Monitoring-as-a-Service
MANO	Management and Network Orchestration
MEC	Multi-access Edge Computing
MEF	Metro Ethernet Forum
mIoT	massive Internet of Things
ML	Machine Learning
MME	Mobility Management Entity
mMTC	massive Machine Type Communications
MNO	Mobile Network Operator
MOI	Managed Object Instances
MPLS	Multiprotocol Label Switching
MS	Management Service
MSBN	Multi-service Broadband Network

MTNSI	Mobile-Transport Network Slice Interface
NAT	Network address translation
NBI	Northbound Interface
NEST	Network Slice Template
NF	Network Function
NFV	Network Function Virtualization
NFVI	Network Functions Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
NG-RAN	Next Generation RAN
NOP	Network Operator
NR	New Radio
NRM	Network Resource Model
NSA	Non-Standalone
NSaaS	Network Slice as a Service
NSD	Network Service Descriptor
NSDT	Network Slice Design Team
NSI	Network Slice Instance
NSMF	Network Slice Management Function
NSSF	Network Slice Selection Function
NSSI	Network Slice Subnet Instance
NSSMF	Network Slice Subnet Management Function
NST	Network Slice Template
OCh	Optical Channel
ODU	Optical Channel Data Unit
ONAP	Open Network Automation Platform
OSM	Open Source MANO
OSS/BSS	Operations Support System / Business Support System
PaaS	Platform-as-a-Service
PCF	Policy Control Function
PGW	Packet Data Network Gateway
PLA	Placement optimization module
PLMN	Public Land Mobile Network
PNF	Physical Network Function
PoP	Point of Presence
QoS	Quality of Service

RAN	Radio Access Network
REST	Representational State Transfer
RFS	Resource Facing Service
RIC	RAN Intelligent Controller
RT	Real Time
SBI	Southbound Interface
SBMA	Service-Based Management Architecture
SD	Service Descriptor
SDI	Software Defined Infrastructure
SDN	Software Defined Network
SLA	Service Level Agreement
SLO	Service Level Objective
SLS	Service Level Specification
SMF	Session Management Function
SO	Service Orchestrator
SOF	Service Orchestration Function
SP	Service Platform
SP	Service Provider
SST	Slice/Service Type
SUCI	Subscription Concealed Identifier
TaaS	Testing-as-a-Service
TMF	Tele Management Forum
TN	Transport Network
TNSM	Transport Network Slicing Management
T-NSSMF	Transport NSSMF
TOSCA	Topology and Orchestration Specification for Cloud Applications
TSC	Transport Slice Controller
TSE	Transport Slice Endpoint
TSRE	Transport Slice Realization Endpoint
UE	User Equipment
UML	Unified Modeling Language
uRLLC	Ultra Reliable Low Latency Communications
V2X	Vehicle-to-Everything
VIM	Virtualized Infrastructure Manager
VINNI-SB	5G-VINNI Service Blueprint

VINNI-SD	5G-VINNI Service Descriptor
VNF	Virtual Network Function
VNFD	VNF Descriptor
VPN	Virtual Private Network
VR	Virtual Reality
WID	Work Item Description
WIM	WAN Infrastructure Manager
YAML	YAML Ain't Markup Language
YANG	Yet Another Next Generation
ZSM	Zero-touch network and Service Management

Definitions

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

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

Terminology used within the 5G-VINNI project draws a separation between '**5G-VINNI Facility**' and '**5G-VINNI Facility Site**'. These terms are used as follows:-

- **5G-VINNI Facility** – this is the total of all parts of the 5G-VINNI network, comprising of the individual 5G-VINNI Facility Sites, and links that are used to interconnect between 5G-VINNI Facility Sites.
- **5G-VINNI Facility Site** – this is the infrastructure and capability built at a single location, associated with 5G-VINNI. This is limited to the infrastructure built under the control of a single 5G-VINNI operational partner.

1 Introduction

1.1 Objective of this document

This document is the second and final deliverable of 5G-VINNI Task 1.3 “Design of a system to support management and orchestration of slices, and slice operations (by a slice instance owner)”. As stated in the project description [1], the objective of T1.3 is to provide the technical foundations to design a supporting system for slice operations and service orchestration, including aspects such as slice management, multi-domain, and relevant APIs/protocols.

This report can be seen as an evolution of D1.3 [2] – topics that were previously addressed in D1.3 are revisited in the present document, including an overview of relevant standardization activities and industry initiatives. Since D1.3 was published, the work conducted in the 5G-VINNI facility sites has provided a breadth of knowledge in relation to network slicing orchestration and management. Lessons learned from the activities carried out by 5G-VINNI have provided useful guidelines, which are reflected in this document, particularly in section 3.

1.2 Scope and relationship with other 5G-VINNI deliverables

The present document is closely related with previous 5G-VINNI deliverables:

- D1.4 “Design of infrastructure architecture and subsystems v2” [3] (which updated D1.1 “Design of infrastructure architecture and subsystems v1” [4]) provides a common 5G-VINNI architecture and describes the final target for the 5G-VINNI facility sites, including their interconnection. D1.6 builds on D1.4 concepts and further develops network slicing concepts, with a special focus on management and orchestration.
- D1.5 “5G-VINNI E2E Network Slice Implementation and Further Design Guidelines” [5] (which updated and refined D1.2 “Design of network slicing and supporting systems v1” [6]) provides design considerations and guidelines for end-to-end network slicing to be adopted by 5G-VINNI final release and specifies 3GPP-compliant network slicing features implemented and deployed in 5G-VINNI end-to-end facility, covering access, transport, and core networks. While both D1.5 and D1.6 address network slicing, D1.6 is especially focused on management and orchestration aspects. Generally speaking, the two reports can be seen as complementary.
- In addition, the present document takes into consideration outcomes from WP3, especially deliverable D3.1 [7], which provides an initial description of the services offered to vertical customers under the Network Slice as a Service (NSaaS) model, from a customer-facing viewpoint.

1.3 Document structure

The rest of the document is structured as follows:

- Section 2 provides an overview of the State of Art in 5G Network Slicing Orchestration and Management, particularly in terms of standardization, as well as open source and industry initiatives. It should be seen as an update of D1.3, published in July 2019.
- Section 3 addresses several topics related to management and orchestration of network slices, including Management and Orchestration of edge clouds, application of cloud native concepts to Management and Network Orchestration (MANO), integration of vertical customers, security, run-time management and multi-domain. It provides general guidelines about management and orchestration of network slicing and also includes a brief description of lessons learned from the practical experimentation activities in the facility sites.

- Section 4 updates the information provided in D1.3 on network slicing management interfaces, particularly Service Orchestration / MANO APIs and testing interfaces.
- Section 5 provides additional results of the work conducted in the framework of 5G-VINNI research topics, focused on Network Function Virtualisation (NFV) MANO towards Platform as a Service (PaaS) interoperability at the edge.
- Finally, Section 6 provides an outlook for the evolution roadmap, which essentially updates a similar analysis that was provided in D1.3.

2 State of the Art

Network slicing was brought to the limelight by 5G, particularly as an enabler of a very wide range of services and applications, with disparate requirements, over a common infrastructure. Additionally, the combination of network slicing with network automation is expected to enable a significant reduction of costs. According to a Bell Labs report, automated sliced networks will see a total of 32% cost decrease (of which, 9% is related to CAPEX and 23% to OPEX) versus the present mode of operations used in traditional WANs [8]. Nonetheless, for service providers and the communications industry at large, network slicing is still in a relatively early stage of maturity and there are still challenges ahead in particular with respect to management and orchestration of network slicing.

This section is intended to update the information (previously provided in D1.3) about the state of the art of network slicing, particularly in relation to management and orchestration. With regard to standardization bodies, updates are provided about 3GPP SA5, ETSI NFV, ETSI ZSM, IETF, BBF and MEF. In addition, four important industry / open source initiatives in this context are briefly analysed – GSM, O-RAN Alliance, OSM and ONAP.

2.1 Standardization update

2.1.1 3GPP SA5

The mission of TSG SA5 is to specify requirements, architecture and solutions for the 3GPP management system, which takes care of the provisioning, fault supervision and performance assurance of 3GPP network functions and associated services, including network slicing. To support management and orchestration of 5G networks, the 3GPP management system leverages the 5G Network Resource Model (NRM). The NRM is an information model that represents the manageable aspects of 5G networks following objected-oriented environment, with the definition of Information Object Class (IOC)[1] and Managed Object Instances (MOI)[2]. As shown in Figure 2.1:

- Vertically, the 5G NRM focus supports modelling 5G network resources (IOC specification). These resources include NG-RAN, 5G Core, Network Slice as well as Generic NRM (be reused or inherited by other domain specific model).
- Horizontally, the 5G NRM provides Stage 1, Stage 2 and Stage 3 definitions for generic or domain specific managed objects (MOI specification). Stage 1 ("requirements-level" stage) intends to provide conceptual and use case definitions for a specific network resource as well as defining subsequent requirements for this resource. Stage 2 ("information service [IS] - level" stage) provides the technology independent specification of a network resource. Stage 3 ("solution set [SS] - level" stage) finally provides the mapping of IS definitions (Unified Modelling Language, UML) into one or more technology-specific Solution Sets, e.g. Ain't Markup Language (YAML), Yet Another Next Generation (YANG).

[1] An IOC represents the management aspects of a 3GPP 5G network resource. It describes the information that can be passed/used in management interfaces. IOC has **attributes** that represents the various properties of the class of objects. Further, IOC can support **operations** providing network management services invocable on demand for that class of objects. An IOC may also support **notifications** that report event occurrences relevant for that class of objects. For example, Network Slice IOC and Network Slice Subnet IOC are used to model the management aspects of a 3GPP network slice and network slice subnet, respectively.

[2] A MOI is an instance of an IOC. Multiple MOIs (objects) can be created from an IOC (class). For example, multiple MOIs can be created from the Network Slice IOC, each associated to a different Network Slice Instance (NSI). Similarly, multiple MOIs can be created from the Network Slice Subnet IOC, each associated to a different Network Slice Subnet Instance (NSSI).

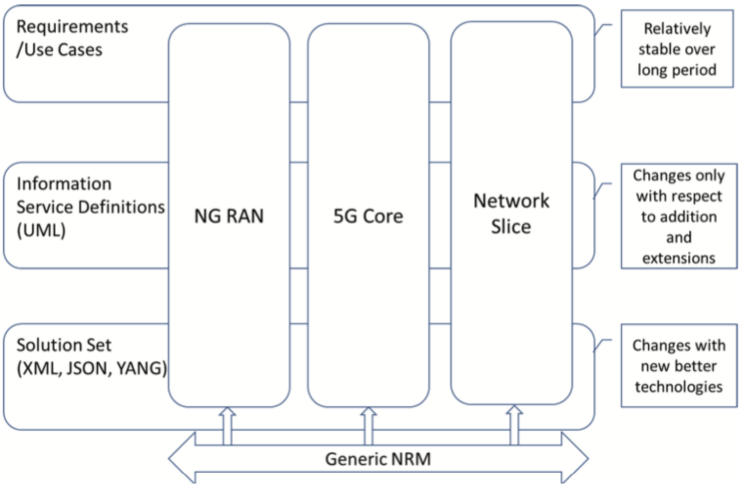

Requirements /Use Cases
Information Service Definitions (UML)
Solution Set (XML, JSON, YANG)
NG RAN
5G Core
Network Slice
Relatively stable over long period
Changes only with respect to addition and extensions
Changes with new better technologies
Generic NRM

Figure 2.1 - Scope of the 3GPP NRM for 5G network (Source: "Network Resource Model for 5G Network and Network Slice" [9])

5G-VINNI D1.3 [2] provided an overview of the Rel-15 work on network slicing management. This work was focused on the specification of network slicing in the 5G NRM, with the definition of IOCs for Network Slice and Network Slice Subnet, and the definition of corresponding management functions, including Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF).

Rel-16 made significant progress in three main directions.

Up-to-date Network Slice modelling

3GPP SA5 and GSMA have collaborated in the complex work of designing a model-based network slice specification. This collaboration, executed through Liaison Statements (LSs) exchanged between both organizations, has been based on keeping Network Slice IOC definition aligned with the GSMA's Generic Slice Template (GST) specification. This alignment responds to the need for a consistent mapping of customer-facing service requirements (GSMA domain) to resource-facing service requirements (3GPP domain). It is built upon the idea that GST attributes representing network slice Service Level Specification (SLS) need to be translated into the 3GPP *ServiceProfile*. The *ServiceProfile* is a construct defined within the Network Slice IOC that allows for the service properties of a network slice (e.g. maximum/guaranteed supported downlink, maximum/guaranteed supported uplink, isolation, packet delay budget, etc.) to be defined.

Figure 2.2 shows how GST attributes are used by 3GPP as inputs to the *ServiceProfle* and then further translated into domain specific requirements. These requirements include 3GPP network slice subnet requirements, contained on individual *SliceProfile* (i.e. 5GC *SliceProfile* and NG-RAN *SliceProfile*), and TN requirements. Finally, these domain specific requirements are translated into domain specific configuration parameters, including 5GC, NG-RAN and TN configuration parameters. Some of these parameters may be injected on individual NFs (configurable parameters), while others will be kept at the management and orchestration level (non-configurable parameter). Examples of 5GC *SliceProfile* configurable parameters include "maxDlThroughputPerSlice" (triggers configuration of corresponding UPFs) and "maxNumberOfPDUSessions" (triggers SMF configuration). Examples of 5GC *SliceProfile* non-configurable parameters include "isolationLevel".

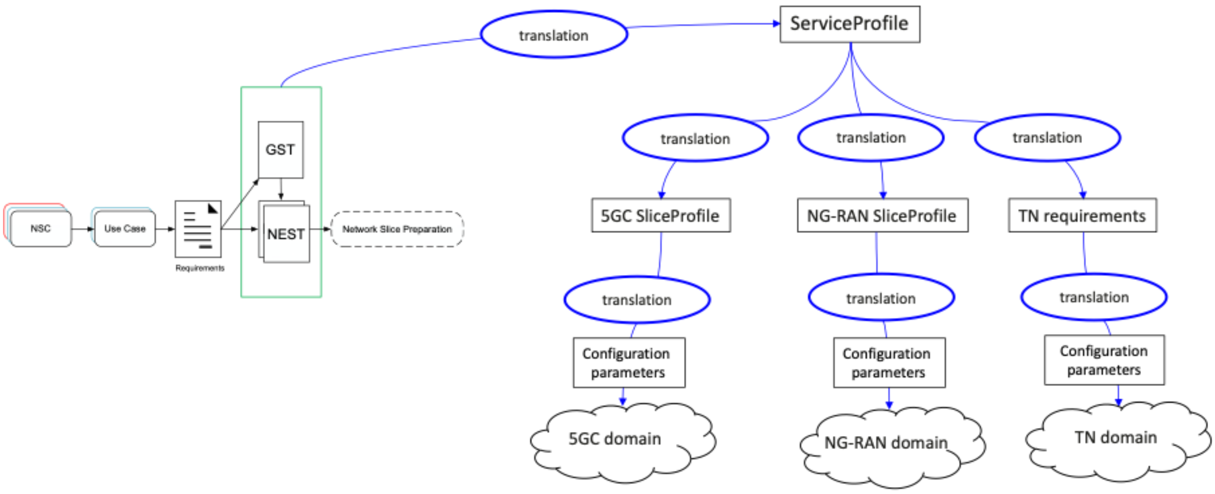


Figure 2.2 - The network slice journey - from GST to network slice configuration parameters (Source: ETSI ISG ZSM, "ETSI GS ZSM 003; End to end management and orchestration of network slicing" [10])

The process of keeping NetworkSlice IOC up-to-date, compliant with latest GST versions, has been executed through the following Rel-16 work item: "Management Aspects of 5G Service-Level Agreement" (MA5SLA).

Migration towards a service-based network slicing management capabilities

One of the main outcomes in Rel-16 has been the definition of a new architecture style for 3GPP management system, based on replacing traditional point-to-point interfaces (e.g. 3GPP Itf-N) with management services. This design approach places emphasis on the services provided by individual architectural components rather than on the relationships between pre-defined pairs of architectural components, thereby allowing 3GPP management system to move away from an integration reference point (IRP) – based architecture towards a Service-Based Management Architecture (SBMA). The SBMA adoption brought the following modifications in Rel-16 network slicing management functionality:

- The replacement of network slice related management functions with network slice related management services (MS) producer/consumer roles. This means that NSMF and NSSMF concepts are removed, and replaced by NS_MS producer/consumer roles and NSS_MS producer/consumer roles, respectively.
- The introduction of innovative management services (e.g. trace control, heartbeat control, SON control, assurance control) and the modification of the existing ones (e.g. provisioning [11], performance assurance [12] and fault supervision [13]).

This workstream has been executed through two Rel-16 work items: "NRM enhancements" (eNRM) and "Enhancement of performance assurance for 5G networks with networks, including network slicing" (5G_SLICE_ePA).

Multi-tenancy support

Rel-16 work has also focused on exploring the concepts, requirements and potential changes to 3GPP management system for network slice delivery in multi-tenant environments. This included:

- Definition of tenancy, tenant and tenant management key concepts.
- Study on different exposure of network management capabilities for different tenants, investigating the potential use of management data (e.g. performance measurements, fault alarms, logs, etc.) in multiple tenant environment.

These activities were carried out through one Rel-16 study item, “Tenancy concept in 5G networks and network slicing management” (FS_TENANCYC), further followed by a Rel-16 work item, “Enhancement of 3GPP management system for multiple tenant environment support” (MEMTANE)

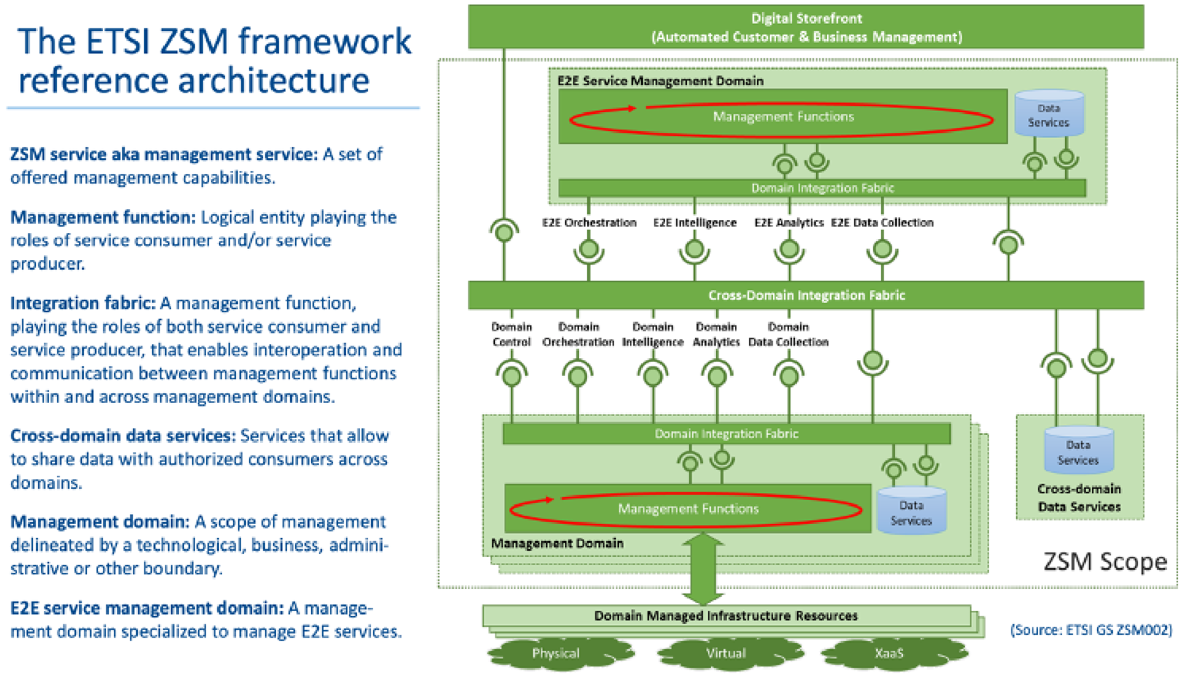
Although 3GPP Release 17 is beyond the scope of the 5G-VINNI project, some of the defined study and work items deserve to be mentioned, as they are closely aligned with the on-going 5G-VINNI research items on network slicing. The most relevant Rel-17 activities are summarized in Table 2.1.

Table 2.1 - Rel-17 work and study items on network slicing management

Work/Study Item (Acronym)	Description
Management of the enhanced tenant concept (eMEMTANE)	Enhancement of Rel-16 MEMTANE work item, assessing feasibility to introduce tenancy artefacts in Network Slice (Subnet) IOC.
Enhancement of Management Aspects of 5G SLA (eMA5SLA)	Continuation of Rel-16 MA5SLA work item, providing up-to-date correspondence between GSMA GST and 3GPP *ServiceProfile*, now with a more focus on the support of Rel-17 related mIoT and V2X services.
Study on network slice management enhancements (FS_NSMEN)	Investigate and propose the potential new management capabilities to support three use cases: E2E network slicing, cross-operator network slice provisioning, and management of slice security (e.g. security isolation, user plane protection policy).

2.1.2 ETSI ZSM

Section 2.2.3 in D1.3 [2] summarized the ZSM architecture principles, based on the use of a SBMA and the decoupling of end-to-end service management domain from individual network management domains, highlighting how the combined use of the ZSM features allows E2E network slicing delivery. Figure 2.3 -captures the main features and elements of this ZSM architecture.



The ETSI ZSM framework reference architecture

ZSM service aka management service: A set of offered management capabilities.

Management function: Logical entity playing the roles of service consumer and/or service producer.

Integration fabric: A management function, playing the roles of both service consumer and service producer, that enables interoperation and communication between management functions within and across management domains.

Cross-domain data services: Services that allow to share data with authorized consumers across domains.

Management domain: A scope of management delineated by a technological, business, administrative or other boundary.

E2E service management domain: A management domain specialized to manage E2E services.

(Source: ETSI GS ZSM002)

Figure 2.3 - ETSI ZSM framework reference architecture (Source: ETSI GS ZSM002 [14])

In early 2018, ETSI ISG ZSM launched a work item entitled “End-to-End management and orchestration of network slicing”. This work item, which will result in the publication of ZSM003 in early 2021 (still in a publicly available draft form at the time of writing) strives to specify requirements and management solutions for the zero-touch automation of provisioning, fault supervision and performance assurance activities on network slices, when deployed across multiple management domains. These requirements and management solutions are technology-agnostic, in the sense they can be applied on individual domains (e.g. access domain, transport domain, cloud domain), whatever their specificities are (e.g. mobile access vs fixed access in access domain, IP/MPLS network vs optical network for access transport, VM-based orchestration vs container-based orchestration). In pursuing this goal, the ETSI ISG ZSM:

- Identifies relevant SDOs working on network slicing specifications, shedding light on their individual scope (e.g. 3GPP for mobile access and core network slicing, IETF for transport network slicing, NFV for virtualized network slice).
- Leverages the on-going work in these SDOs, identifying existing gaps/inconsistencies across them and providing necessary means for their addressment. These means, based either on plug-in-based adaptations and model translations among domains, are supplied by ZSM cross-domain integration fabric.
- Proposes management exposure framework for network slicing scenarios, including support for "Network Slice as NOP internal" and "Network Slice as a Service (NSaaS)". The latter, whereby the network operator makes slicing management capabilities available to the verticals, assumes the possibility of having different levels of exposure. This is aligned with the 5G-VINNI vision documented in D3.1 [7].

2.1.3 ETSI NFV

The relationship between network slices and NFV network services was presented and thoroughly discussed in D1.3 [2]. This relationship, summarized in the statement "an NFV network service can be regarded as the resource-facing view of a network slice subnet, for the cases where at least one of the network slice subnet's network functions is deployed as a VNF", still prevails.

At the time of writing D1.3, ETSI NFV Release 3 had not ended yet. Now, with the launch of NFV Release 4 and the work carried out in the four defined technical areas (see D1.4 [3] for more details), new features aiming for a much more simplified management of VNFs, accompanied with a boost of their individual performance, can be applied to the deployment and operation of network slices. The features relevant for network slicing management can be grouped into two clusters:

- **Advanced built-in NFVI technologies**, in search of addressing the performance gap that today exists between VNFs (i.e. software images running on commodity hardware) and middleboxes (i.e. purpose-built hardware devices). This led to increased interest in hardware-acceleration technologies for VNFs using externally connected hardware devices, such as Graphic Processing Units (GPUs), Field Programmable Gate Arrays (FPGAs), Smart NIC or Network Processing Units (NPUs). Hardware accelerators and CPUs can be used in conjunction such that CPU-intensive tasks (e.g. security, packet processing) can be offloaded from VNFs to hardware-accelerators, and the rest of the VNF operations can be performed by the CPUs of general-purpose hardware. This approach not only allows achieving much more performant slices (i.e. increased KPIs on their user plane components), but also a more efficient resource usage (i.e. by freeing up more CPU cores that can now be dedicated to host new VNFs, from the same or other slices). Figure 2.4 captures the most relevant hardware accelerators, also illustrating their applicability to the corresponding use cases and scenarios.
- **Evolution towards cloud-native practices**. This evolution is articulated in three main workstreams: (i) design of lightweight and stateless VNFs, for an improved VNF scalability; (ii) evolving from VM-based orchestration (e.g. OpenStack) to a container-based orchestration (e.g. Kubernetes), when required to achieve higher agility in VNF operation; and (iii) application of Platform-as-a-Service (PaaS) capabilities [15], pursuing reusability and sharing of common services (e.g. monitoring, security services) across NS instances.

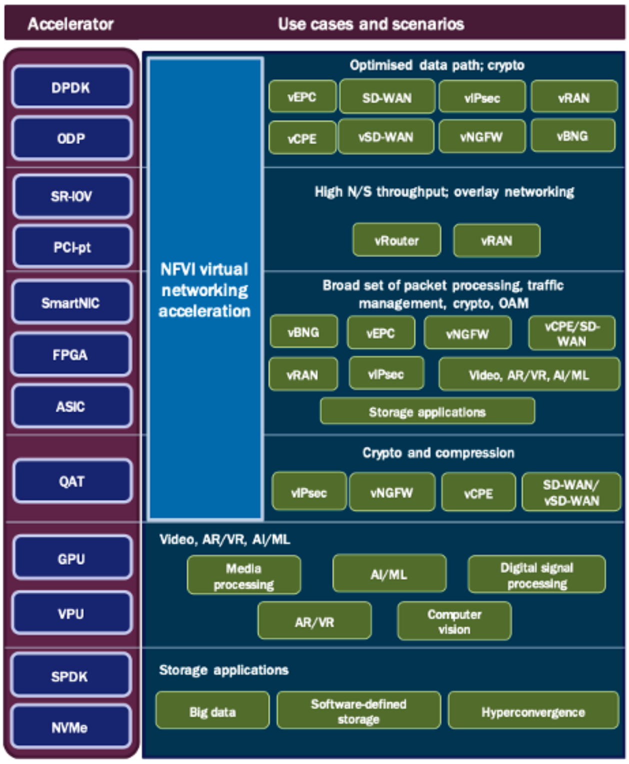


Figure 2.4 - Accelerators and use cases (Source: Analysis Mason, "Acceleration technologies: realizing the potential of network virtualization," [16])

These two clusters, when combined, allow for a much more flexible virtualization environment for the execution of multiple slices. 5G-VINNI facility operators might consider the inclusion of some of the above-referred features to transform their NFVI and MANO stacks in the mid-term, especially when hosting ICT-19 use-cases with demanding SLAs.

NFV has also launched a study item on "multi-tenancy support". The aim pursued by this study is similar to 3GPP Rel-17 eMEMTANE, but applied to Release 4 NFV MANO system. The outcomes of this study, which are to be published in the Technical Report NFV-IFA 018, may unveil mechanisms to ensure segregation of VNFM and NFVO services for different tenants[3], with the definition of separate tenant spaces. This may help 5G-VINNI facility to improve the enforcement of the four capability exposure levels defined in D3.1 [7].

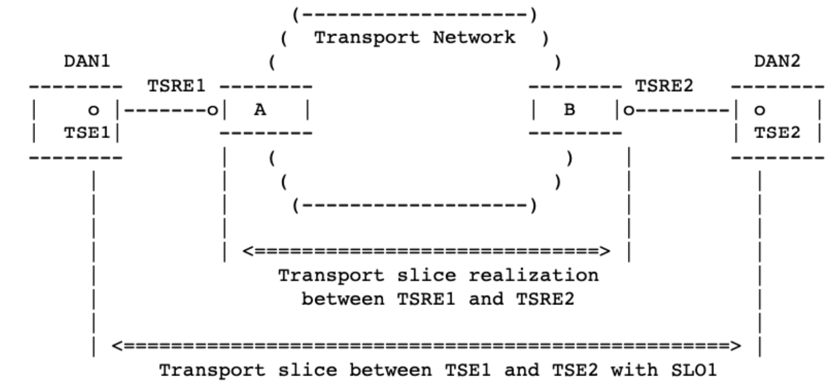
2.1.4 IETF

The Internet Engineering Task Force (IETF) has recently started working in network slicing enablers, developing specifications to fulfil the requirements of transport part of the end-to-end network slice, i.e. transport slice. These specifications are updated and maintained by the Network Slice Design Team (NSDT), a task force formed in the Traffic Engineering and Architecture Signaling (TEAS) WG. The mission of the NSDT is to develop a framework for providing transport network slicing using IETF traffic engineered technologies, e.g. IP, (G)MPLS, Segment Routing and other enhanced Virtual Private Network (VPN) solutions), IETF architecture solutions, e.g. Application-Layer Traffic Optimization (ALTO), Abstraction and Control of Traffic-Engineered Networks (ACTN), and IETF service delivery models, e.g. Layer Two Virtual Private Network Service Model (L2SM), Layer Three Virtual Private Network Service Model (L3SM). The expectation is for these IETF assets to be used to

[3] When 3GPP management system consumes NFV MANO services, the concept of 3GPP tenant may be mapped to NFV tenant concept.

create individual transport slices, each representing a specific, isolated, and managed logical network instance executed atop a common transport infrastructure.

According to NSDT scope, a transport slice is a logical network topology connecting several endpoints with a set of shared or dedicated network resources, which are used to satisfy specific Service Level Objective (SLO) [17]. These SLOs do not describe how the transport slices will be implemented or realized in the underlying network layers; instead, they are defined in terms of dimensions of operations (e.g. time, capacity), availability and other attributes. Examples of SLOs are guaranteed minimum bandwidth, guaranteed maximum latency, maximum permissible delay variation, maximum permissible packet loss rate and availability. This rationale allows establishing a clear demarcation point between traditional L2/L3VPNs, focused on segmentation (i.e. creation and management of private networks) and bound to a specific technology and traffic type, and transport network slices, concerned with the assurance of SLOs and unaware of underlying infrastructure connectivity. This technology-agnostic feature provides means for the network operator to flexibly decide on the realization of individual transport slices, depending on the technology and traffic engineering mechanisms available for use on the operator's managed transport network infrastructure.


Transport Network
DAN1
TSRE1
A
B
TSRE2
DAN2
TSE1
TSE2
Transport slice realization
between TSRE1 and TSRE2
Transport slice between TSE1 and TSE2 with SLO1

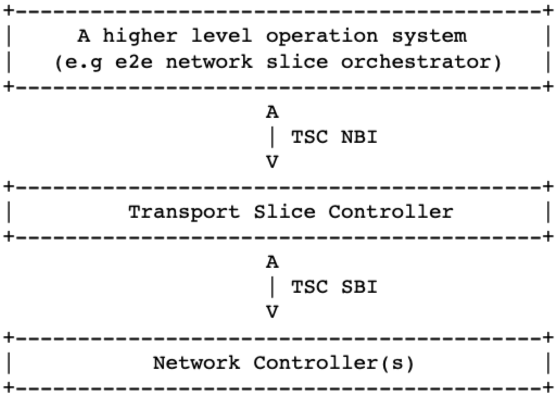
Legend:
DAN: Device, application and/or network function

Figure 2.5 - An IETF transport slice between TSEs and its realization between TSREs (Source: IETF Definition of Transport Slice [17])

Figure 2.5 -captures an example of a transport slice and its realization between multiple Transport Slice Endpoints (TSEs) - conceptual points of connection of a network function/device/application to the transport slice - and Transport Slice Realization Endpoints (TSREs) – mapping of TSEs into technology-specific nodes. For the provisioning of individual transport slices, [17] specifies a new management entity: the Transport Slice Controller (TSC). Conceptually equivalent to 3GPP/BBF referred T-NSSMF, the TSC is the entity in charge of realizing a transport slice in the network, maintaining and monitoring the run-time state of resources and topologies associated with it. For the interaction with upper/lower management systems, the TSC defines two interfaces:

- TSC Northbound Interface (NBI): it is the interface that allows the exposure of TSC capabilities to higher level operation systems e.g. 3GPP management system playing NS_MS producer and NSS_MS consumer roles like traditional NSMF. It is a technology-agnostic interface. Over this NBI, slice characteristics and other requirements can be communicated to TSC and the operational state of a transport slice may be requested.
- TSC Southbound Interface (SBI): it is the interface that allows TSC to interact with underlying network controller(s) e.g. IP/MPLS controller, optical controller, microwave controllers. These interfaces are technology-specific and leverage many of the existing network models.

Figure 2.6 illustrates the TSC interfaces and their interaction with other management systems. The resulting architectural framework can be easily mapped with the ACTN framework, as shown in [18].



```
+------------------------------------------+
|      A higher level operation system     |
|   (e.g e2e network slice orchestrator)   |
+------------------------------------------+
                     A
                     | TSC NBI
                     V
+------------------------------------------+
|         Transport Slice Controller       |
+------------------------------------------+
                     A
                     | TSC SBI
                     V
+------------------------------------------+
|           Network Controller(s)          |
+------------------------------------------+
```

Figure 2.6 - Architectural framework for the provision of transport slices. The core management entity is the Transport Slice Controller (TSC) (Source: IETF Definition of Transport Slice, IETF Draft [17])

Building upon the agreed transport slice and TSC concepts, the NSDT contributors are exploring solutions for transport network slicing in two separate workstreams: on modelling the TSC NBI and on technologies enabling transport slice realization. Table 2.2 summarizes the ongoing work in both workstreams.

Table 2.2 - NSDT workstreams and related Internet-drafts

NSDT workstream	Internet-drafts	Description
On modelling the TSC NBI	Considerations for defining a Transport Slice NBI [19]	This document analyses different use cases deriving the needs of potential transport slice customers in order to identify the functionality required on the TSC NBI to be exposed towards such transport slice customers.
	Transport Network Slice YANG Data Model [20]	This document describes a provider-centric YANG data model for the management and control of individual transport slices. This model can be used by the TSC operator to provision transport slices towards customers using the NBI.
	A YANG Data Model for Transport Slice NBI [21]	This document describes a customer-centric YANG data model for the management and control of individual transport slices. This model can be used by the TSC customers to request, configure and manage the components of individual transport slices.
On technologies enabling transport slice realization	Packet Network Slicing using Segment Routing [22]	This document presents a mechanism aimed at providing a solution for transport network slicing, based on the use of a unified administrative instance identifier to distinguish different virtual network resources for both intra-domain and inter-domain scenarios. Combined with the SR-TE, the mechanism can be used for both best-effort and TE services for

		tenants.
	Network Slicing with Flexible Traffic Engineering [23]	This document specifies procedures and signalling enhancement to Flexible Algorithm to ease provisioning and to scale it better via Flexible Traffic Engineering (FTE). FTE is an integration of Flexible Algorithm (FlexAlgo) and Segment Routing Traffic Engineering (SR-TE)

2.1.5 Broadband Forum

The purpose of the Broadband Forum (BBF) project SD-406 [24] is to investigate Transport Network Slicing Management (TNSM) from end-to-end perspective supported by the BBF Multi-service Broadband Network (MSBN) architecture. Transport network[4] slicing is considered as a fundamental enabler to migrate the MSBN architecture from "one architecture fits all" to the “logical network per service". The transport network slicing use cases can be organized into different types, these being;

- Network Slice as a Service. It enables an on-demand customized fixed broadband network resource leasing business model on the top of a common network infrastructure. The customized logical network can be modified dynamically to suit service demands.
- Supporting 5G related 3GPP Use Cases. From service management perspective, the identified requirements from 3GPP use cases to MSBN can be seen as a set of link requirements (e.g. topology, QoS parameters, etc.). Such link requirements are communicated to the transport network in order to support connectivity between the 3GPP RAN and/or core networks nodes that belong to the network slice instance, while the 3GPP management system configures the corresponding 3GPP nodes to use such links.
- Slicing across Fixed-Mobile Converged Networks. A Fixed-Mobile Converged (FMC) network slice is built on the top of SD-407 [25] by combining resources from both fixed and mobile, i.e. 3GPP, networks, with optimization of service provision and availability by offering various degrees of deterministic performance in terms of throughput, latency, resiliency, etc. As shown in Figure 2.7, the processes and operations of Service Management and Network Slice Control supported by MSBN requires a continuous process capable to analyse the service requirements and assure the desired performance even when the conditions of the network change or the requirements from the customer perspective evolve with time.

[4] The use of the term “Transport Network” is not fully consistent among different SDOs, e.g. 3GPP, IETF, ETSI, BBF, MEF. Section 4.5.1 of 5G-VINNI D1.3 [2] provides a general overview of the approaches followed by different SDOs in relation to transport networks in the context of 5G. In this section, the meaning of the term is supposed to be aligned with the BBF vision, which is mainly focused on the fixed access and aggregation network segments.

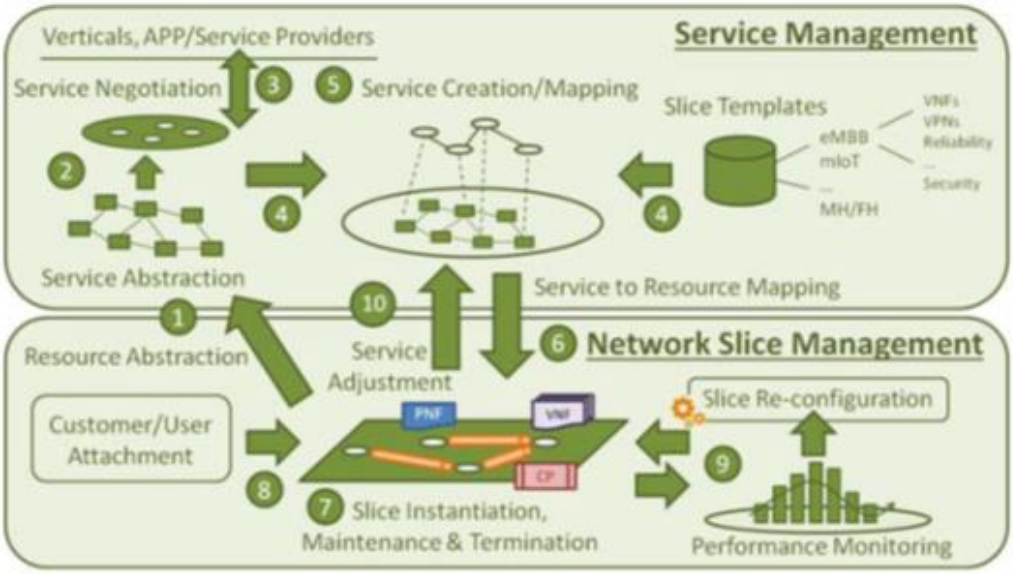


Figure 2.7 - MSBN service management and network slice management processes and operations (Source: Broadband Forum, "SD-406 End-to-End Network Slicing," [24])

From a BBF perspective, in the context of transport networks, Network Slice Management combined with Service Management can be regarded as the transport management domain, which provides capabilities such as service abstraction, service negotiation, service operations, service adjustment, and service template to verticals, application/service providers and 3rd parties for end-to-end service management. The Transport Network Slice Management (TNSM) documented in [24] and [2] takes care of the slice lifecycle management of the transport network Sub-Network Slice Instance (S-NSI) and provides the capability exposure of the transport network via Mobile-Transport Network Slice Interface (MTNSI) to the 3GPP mobile network, i.e. towards the network slice management function. It also provides the mapping of the 3GPP mobile network requirements to the corresponding transport network. The Transport Network Slice Management in BBF includes service management and network slice orchestration aspects considering the lifecycle management operations, service exposure, and interaction with mobile network and multi-administrative domain support.

2.1.6 MEF

Network automation with MEF LSO helps reduce management complexity so that operators can achieve the speed and efficiency they need to make 5G network slicing economical. Network automation is a key ingredient to enable reduction of costs, both CAPEX and OPEX.

MEF LSO provides APIs to automate the entire lifecycle of services orchestrated across multiple provider networks and multiple technology domains within a provider network [26]. The LSO reference architecture, shown in Figure 2.8, characterizes the management and control domains (e.g., Service Provider (SP) and Partner) and functional management entities (e.g., Business Applications) that enable inter-provider orchestration. The architecture also identifies the management interface reference points (e.g., LSO Sonata) which are the logical points of interaction between specific functional management entities. These management interface reference points are further defined by interface profiles and implemented as APIs. This is a functional architecture and does not describe how the functional management entities are implemented (e.g., single vs. multiple instances), but rather identifies functional management entities that provide logical functionality as well as the points of interaction among them.

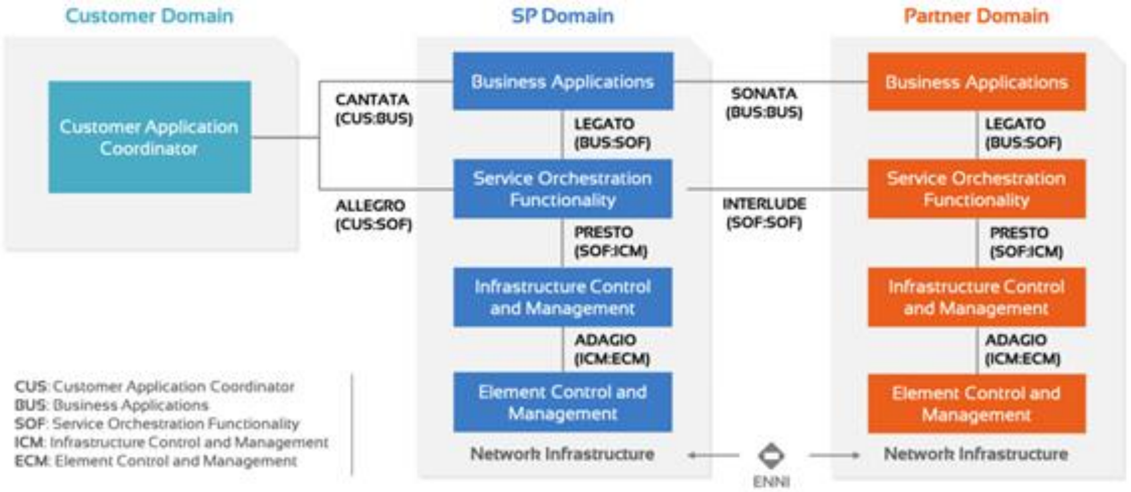

Customer Domain
SP Domain
Partner Domain
Customer Application Coordinator
CANTATA (CUS:BUS)
ALLEGRO (CUS:SOF)
Business Applications
LEGATO (BUS:SOF)
Service Orchestration Functionality
PRESTO (SOF:ICM)
Infrastructure Control and Management
ADAGIO (ICM:ECM)
Element Control and Management
Network Infrastructure
SONATA (BUS:BUS)
INTERLUDE (SOF:SOF)
ENNI
Business Applications
LEGATO (BUS:SOF)
Service Orchestration Functionality
PRESTO (SOF:ICM)
Infrastructure Control and Management
ADAGIO (ICM:ECM)
Element Control and Management
Network Infrastructure
CUS: Customer Application Coordinator
BUS: Business Applications
SOF: Service Orchestration Functionality
ICM: Infrastructure Control and Management
ECM: Element Control and Management

Figure 2.8 - MEF LSO Reference Architecture (Source: "MEF 55; Lifecycle Service Orchestration (LSO): Reference Architecture and Framework," [26]

In LSO, services are orchestrated by a Service Provider across all internal and external network domains from one or more network operators. These network domains may be operated by, among others, Communication Service Providers (CSPs), data center operators, enterprises, wireless network operators, virtual network operators, and content providers. LSO spans in a federated approach all those network domains that require coordinated management and control to deliver end-to-end services. The LSO Cantata interface is used for business-related interactions such as ordering and billing between the Customer and the Service Provider, and LSO Sonata is used for similar business-related interactions between Service Providers. The LSO Allegro interface is used for configuration and control-related management interactions that are allowed by the respective service agreement such as operational state queries, request up-dates to service parameters, or requests to instantiate other services. The LSO Presto interface is used for orchestrating within the Service Provider domain at the network level and the LSO Adagio interface correspondingly orchestrates at the resource level.

LSO orchestration of transport slices is an important factor to consider in the context of supporting 3GPP 5G network slices. MEF LSO functions are correlated to 3GPP network slice and subnetwork slice management functions as follows:

- The MEF end-to-end Service Orchestration Function (SOF) is referenced by 3GPP as the E2E Network Slice Management Function (NSMF)
- MEF network domain controllers for RAN, transport and core network domains which provide Infrastructure Control and Management functions (ICM) are referenced by 3GPP as the Network Slice Subnet Management Function (NSSMF)

3GPP has defined the interfaces for the NSMF to communicate with both the RAN NSSMF and core NSSMF. However, 3GPP has not, as yet, defined the same interface for the transport domain. In this use case the MEF LSO Presto interface reference point (SOF:ICM) is applicable for the NSMF to communicate with the transport NSSMF.

In support of services on 3GPP-defined 5G E2E network slices, transport slices provide the corresponding mobile transport networks. These enable consistent operational practices and automation on less complex networks, thus accelerating service delivery. Transport slices also enable different endpoints with specific Service Level Specifications (SLSs) to be connected using a multitude of types of shared or dedicated network resources with differing levels of isolation. There is a need for flexibility in implementing transport slices to support the delivery of 5G services across mobile transport networks consisting of products from multiple vendors, multiple domains and using various transport network technologies, tunnel types (e.g., ODU/OCh, Ethernet, IP, MPLS, segment routing) and MEF Service types (e.g., Optical transport, Carrier Ethernet, IP VPN). This implementation flexibility enables support for a wide range of E2E 5G deployment scenarios and use cases, including for 4G/5G hybrid networks. For example, in Figure 2.9, a single Service Provider is both the MNO and the transport network provider. The E2E 5G network deploys an E2E network slice composed of a RAN subnetwork slice, three transport slices and a core subnetwork slice. The transport slices enable the transport connectivity between network elements in the RAN and core subnetwork slices across low latency Mobile Fronthaul (Low Layer Split - LLS) (slice 1 with blue connections), high latency Mobile Fronthaul (High Layer Split - HLS) (slice 2 with yellow connections) and Mobile Backhaul (slice 3 with red and green connections). Transport slicing may also be applied from the 5G core to public networks or clouds.

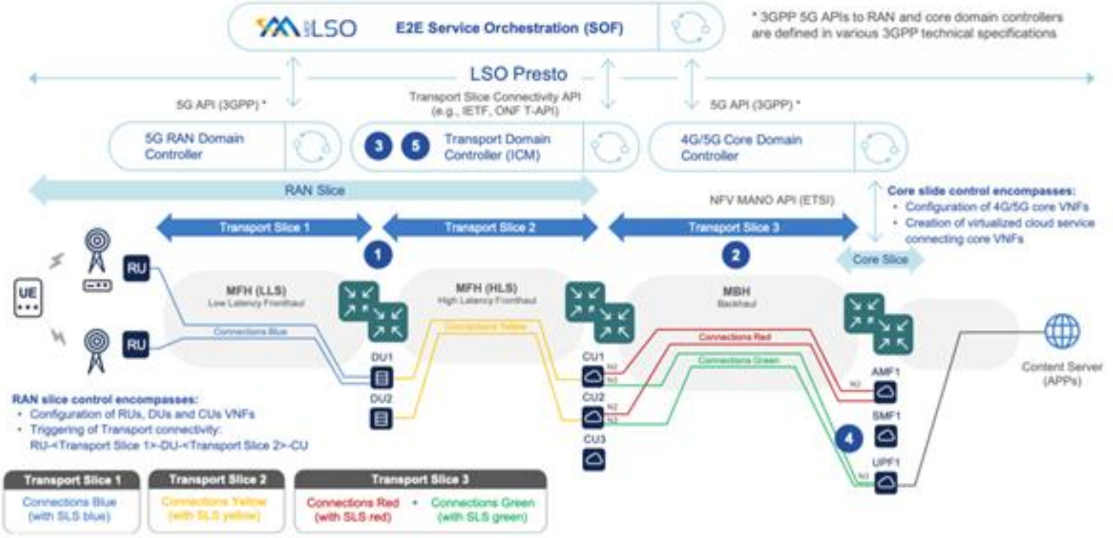

LSO
E2E Service Orchestration (SOF)
* 3GPP 5G APIs to RAN and core domain controllers are defined in various 3GPP technical specifications
LSO Presto
5G API (3GPP) *
Transport Slice Connectivity API (e.g., IETF, ONF T-API)
5G API (3GPP) *
5G RAN Domain Controller
3
5
Transport Domain Controller (ICM)
4G/5G Core Domain Controller
RAN Slice
NFV MANO API (ETSI)
Core slide control encompasses:
• Configuration of 4G/5G core VNFs
• Creation of virtualized cloud service connecting core VNFs
Transport Slice 1
Transport Slice 2
Transport Slice 3
1
2
Core Slice
UE
RU
RU
MFH (LLS)
Low Latency Fronthaul
MFH (HLS)
High Latency Fronthaul
MBH
Backhaul
Connections Blue
Connections Yellow
Connections Red
Connections Green
DU1
DU2
CU1
CU2
CU3
N2
N3
AMF1
SMF1
UPF1
4
Content Server (APPs)
RAN slice control encompasses:
• Configuration of RUs, DUs and CUs VNFs
• Triggering of Transport connectivity:
RU-<Transport Slice 1>-DU-<Transport Slice 2>-CU
Transport Slice 1
Connections Blue (with SLS blue)
Transport Slice 2
Connections Yellow (with SLS yellow)
Transport Slice 3
Connections Red (with SLS red) + Connections Green (with SLS green)

Figure 2.9 - E2E 5G services support using 3GPP and MEF LSO for mobile transport domain (Source: MEF, "Slicing for Shared 5G Fronthaul and Backhaul," [27])

The E2E network slice is orchestrated by the E2E Service Orchestration Function (SOF) using the RAN, core and transport domain controllers via APIs at the MEF LSO Presto reference point. The domain controllers can expose APIs at LSO Presto that can be implemented compatibly with relevant standards (e.g., 3GPP, bETSI-NFV, bvIETF, ONF T-API [28], MEF NRM [29]).

2.2 Industry and Open Source initiatives

2.2.1 GSMA

The GSM Association (GSMA) is a telco industry association representing the interests of mobile operators and technology providers worldwide, uniting nearly 800 operators with almost 300 companies in the broader ecosystem. A key recognized activity of GSMA is to collect information on service requirements and regulatory issues from different vertical industry associations (e.g. 5G-ACIA, 5GAA), identify potential technologies that can satisfy these requirements, and inform corresponding SDOs (e.g. 3GPP, ETSI, IETF), so that they can develop corresponding technology solutions. One of the key technologies in this regard is network slicing.

GSMA vision on network slicing was first presented in [30]. This document was followed by [31], where GSMA provided a comprehensive overview about the service requirements on network slicing expressed by business customers from different vertical industries, including AR/VR, automotive, energy, healthcare, manufacturing (I4.0), LPWA, public safety, smart cities, etc. From the analysis conducted in [31], GSMA noted that service requirements on network slicing could be classified into performance, functional and control and management requirements. However, it concluded that there was no agreement on how vertical industries should express these requirements towards network operators. In this regard, GSMA agreed on the need to harmonize network slicing definition, identify network slice types with distinct characteristics and consolidate parameter and functionality requirements, from end-to-end perspective.

GSMA work on network slicing management is organized into two main workstreams, both led by the GSMA Networks Group (NG).

The first workstream has the target to map service requirements from vertical industry use cases into network slice requirements. Based on the conclusions from [31], GSMA suggested that it was necessary to develop a solution able to offer guidelines to vertical industries on how to issue service requirements on network slicing towards network operators, therefore addressing the existing gap between vertical and telco industries. To that end, the Generic (network) Slice Template (GST) has been defined and documented in GSMA PRD NG.116 [32]. The GST provides a universal description of a network slice containing all the potential attributes that can be used to characterize one slice separately from another. It allows the network slice provider (e.g. network operator) and network slice customer (e.g. industry vertical) to agree on the Service Level Specification (SLS) for a network slice, by means of filling GST attributes with values based on service requirements. The result of this mapping defines a Network Slice Template (NEST), which in essence is a filled-in version of the GST that allows characterizing a network slice based on a service type. Different NESTs allow describing different types of network slices. On the one hand, for slices based on 3GPP 5G service categories (e.g. eMBB, mIoT, uRLLC and V2X), the operator may have a set of standardized NESTs (S-NEST). On the other hand, for slices addressing specific industry use cases (e.g. industry 4.0 use case, logistics use case, eHealth use case), the operator can define additional private NESTs (P-NESTs). Both S-NESTs and P-NESTs are registered and published in the operator's service catalogue. Figure 2.10 captures the concept of GST and NESTs as defined in the GSMA.

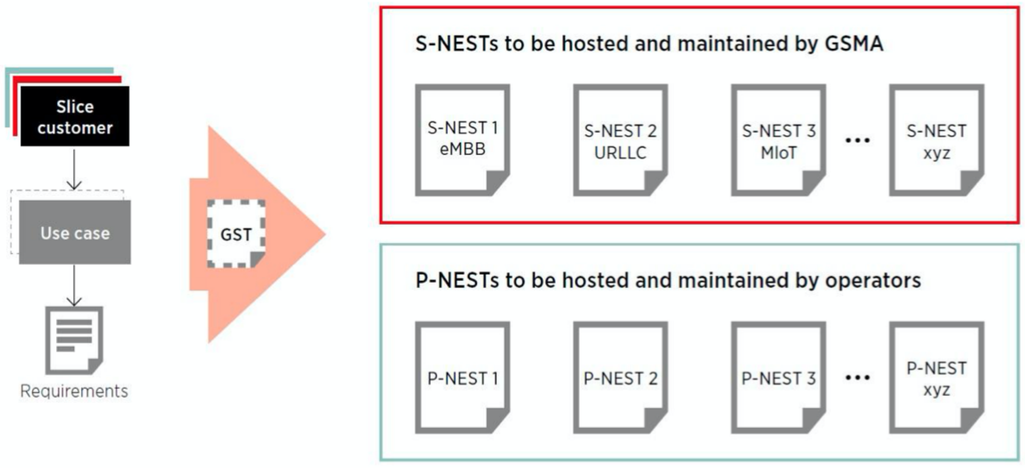


Figure 2.10 - S-NEST and P-NEST guidance from the GSMA (source: GSMA, "From Vertical Industry Requirements to Network Slice Characteristics," [33])

The 5G-VINNI project has kept a close eye on the GST progress since the very beginning. The concept of 5G-VINNI Service Blueprint (VINNI-SB), documented in D3.1 [7] and further elaborated in D3.2 [34], leverages on attributes from the GST v1.0. Based on these attributes, four 5G-VINNI Service Descriptors (VINNI-SDs) have been defined: three corresponding to S-NESTs (eMBB, uRLLC and mIoT VINNI-SDs), and other with a P-NEST (customized VINNI-SDs). 5G-VINNI has also been actively involved in the GSMA NG work on GST maintenance, providing feedback on plausible GST attribute values for some of the use cases documented in [7], which has ultimately resulted in the refinement of the GST specification in later versions, including GST v2.0 (2019) and v3.0 (2020).

The second workstream has the target to provide a deep E2E network slice architecture analysis, with the mission of identifying existing gaps and informing corresponding SDOs, fostering cross-standardization collaboration to address these gaps. This activity, initiated in October 2020, will last until June 2021, when the publication of a white paper is expected. This white paper aims at reflecting the operators' agreed vision on E2E slicing, so that technology providers (e.g. vendors, system integrators, SMEs) can develop necessary solutions fitting operator needs.

2.2.2 O-RAN Alliance

The O-RAN Alliance [35] is a world-wide, carrier-led effort to drive new levels of openness in the radio access network of next-generation wireless systems. It was created in 2018 as a result of the merger of two previously existing initiatives – the C-RAN Alliance and the xRAN Forum. The key principles of the O-RAN Alliance include [36]:

1. Leading the industry towards open, interoperable interfaces, RAN virtualization, and big data and AI enabled RAN intelligence;
2. Maximizing the use of common-off-the-shelf hardware and merchant silicon and minimizing proprietary hardware;
3. Specifying APIs and interfaces, driving standards to adopt them as appropriate, and exploring open source where appropriate.

The O-RAN architecture is based on well-defined, standardized interfaces that are compatible with 3GPP to enable an open, interoperable RAN. The logical architecture, including the relevant interfaces, is represented in Figure 2.11, O-RAN is responsible for defining and maintaining the interfaces A1, O1, O2, E2 and the Open Fronthaul interface. Other interfaces represented in the figure, namely E1, F1-c, F1-u, NG-c, NG-u, X2-c, X2-u, Xn-c, Xn-u, Uu, are defined and maintained by 3GPP, but can be seen also as part of the O-RAN architecture.

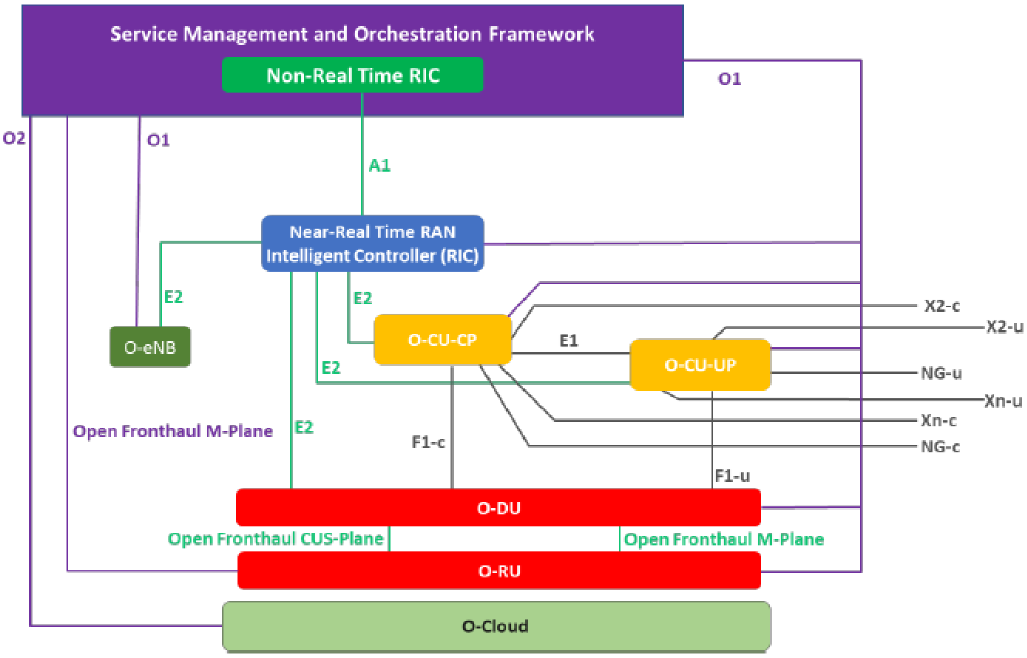


Figure 2.11 - O-RAN logical architecture (Source: O-RAN Alliance, "O-RAN Working Group 1 Slicing Architecture," [37])

One of the main building blocks of the O-RAN architecture is the RAN Intelligent Controller (RIC), which consists of a Non-Real-time Controller (non-RT RIC), handling tasks that can tolerate latency above 1 s (e.g. optimization of RAN elements and resources, service and policy management, RAN analytics and model-training for the Near-RT RAN), and a Near-Real Time controller (near-RT RIC), for latency lower than 1s (e.g. near-real-time control and optimization of O-RAN elements and resources via fine-grained data collection and actions over E2 interface). Essentially, RIC functionality delivers intelligence into the Open RAN network with near-RT RIC functionality providing real-time optimization for mobility and handover management, and non-RT RIC providing not only visibility into the network, but also AI-based feeds and recommendations to near-RT RIC, working together to deliver optimal network performance for optimal subscriber experience [38].

O-RAN has also addressed the topic of slicing. O-RAN reference slicing architecture is described in [37]. It includes slice management functions which essentially correspond to 3GPP defined NSMF and NSSMF, enhanced with extensions for O-RAN network functions. Figure 2.12 represents the O-RAN slicing architecture and the main building blocks.

In O-RAN slicing architecture the fundamental role of the Non-RT RIC is to gather long term slice related data through interaction with the Service and Management Orchestration (SMO) framework and apply AI/ML based approaches interworking with the Near-RT RIC to provide innovative RAN slicing use cases. For this purpose, Non-RT RIC should be aware of RAN slices and their respective SLAs through SMO. In addition, Non-RT RIC may retrieve enrichment information from 3rd party applications enabling advanced RAN slicing technology to be applied in O-RAN framework.

Near-RT RIC enables near-real-time RAN slice optimization through execution of slicing related xApps (applications designed to run on the near-RT RIC, which may be provided by third parties) and communicating necessary parameters to O-CU and O-DU through the E2 interface. Deployed xApps may utilize either AI/ML based models or other control schemes which can further be guided by A1 policies that are generated by Non-RT RIC.

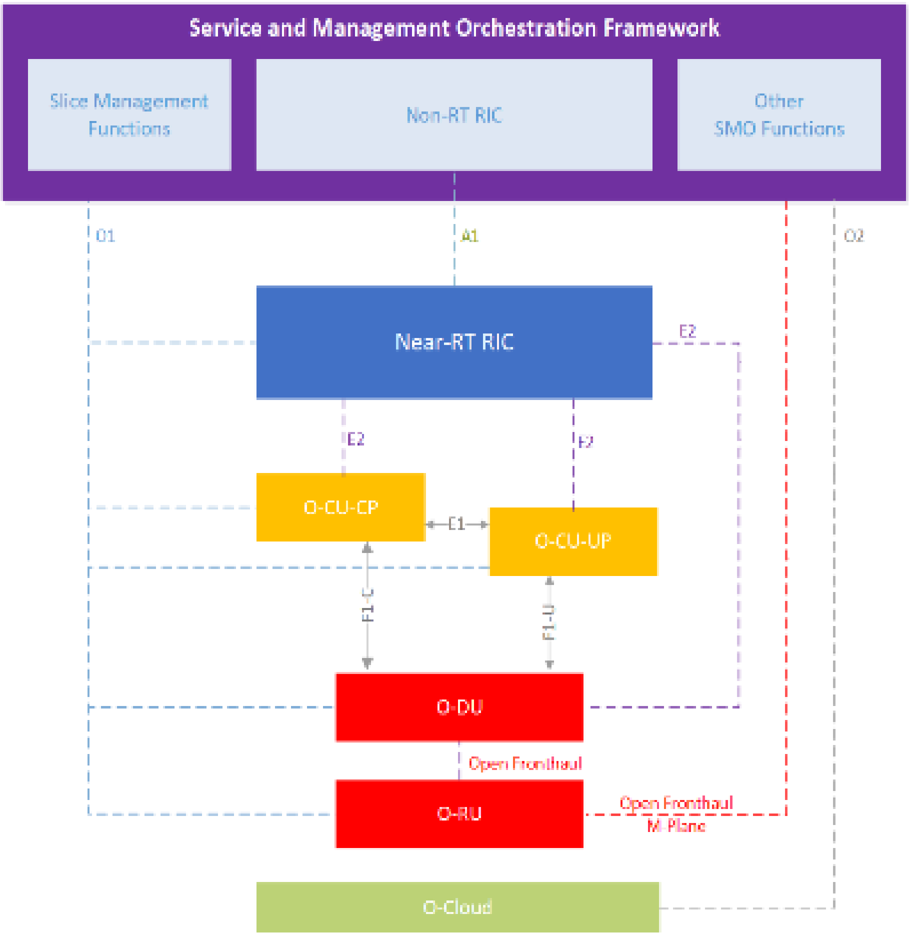


Figure 2.12 - O-RAN slicing architecture (Source: O-RAN Alliance, "O-RAN Working Group 1 Slicing Architecture," [37])

2.2.3 Open Source MANO (OSM)

Open Source MANO (OSM) is an ETSI-hosted project to develop an open source Management and Orchestration (MANO) stack aligned with ETSI NFV information models. The scope of OSM project covers both design-time and run-time aspects related to service delivery for telco service provider environments.

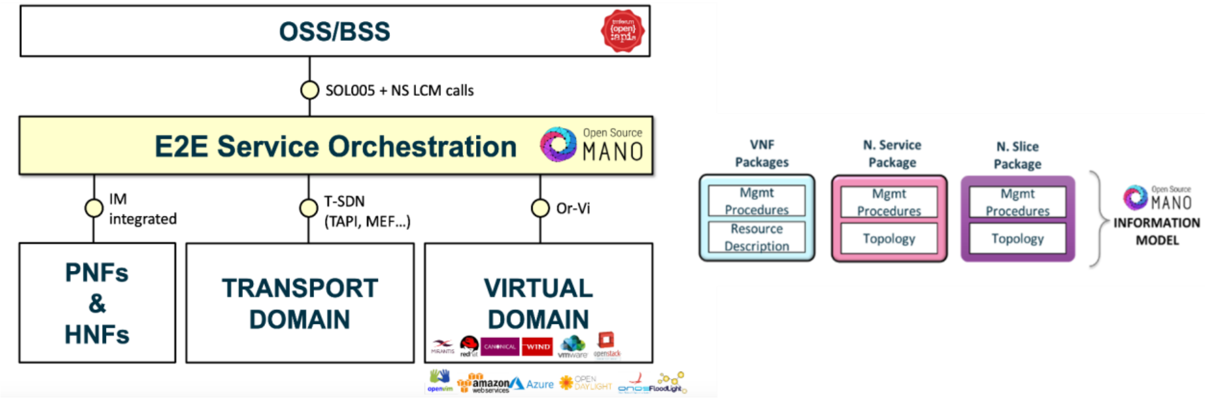


Figure 2.13 - OSM Reference Architecture (Source: “OSM#9 Hackfest, Hack 0: Introduction to NFV and OSM” [39])

The OSM Reference Architecture is presented in Figure 2.13. The scope of OSM includes the ability to orchestrate E2E network services and slices across virtual domains, network domains, as well as physical and hybrid network elements. For the interaction with external systems, OSM is imbued with:

- A unified NorthBound Interface (NBI). OSM uses the NBI to expose management capabilities to other network and service management systems (e.g. 3GPP management system), collectively represented as OSS/BSS in Figure 2.13. OSM’s NBI provides a superset of ETSI NFV SOL005 [40] APIs together with the ability to handle network slices from a resource management viewpoint. This viewpoint allows modelling a network slice as a composition of individual network slice subnets, each deployed as an exclusive or shared network service.
- A number of SouthBound Interfaces (SBI), to interact with underlying infrastructure resources. These SBIs include plugins towards virtual and transport domains (i.e. VIM, WIM and SDN controller plugins) as well as configuration interfaces towards individual non-virtualized network functions, i.e. Physical Network Functions (PNF) and Hybrid Network Functions (HNF).

OSM allows different modes to control and manage the lifecycle of network slice instances. In the **full E2E management mode**, OSM takes the full control over individual instances, managing them from their commissioning (instantiation + day-1 configuration) to their de-commissioning. In the **stand-alone management mode**, a 3rdparty standalone slice manager takes the role of managing slices via the OSM exposed SOL005 APIs, with OSM acting as NFVO. These two different management modes reflect aspects of the OSM capability exposure.

An overall description of OSM capabilities was provided in D1.3 [2]. At that point, the latest OSM version was Release FIVE. OSM community has kept the six-month cycles of releases, with OSM Release EIGHT being the current OSM version. Figure 2.14 shows the OSM Release EIGHT architecture. [41] provides a detailed description of the gold nuggets in this new Release. Among them, two features deserve to be mentioned, due to their impact on OSM network slicing management capabilities: the **definition of the Placement optimization module (PLA), and the quotas management functionality.**

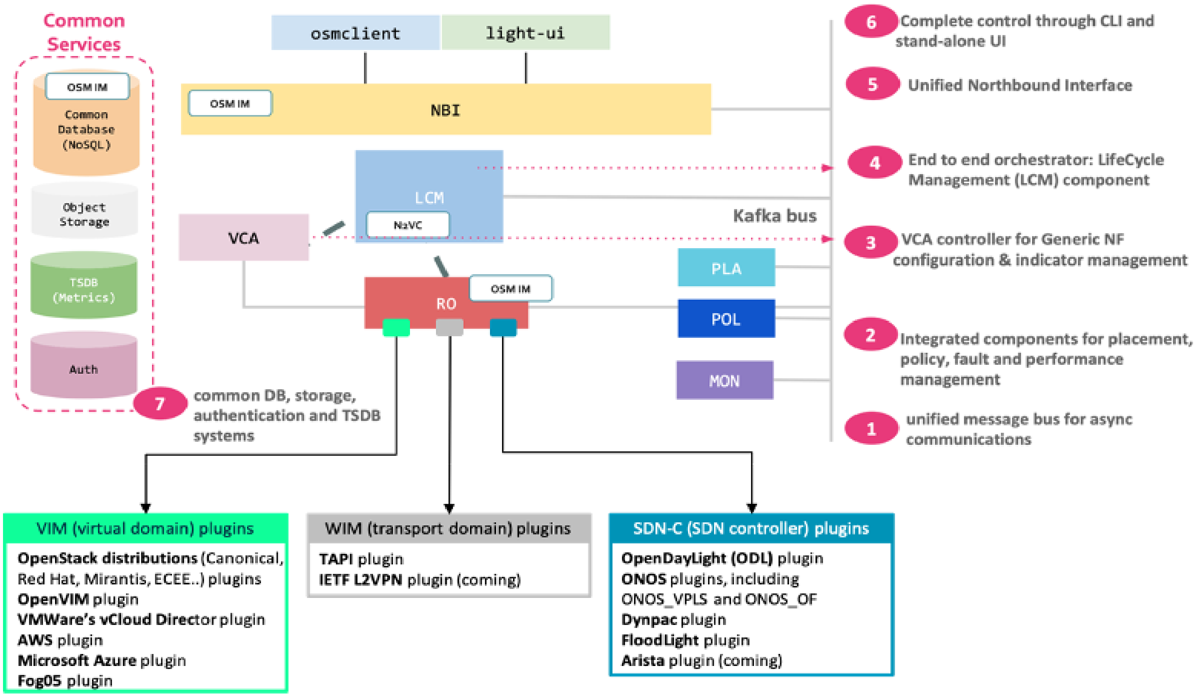
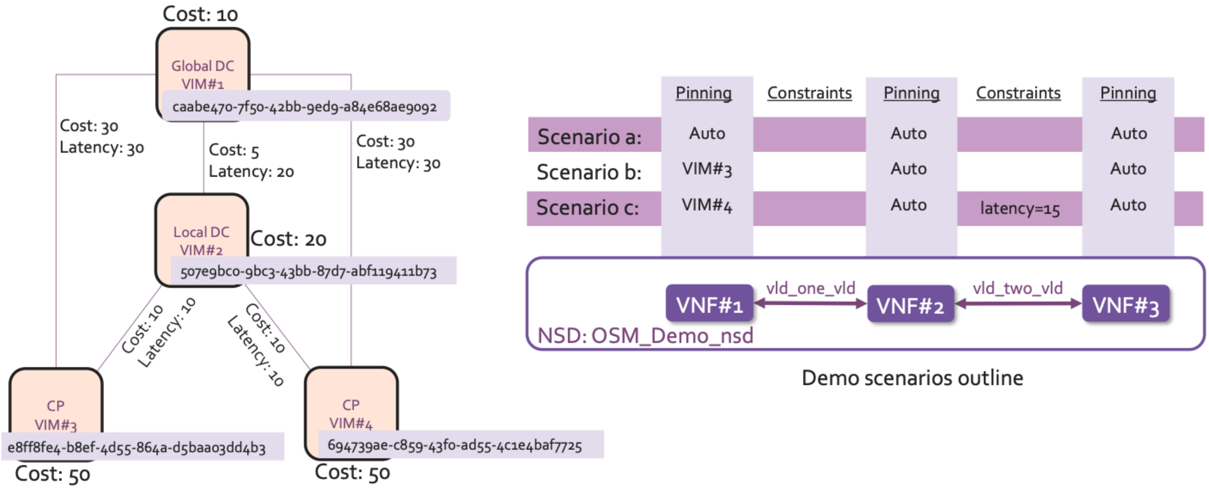

Common Services
OSM IM
Common Database (NoSQL)
Object Storage
TSDB (Metrics)
Auth
osmclient
light-ui
OSM IM
NBI
LCM
N2VC
VCA
Kafka bus
RO
OSM IM
PLA
POL
MON
7 common DB, storage, authentication and TSDB systems
6 Complete control through CLI and stand-alone UI
5 Unified Northbound Interface
4 End to end orchestrator: LifeCycle Management (LCM) component
3 VCA controller for Generic NF configuration & indicator management
2 Integrated components for placement, policy, fault and performance management
1 unified message bus for async communications
VIM (virtual domain) plugins
OpenStack distributions (Canonical, Red Hat, Mirantis, ECEE..) plugins
OpenVIM plugin
VMWare's vCloud Director plugin
AWS plugin
Microsoft Azure plugin
Fog05 plugin
WIM (transport domain) plugins
TAPI plugin
IETF L2VPN plugin (coming)
SDN-C (SDN controller) plugins
OpenDayLight (ODL) plugin
ONOS plugins, including ONOS_VPLS and ONOS_OF
Dynpac plugin
FloodLight plugin
Arista plugin (coming)

Figure 2.14 - OSM Release EIGHT architecture

On the one hand, the definition of the PLA enriches OSM network slicing management capabilities at instantiation time. This module helps the OSM user to find an optimal deployment of network slice, distributing the individual VNFs over the set of available VIMs. This distribution should be done according to optimization criteria, based on user-provided models of (i) compute and networking cost, and/or (ii) latency and jitter metrics of inter-VIM connectivity. Optimal placement of VNFs over the VIMs is done by matching network slice specific requirements to infrastructure availability and metrics, while considering cost of compute and networking. Figure 2.15 shows an example on how the PLA module can be applied for a particular network service setup.



	Pinning	Constraints	Pinning	Constraints	Pinning
Scenario a:	Auto		Auto		Auto
Scenario b:	VIM#3		Auto		Auto
Scenario c:	VIM#4		Auto	latency=15	Auto

Figure 2.15 - Network service scenario for PLA-based allocation. This scenario is part of the demo presented in the OSM Hackfest 8 (OSM-MR#8)

On the other hand, the quotas management functionality allows setting limits for the infrastructure (number of VIM, WIM, SDN controllers, Kubernetes clusters), packages (VNFDs, NSDs, NSTs) and deployed instances (network service and network slice instances). Once the limit is reached, any attempt to create a new item will be rejected to prevent overloading the system. This feature enhances security and provides a more granular control of OSM usage, which is key for multi-tenancy support. OSM can support multi tenancy environment by provide separate projects for different OSM clients. The project information contains the quotas assigned to the corresponding tenant. These quotas can be changed on-demand, by simply re-configuring the project settings at run-time.

In OSM Release SEVEN, the OSM framework started to support CNFs by leveraging already-deployed or new K8s clusters. Since Release EIGHT, OSM has evolved and supports K8s in the following levels:

- **Underlying technology to run OSM framework**: OSM is deployed using Docker images and containers, which may be instantiated as part of a K8s cluster.
- **Underlying technology to run CNFs**: OSM has the capabilities to run, deploy, orchestrate and manage K8s-based Network Functions (KNFs – Slight variation of the term "CNF" aiming at CNFs which run with K8s) and their LCM in a K8s cluster.
- **K8s proxy charms**: Prior to Release EIGHT Day-1 and Day-2 configuration operations were performed using Juju Proxy charms that were deployed on a LXD cloud. Currently, OSM allows running those operations through proxy charms deployed in K8s instead of LXD with the consequent reduction of the overall deployment time and the self-healing benefits inherent to K8s deployments.

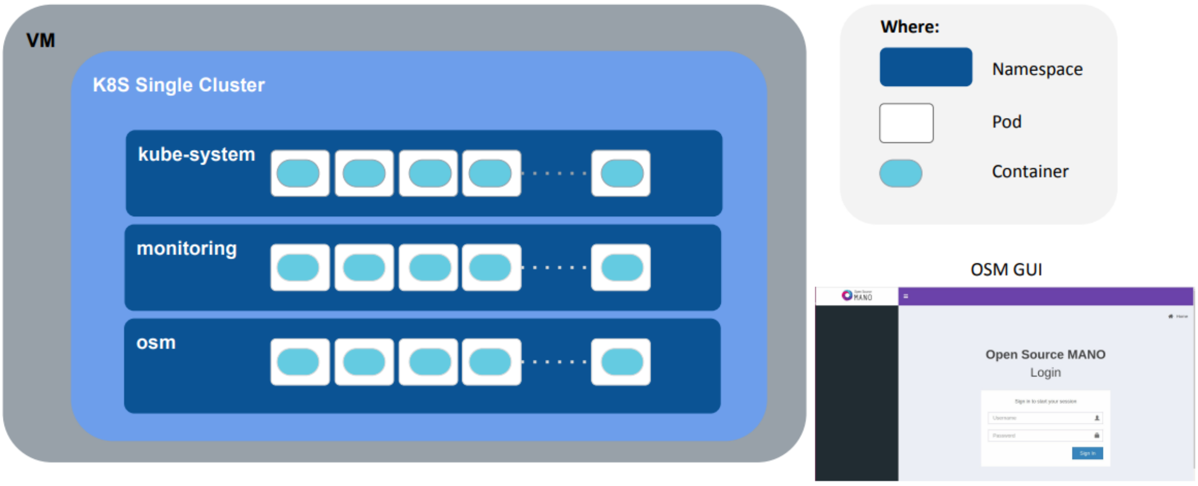


Figure 2.16 - OSM EIGHT framework using a K8s cluster (Source: OSM#9 Hackfest, "Hack 1: Architecture & Installation" [42])

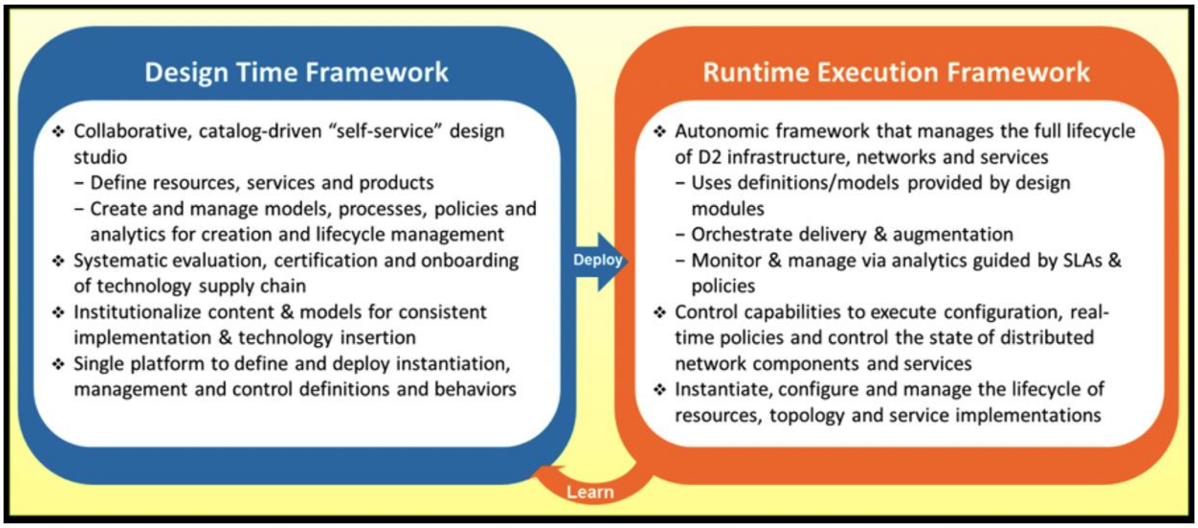
OSM Rel. EIGHT can be deployed on a single host as a K8s cluster, although the default option is to use Docker Swarm. As it can be seen in Figure 2.16, this installation method creates a single K8s cluster that is composed of three different namespaces:

- **Kube-system:** Contains the required functional K8s pods.
- **Monitoring:** Is comprised of the different pods that will enable the monitoring capabilities in the framework.
- **OSM:** Contains OSM functional pods.

2.2.4 Open Network Automation Platform (ONAP)

ONAP [43] is a framework developed as an Open Source Project supported by the Linux Foundation, created in March 2017 with the goal of creating an Open Source, Unified, Multi-vendor project which would manage and automate the network (VNF/PNF). This framework was originally conceived and release from two Service Providers solutions, E-Comp (AT&T) and Open-O (CMCC), were merged.

One of the main requirements of this solution is to allow a service to be created and launched without any disruption, which would support different technologies (e.g. 5G; LTE/4G). In order to fulfil such a requirement there is a separation of Service Design Time and Run Time, making use of existing Service Catalogues.



- ❖ Collaborative, catalog-driven "self-service" design studio
  - – Define resources, services and products
  - – Create and manage models, processes, policies and analytics for creation and lifecycle management
- ❖ Systematic evaluation, certification and onboarding of technology supply chain
- ❖ Institutionalize content & models for consistent implementation & technology insertion
- ❖ Single platform to define and deploy instantiation, management and control definitions and behaviors

- ❖ Autonomic framework that manages the full lifecycle of D2 infrastructure, networks and services
  - – Uses definitions/models provided by design modules
  - – Orchestrate delivery & augmentation
  - – Monitor & manage via analytics guided by SLAs & policies
- ❖ Control capabilities to execute configuration, real-time policies and control the state of distributed network components and services
- ❖ Instantiate, configure and manage the lifecycle of resources, topology and service implementations

Figure 2.17 - Service Design Time and Runtime Execution Frameworks in ONAP (Source: "ONAP Architecture Overview" Invalid source specified.**)**

The Service Design & Creation component enables the capability of a Service Provider to create and define services and resources, ultimately supporting different use cases [44]. Nevertheless, the ONAP community still consider a service as Network Service without the level of abstraction defined in ETSI, which is why most of the use cases defined for usage in an ONAP framework are network related ones.

Service Design time

The user interface shall allow the creation of new services, making use of existing ones from a catalogue. It is expected that this module will make available:

- Resources: VNF, PNF, IP elements, network connectivity
- Price Model (e.g. possibility of creating a new one)

It shall be possible to test a new service without the need to launch it. A similar behaviour to creating a sandbox is expected, where a user can test a network change or an application launch isolating the environment from a disastrous launch.

Service Run Time

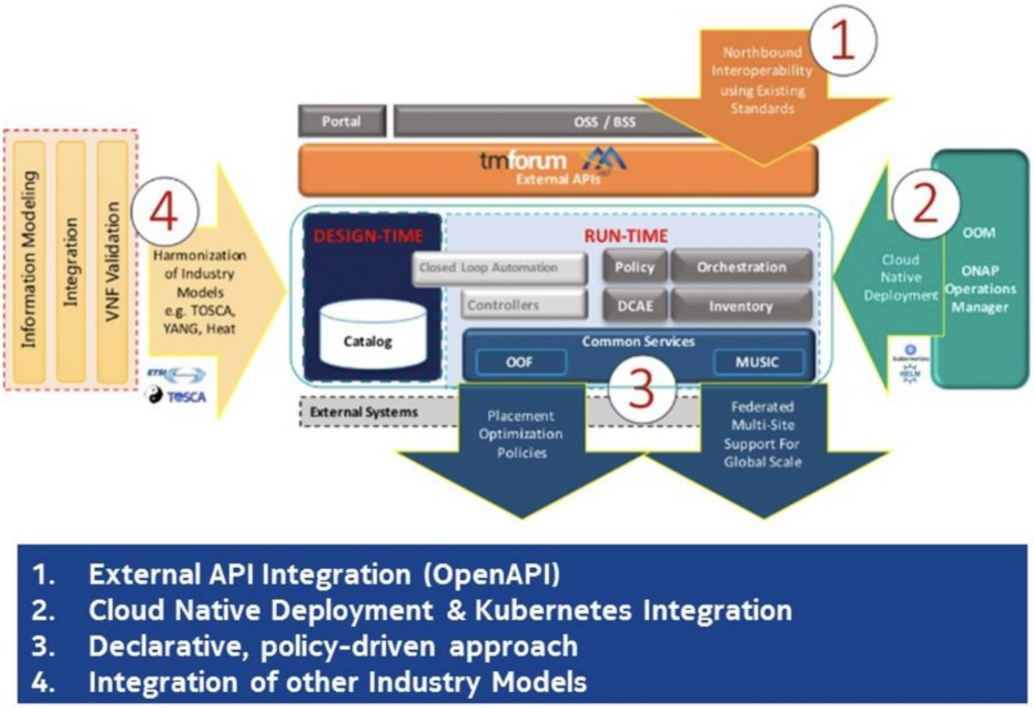
Once a service is available, it can be selected either manually or automatically, and then placed on a runtime environment (again without start/stop processes actions), capable of executing its service rules and policies. In doing this, the solution will benefit the customer, by providing a flexibility that was not achieved by others before.

The runtime framework needs to be isolated and guarantee a 5 nines availability. Therefore, an endless monitoring loop runs in real-time, making use of Analytics and respective SLA to guarantee customer service availability. This can be extended by offering (new) services to customers once behavioural patterns are identified.

Furthermore, the Orchestration function plays a major role by performing resource, service and network orchestration. It has and E2E view of resources, infrastructure, network and applications.

There is an external integration with BSS systems through an API layer (Northbound), guaranteeing service monetization.

The blocks represented in Figure 2.18 are considered the key ones for the ONAP framework:



1. External API Integration (OpenAPI)
2. Cloud Native Deployment & Kubernetes Integration
3. Declarative, policy-driven approach
4. Integration of other Industry Models

Figure 2.18 - ONAP Functional Architecture (Source: ONAP, " ONAP Developer Wiki," [45])

(1) External APIs components: Allow the integration and enables interoperability across Multi-VIM/Cloud instances which facilitates an ONAP workload deployment across 3P Clouds, which it is a requirement for some use cases

(2) ONAP Operations Manager (OOM) facilitates the network orchestration of workload deployments across Cloud Infrastructure and Containerized based deployments

(3) Common Services: This block is focused on providing capabilities for ONAP modules. MUSIC allows ONAP to scale to multi-site environments to support global scale infrastructure requirements. The ONAP Optimization Framework (OOF) provides a declarative, policy-driven approach for creating and running optimization applications like Homing/Placement, and Change Management Scheduling Optimization.

(4) One of key aspects of the ONAP framework is about using the same Information Model for all the framework, which is mandatory to guarantee the harmonisation among different components, like the inventory, and subsequently the network topology, and policy models, linked to services. There is a push to evolve the information model to support and cover other SDOs information models (IETF, ETSI NFV MANO, TMF SID).

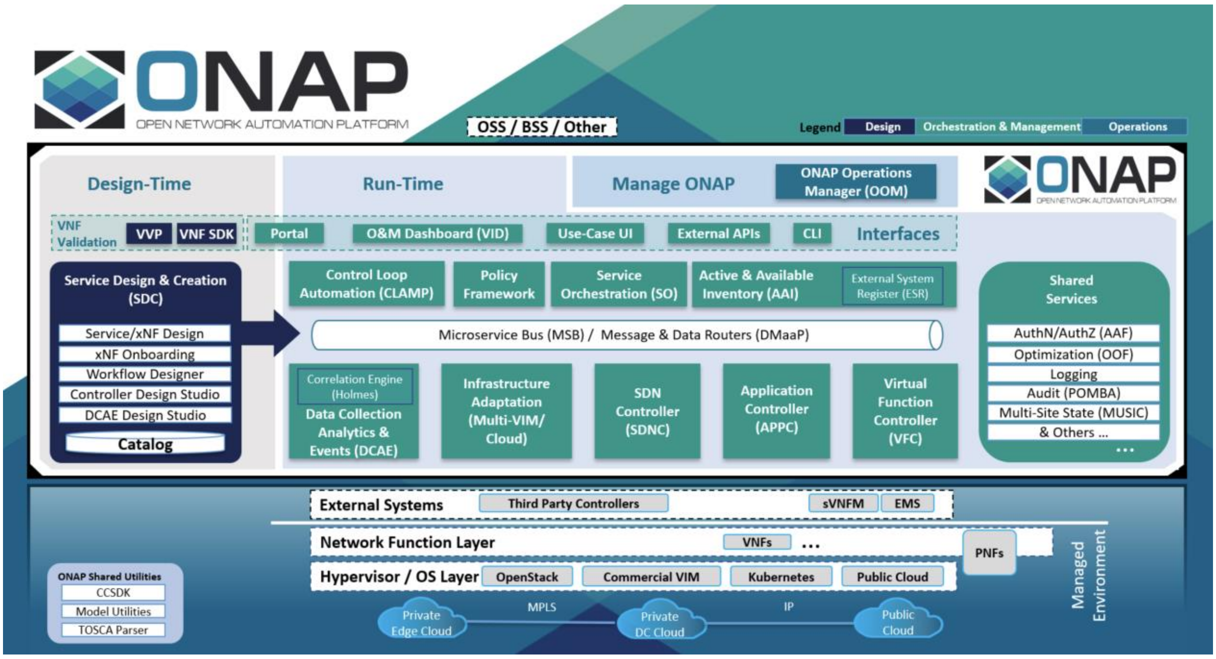


Figure 2.19 - ONAP Architecture (Frankfurt Release) (Source: ONAP, “ ONAP Developer Wiki,” [45])

3 Management and Orchestration of Network Slices

Network slicing is a core part of 5G and B5G networks. It allows network operators to dynamically allocate and multiplex virtualized network resources and offer them as a service to their customers. Thus, a single physical network can accommodate the different and often contrasting or competing QoS requirements of its tenants. This enables the coexistence of the three main 5G use cases: machine-type communication, ultra-reliable low latency communication and enhanced mobile broadband content delivery, as well as a plethora of multiple other logical networks over a shared infrastructure. Slices should be isolated, so that 1) the performance of one slice does not affect the performance / SLAs of the others, 2) an attack or bug is sandboxed within the affected slice (slice security) and 3) sensitive information of one slice is not leaked / shared (slice privacy).

D1.3 provided an introduction to the concept of network slice management and orchestration and described how it can be used in the 5G-VINNI facilities. This deliverable D1.6 leverages results from use cases and early experiments performed at the 5G-VINNI facilities and acts as an update to our approach to slicing. Further, updates from the state of the art are presented, such as the integration of cloud-native applications to NFV deployments (i.e. Usage of Kubernetes within OSM Release SEVEN) and the scope of slice lifecycle management is expanded, discussing Lifecycle, Fault, Performance, Configuration Management.

This section is focused on management and orchestration (M&O) of network slices, updating and extending the information previously provided in D1.3, and incorporating learnings from the practical deployment of the 5G-VINNI facility sites. The following sub-sections address the topic from different (but in some cases inter-related) perspectives, such as lifecycle management, container orchestration, integration of vertical customers, security, run time management, KPI monitoring, multi-domain and edge clouds M&O. At the end of the section, a summary of lessons learned by each 5G-VINNI facility site is provided.

3.1 Lifecycle Management

The lifecycle management represents the set of management capabilities that are used for changing the state of manageable objects throughout the entire lifecycle, from their creation to their termination. These capabilities, presumably offered via industry-standard northbound interfaces, convey the following two artefacts:

- A group of management operations[5] and/or notifications, providing primitives to view and manipulate objects. These primitives are network-agnostic, in the sense they do not include information about the semantics of the managed objects. The implementation of these primitives is evolving towards RESTful HTTP-based APIs, although other protocols (e.g. RESTCONF, NETCONF) can also be used.
- An information model, specifying which network entity is managed using these capabilities. This information model describes the semantics of the class representing that network entity. This semantics (relationships, constraints) allows associating objects with instances of that network entity. Information model definitions, typically specified using protocol-neutral language like UML, are mapped into data model definition (e.g. YAML, YANG, TOSCA) used for implementation

In network slice lifecycle management, the managed objects are associated with instances of network slices. In 3GPP, the lifecycle management of NSIs is achieved by pinning network-agnostic operations (first artefact) down to the corresponding 3GPP network slicing model (second artefact), i.e. Network Slice IOC. According to 3GPP [46], the lifecycle of a NSI is split into four phases (see

[5] These management operations can include generic management operations (e.g. create, read, update, delete, subscribe/unsubscribe) and resource elasticity related management operations (e.g. scale-in, scale-out)

Figure 3.1), these being the preparation phase, the commissioning phase, the operation phase, and the decommissioning phase. For further details on the individual phases, see D1.3 [2].

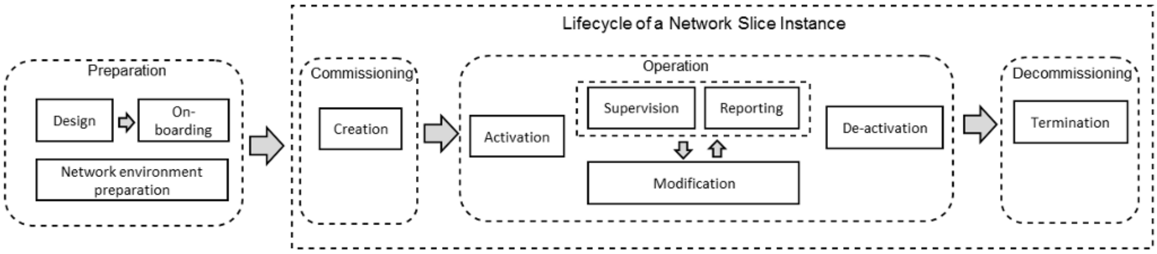


Figure 3.1 - 3GPP vision on NSI lifecycle (Source: Metro Ethernet Forum, "MEF 55; Lifecycle Service Orchestration (LSO): Reference Architecture and Framework," [26])

In 5G-VINNI, NSI lifecycle follows the 3GPP vision, with the exception of the preparation phase. The 5G-VINNI facility sites are deployed for advanced vertical experimentation, provide individual tailored service platforms where industry verticals (i.e. 5G-VINNI facility customers) can execute and assess their use cases. This experiment-driven definition of 5G-VINNI network slices requires additional testing and validation mechanisms to be incorporated in design and onboarding stages of the preparation phase, before making the service offerings publicly available in the 5G-VINNI service catalogue. In the following, we provide a description of 5G-VINNI lifecycle management phases.

3.1.1 Preparation phase

The preparation phase includes all the activities that are needed before publishing a 5G-VINNI Service Blueprint (VINNI-SB) [7] in the Service Catalogue. As captured in Figure 3.2, the preparation phase consists of three group of activities, arranged in the so-called periods.

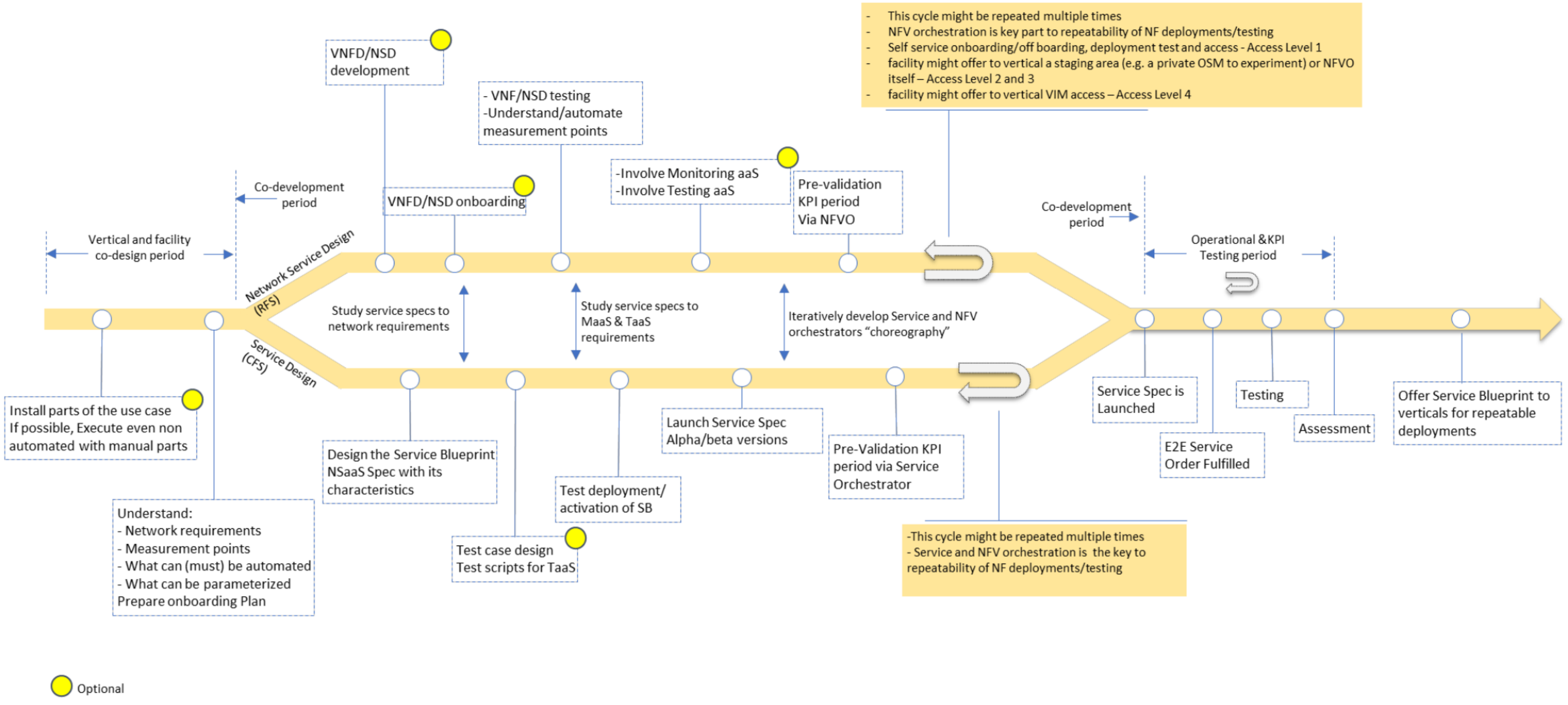

Vertical and facility co-design period
Co-development period
Network Service Design (RFS)
Service Design (CFS)
Install parts of the use case If possible, Execute even non automated with manual parts
Understand:
- Network requirements
- Measurement points
- What can (must) be automated
- What can be parameterized
Prepare onboarding Plan
VNFD/NSD development
VNFD/NSD onboarding
- VNF/NSD testing
-Understand/automate measurement points
-Involve Monitoring aaS
-Involve Testing aaS
Pre-validation KPI period Via NFVO
Study service specs to network requirements
Study service specs to MaaS & TaaS requirements
Iteratively develop Service and NFV orchestrators "choreography"
Design the Service Blueprint NSaaS Spec with its characteristics
Test case design Test scripts for TaaS
Test deployment/ activation of SB
Launch Service Spec Alpha/beta versions
Pre-Validation KPI period via Service Orchestrator
- This cycle might be repeated multiple times
- NFV orchestration is key part to repeatability of NF deployments/testing
- Self service onboarding/off boarding, deployment test and access - Access Level 1
- facility might offer to vertical a staging area (e.g. a private OSM to experiment) or NFVO itself – Access Level 2 and 3
- facility might offer to vertical VIM access – Access Level 4
-This cycle might be repeated multiple times
- Service and NFV orchestration is the key to repeatability of NF deployments/testing
Co-development period
Operational &KPI Testing period
Service Spec is Launched
E2E Service Order Fulfilled
Testing
Assessment
Offer Service Blueprint to verticals for repeatable deployments
Optional

Figure 3.2 - 5G-VINNI NSI preparation phase

The co-design period. During this initial period, the vertical decides on which 5G-VINNI facility site(s) the 5G-VINNI slice is deployed. The corresponding 5G-VINNI facility site operator(s) must understand the vertical needs from the 5G-VINNI facility in terms of (i) *what needs to be extended and co-developed*; (ii) *what and where needs to be measured and tested*; (iii) *what is the plan of onboarding*. For (i), an initial sanity check of the vertical use case is performed. This early installation of the use case allows both stakeholders to answer (ii) and (iii), by reaching a common understanding with respect to:

- Vertical requirements from the 5G system and the capabilities of the 5G facility.
- Service onboarding, as some parts of the vertical service might need to be developed for an automated deployment by orchestrators. Here, the verticals need to understand the target orchestrator and the supported NFV models (YANG or TOSCA).
- What can be parameterized. This involves the vertical to clearly identify if there are any service parameters that can be parametrized and reconfigured during orchestration.
- The procedure and tools for automated test, measurement and validation. This also involves shepherding the vertical in order to reuse test scripts and develop the plugins for testing.

The iterative co-development period. Upon reaching a common understanding, the vertical and the 5G facility site operator can proceed with the development of corresponding descriptors in two separate branches: Customer-Facing Service (CFS) descriptors and Resource-Facing Service (RFS) descriptors. In the first branch, the 5G-VINNI facility operator develops the VINNI-SB together with the vertical. In this collaborative activity, the VINNI-SB is designed, validated and pre-launched via the Service Orchestrator, with the invocation of underlying Testing-as-a-Service (TaaS) and Monitoring-as-a-Service (MaaS) capabilities. In the second branch, the 5G-VINNI facility operator develops the required NFV related descriptors, including VNFDs and NSDs, which enforce VINNI-SB at the NFV infrastructure level. This development includes the VNFD/NSD design, on-boarding and validation, all these activities being executed by the NFVO together with consumable TaaS and MaaS capabilities. It is worth noting that the individual branches are iterative and intertwined, which means that the Network slice design process (CFS descriptors branch) can interact with the NFV service design process (RFS descriptors branch) and the other way around.

The operational and KPI testing period. Once CFS and RFS descriptors are in place, an official testing period can start according to the 5G-VINNI facility plan. This period allows the vertical to make repeatable and scheduled service orders of the developed VINNI-SB via the portal, as well as perform KPI monitoring and assessment, to ensure the service behaves as expected.

Once the period is done, the designed VINNI-SB can be qualified as “mature” and “stable”. At this stage, the 5G-VINNI facility operator can register this VINNI-SB into the 5G-VINNI Service Catalogue, making it publicly available for any 5G-VINNI facility customer.

3.1.2 Commissioning phase

In this phase, a 5G-VINNI network slice instance is deployed and made available for consumption. To achieve this, the following steps are applied:

1. The vertical browses the 5G-VINNI service catalogue, identifies the most relevant slice and selects the corresponding VINNI-SB.
2. (Optional) If 3rd party VNF hosting is allowed and the vertical wants to include his own VNFs, he proceeds with the on-boarding of corresponding VNFDs. This entails going back for the preparation phase, and execute co-development period on the second branch for VNFD validation and testing.
3. The vertical issues a service order. To this end, the vertical fills in the selected VINNI-SB, specifying values for the modifiable parameters according to his service needs, and sends it out to the 5G-VINNI facility operator.

4. The 5G-VINNI facility site operator verifies that the received VINNI-SB is duly filled. There are two possible scenarios in this regard:
    a. If duly filled, this means that service order is correct. This allows going to step 5
    b. Otherwise, the operator informs the vertical of the cause of the error, going back to step 3.
5. The 5G-VINNI facility site operator generates a 5G-VINNI Service Descriptor (VINNI-SD) out of the service order, and makes it available for the vertical. Until completion of step 6, the vertical can query, update and delete the service order.
6. The 5G-VINNI facility site operator puts the service order in queue for feasibility check. In this process, the operator assesses if the service requirements declared in the VINNI-SD can be satisfied at the infrastructure level. This requires comparing resource requirements specified in referenced VNFDs/NSDs against the 5G-VINNI facility site infrastructure status. Two scenarios are possible in this regard:
    a. If VINNI-SD is feasible, then to go to step 7
    b. Otherwise, the operator informs the vertical of the cause of the error, going back to step 3.
7. The 5G-VINNI facility site operator produces the SLA, taking the relevant information from the VINNI-SD. It also creates a new entry in the 5G-VINNI facility site repository, where the record of the network slice instance to be deployed will be kept.
8. The 5G-VINNI facility operator deploys the 5G-VINNI NSI, by instantiating required network services and providing connectivity across them.
9. The 5G-VINNI facility operator provides the deployed 5G-VINNI NSI with the appropriate semantics, by injecting day-1 configuration primitives on individual PNFs and VNFs.
10. The 5G-VINNI facility operator provisions the 5G-VINNI NSI to the vertical, making it available for consumption according to the agreed exposure level [7].

3.1.3 Operation phase

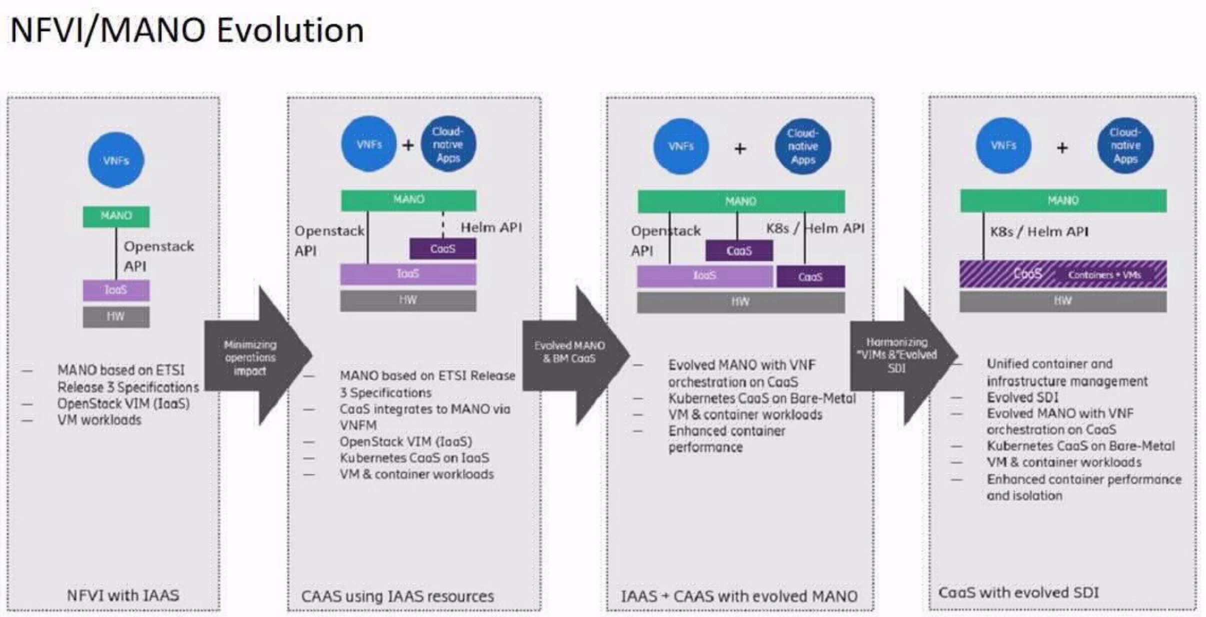
In this phase, the vertical sets up trials of innovative use cases atop the 5G-VINNI NSI, validating their KPIs and assessing their readiness through the execution of a test campaign. As part of the experimentation, the in-NSI resource capacity and traffic load can be modified, so the vertical can test the use case behaviour in different network environments. Depending on the agreed exposure level, it will be the 5G-VINNI facility operator or the vertical who will be in charge of triggering this NSI modification at run-time.

3.1.4 Decommissioning phase

Once the test campaign is done, the NSI is no longer needed. In this decommissioning phase, the NSI is terminated and its resources are de-allocated, freeing them up for use in other NSIs. In case some NSI network functions were shared with other NSIs, re-configuration (if PNF and/or VNF sharing) and scale-in (if VNF sharing) operations MAY also be needed in this phase.

3.2 Extending MANO with container orchestration

Nowadays, the vast majority of the cloud-native NFV environments are based on the ETSI NFV blueprint and architecture depicted in ETSI GS NFV-SWA 001, which was later superseded by ETSI GS NFV 002 [47] with OpenStack as the dominating VIM. Taking the assumption that in this model 5GC is cloud-native, new functionalities are required and, therefore, a potential dual-mode NFVI with Kubernetes (K8s) as the orchestration engine for the Container Network Functions (CNFs) is needed. Figure 3.3 shows a path towards the evolution of the NFVI/MANO architecture to a simplified NFV model where the Container as a Service (CaaS) layer will be used both for CNFs and VNFs.



NFVI/MANO Evolution

- MANO based on ETSI Release 3 Specifications
- OpenStack VIM (IaaS)
- VM workloads

NFVI with IAAS

- MANO based on ETSI Release 3 Specifications
- CaaS integrates to MANO via VNFM
- OpenStack VIM (IaaS)
- Kubernetes CaaS on IaaS
- VM & container workloads

CAAS using IAAS resources

- Evolved MANO with VNF orchestration on CaaS
- Kubernetes CaaS on Bare-Metal
- VM & container workloads
- Enhanced container performance

IAAS + CAAS with evolved MANO

- Unified container and infrastructure management
- Evolved SDI
- Evolved MANO with VNF orchestration on CaaS
- Kubernetes CaaS on Bare-Metal
- VM & container workloads
- Enhanced container performance and isolation

CaaS with evolved SDI

Figure 3.3 - NFVI/MANO Evolution (Source: Ericsson, "NFVI evolution," [48])

The first step towards this final view is pictured as a CaaS layer introduced on top of the NFVI working as IaaS integrated with the MANO platform via the VNFM. In the second step, the MANO platform is able to directly manage and orchestrate CNFs as CaaS and K8s can be run directly on bare metal. It is important to remark that, currently, this is the closest step that solutions such as ETSI OSM version EIGHT are heading to. Finally, as the last step, it is proposed to unify the containers and VMs management layer and as a result obtain a simplified NFV architecture with an evolved MANO and Software Defined Infrastructure (SDI) platform.

As mentioned above, traditionally the usage of hypervisors or VIMs such as OpenStack has been the main approach to support NF deployments. In ETSI GS NFV-MAN 001 [49] MANO functions and interfaces assume that CNFs are running on VMs created by hypervisors, since the operations carried out by the Nf-Vi reference point aim at VMs rather than any kind of virtualization container. ETSI GS NFV-EVE 004 [50] studies the impact of a wide range of virtualization technologies in the NFV framework. More concretely, there are two sections dedicated to explain container-based solutions ("4.3 OS Containers" and "4.4 Higher-level containers") and two sections dedicated to describe the impact of these solutions over the NFV/MANO framework ("5.2 OS Containers" and "5.3 Higher-level containers").

ETSI GS NFV-EVE 004, clause 5.2.1 states that: "The larger degree of sharing the underlying infrastructure allows an operator to run a high number of VNFC instances per compute/storage unit compared to the virtualization technologies based on a hypervisor."

It is important to note that as the kernel running VNFC instances is shared, all the VNFCs sharing that kernel will have the same functionality and configuration which is not appropriate for those scenarios where VNFCs require dedicated kernel configurations.

Besides, ETSI GS NFV-EVE 004, clause 5.2.2 discusses the impact of this virtualization technology on MANO functional blocks stating that there is not a significant impact on them except for the VIM, the Nf-Vi reference point and on VNFDs elements. Thereupon, the adoption of cloud-native and container-based solutions in the MANO framework is viable. Sections 5.3.1, 5.3.2 and 5.3.3 explore the implementation of this approach in existing solutions.

There is a specific CNF case, Kubernetes-based VNFs (KNFs). The usage of KNFs unlocks a vast volume of new services and packages that can be deployed besides VNFs and PNFs, thanks to the K8s/Docker image registry. In order to deploy a KNF (as a special case of CNF) a functional and operational K8s cluster SHOULD be required in order to manage all the K8s internal management operations. This K8s cluster can be seen as a technology that enables the deployment and management of micro-services in a cloud-native way, not as an isolated cloud element.

The K8s cluster MAY be created using a wide range of methods. Below, two generic approaches are described to achieve the aforementioned scope:

1. **K8s NS**: The K8s cluster can be deployed as a NS within the network management reachability of the constituent VIM, as any other NS comprised of multiple VNFs/PNFs. See Figure 3.4.
2. **External K8s Bare Metal Cluster**: The K8s cluster MAY be required to be allocated on an external bare metal device, outside the network management reachability of the VIM but with a physical connection at the NFVI level (a link between the bare metal device and the VIM. See Figure 3.5.

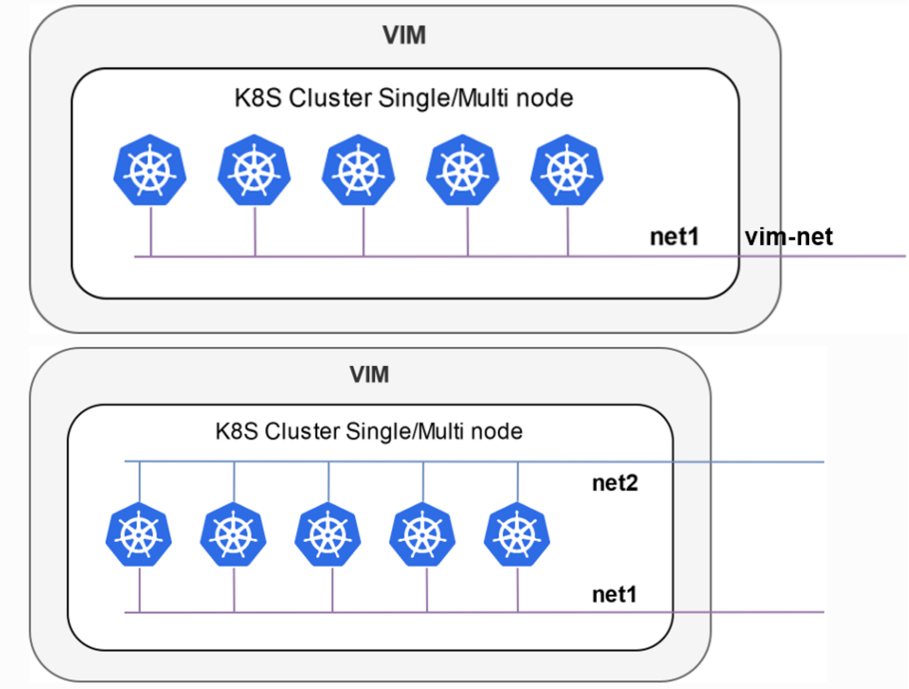


Figure 3.4 - K8s inside a VIM (single or multi-net) (Source: “Open Source MANO’s documentation” [51])

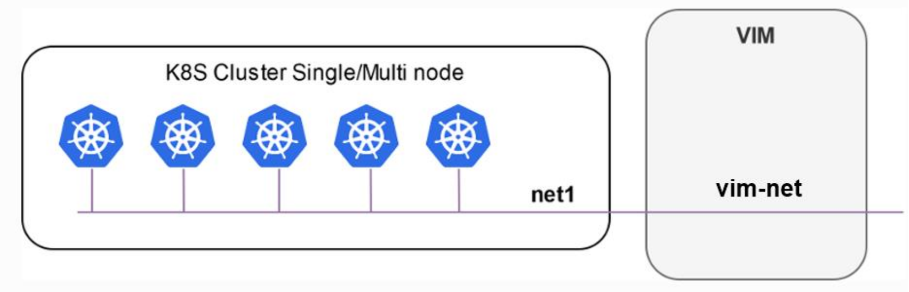


Figure 3.5 - K8s cluster outside a VIM (Source: “Open Source MANO’s documentation” [51])

Once the deployment of the K8s cluster has been completed, the respective NFV MANO platform will be able to deploy KNFs as long as these two requirements are fulfilled:

1. KNFs/CNFs should be exposed to the network in order to expose their services and/or interact with other NFs.
2. Persistent storage is supported in KNFs/CNFs.

The instantiation procedure of a KNF MAY differ from VNF and PNF instantiation within an NFV MANO platform. An extension of the NFV MANO IM MAY be required in order to fulfil the specific requirements of this new kind of NFs i.e. OSM IM [41] had to be extended with a new entry known as "kdu – Kubernetes Device Unit" and a "k8s-cluster" entry. The inclusion of this new type on NF contemplates several new NF composition methods ranging from pure CNFs/KNFs to Hybrid Network Functions (HNFs) composed of mixed NFs types.

There are clear requirements related to Container Management and Orchestration, highlighted in the ETSI NFV specifications, which SHALL be followed to guarantee an ETSI NFV Compliance by 5G-VINNI Ecosystem:

- ETSI GS NFV-IFA 010: "Management and Orchestration; Functional requirements specification" [52];
- ETSI GS NFV-IFA 036: "Specification of requirements for the management and orchestration of container cluster nodes" [53];
- ETSI GS NFV-IFA 040: "Requirements for service interfaces and object model for OS container management and orchestration specification" [54].

The following high level requirements are detailed in the above specifications which SHALL be taken into consideration for including the KNF (CNF) capabilities in a NS:

- Requirements on CISM/CIR (Container Infrastructure Service Management/Container Image Registry) exposed service interfaces
- Requirements on M&O of virtualised containers
- CISM SHALL expose the following services to NFVO:
    - OS container workload management
    - OS container compute management
    - OS container storage management
    - OS container network management
    - OS container configuration management.

3.3 Integration of vertical customers

The 5G-VINNI facility leverages on the service capability exposure feature to make management capabilities available for consumption towards individual 5G-VINNI facility customers (e.g. ICT-19 verticals), thereby facilitating the integration of their M&O solutions (e.g. ICT-19 system) with the 5G-VINNI software stack. This feature, which provides the 5G-VINNI facility operator with the ability to securely expose management capabilities of 5G-VINNI facility towards authorized customers, is based on the definition of four capability exposure levels. The selection of a specific level allows a vertical to get a different set of operational capabilities from 5G-VINNI facility. For more details on these 5G-VINNI levels, see [7] and Table 3.1.

Table 3.1 - 5G-VINNI capability exposure levels

Exposure Level	NBI's exposed to 5G-VINNI facility customers	Main 5G-VINNI facility sites			
		Norway	UK	Spain	Greece
Level 1	E2E Service Management and Operation NBI	FO APIs	FO APIs	OSM NBI APIs	OSM NBI APIs
Level 2	Level 1 NBI + Domain controllers NBI's (5G-RAN NBI + 5G-CORE NBI + Transport controller NBI)	FO APIs	See NOTE 3	See NOTE 4	See NOTE 4

Level 3	Level 2 NBI's + NFVO NBI (see NOTE1)	See NOTE 4	See NOTE 3	See NOTE 4	See NOTE 4
Level 4	Level 3 NBI's + VIM NBI (see NOTE2)	Level 3 APIs + OpenStack project/tenant APIs			
NOTE1: Different 5G-VINNI facility customers consuming the same OSM NBI require the definition of different NFVO tenants. NOTE 2: Different 5G-VINNI facility customers consuming the same VIM NBI require the definition of different VIM tenants. NOTE3: The UK facility will be implementing the upgrade to SA during Q1 2021. APIs for levels 2 and 3 will be defined once this is complete. NOTE 4: Not yet clear if it will be supported by the respective facility site at the time of writing.					

The definition of the service capability exposure has been recognized in SDOs including 3GPP SA5 and ETSI ISG ZSM as a key enabler for the support of PNI-NPN scenarios, from operational viewpoint. 5G-VINNI related use cases fit these scenarios, as follows:

- 5G-VINNI facility can be modelled as the public network (PLMN), formed by a collection of public nodes, with **5G-VINNI facility operator playing the role of public NOP**.
- 5G-VINNI facility customer defined premises can be modelled as a private node, **with 5G-VINNI facility customer playing the role of private NOP**.

The 5G-VINNI system leverages the above rationale for the design of the service capability exposure mechanism that can be used in archetypical ICT-19 use cases: an end-to-end vertical service where some VNFs are deployed on the ICT-19 site (private node), and the rest can be executed on a 5G-VINNI facility site (public node). Examples of these VNFs are summarized in Table 3.2. This mechanism:

- ensures the private NOP to retain control over the private VNFs, when these are deployed on the 5G-VINNI facility. This means that, apart from leading VNF configuration management, the private NOP should be able to trigger lifecycle management operations (e.g. scaling) over those VNFs, when required. To that end, the 5G-VINNI facility needs to expose necessary capabilities to the private NOP.
- ensures the private NOP is able to configure and manage connectivity between the ICT-19 site and the 5G-VINNI facility site, in case private VNFs are deployed on the ICT-19 site. This requires the public NOP to communicate external-facing connectivity information towards the private NOP, so that the latter can set up VPN services across the 5G-VINNI facility site and the ICT-19 site. Examples of this information include IP addresses and ports of the corresponding 5G-VINNI facility site gateway.

Table 3.2 - Examples of VNFs in archetypal 5G-VINNI use cases

VNFs		VNF Execution	
		ICT-19 site (on-premise)	5G-VINNI facility site (off-premise)
VNF nature	Private	gNB, edge application	Application server
	Public	Core UP (PGW-U/UPF), Backhaul transport nodes	Backhaul transport nodes, Core CP (MME+PGW-C/ AMF+SMF), Firewall, NAT.

For a fine-grained control of this capability exposure across public and private NOPs, 5G-VINNI system MAY use token-based authentication. When the private NOP registers into the public NOP admin domain for the first time, the first user is granted with a unique access token that specifies the set of management services it can consume at operation time. The logic behind this token-based authentication is as follows: every time a public NOP invokes a capability from a management function (e.g. E2E Service Management and Operation, 5G-RAN Controller, 5G-CORE Controller, Transport Controller, NFVO, VIM), the management function checks the permissions imbued in the token assigned to the private NOP. If these permissions include the requested capability, then the management function authorizes the corresponding API invocation. The process for the token verification by the management function (API producer) depends on the token format and the associated metadata, to be decided by individual 5G-VINNI facility sites. Figure 3.6 illustrates the applicability of token-based authentication in 5G-VINNI facility.

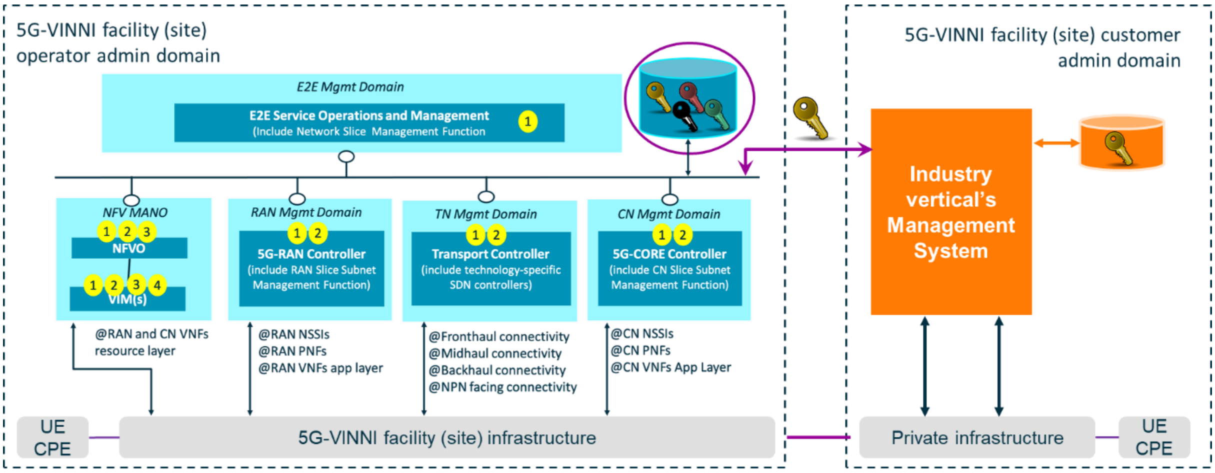


Figure 3.6 - Token-based authentication in 5G-VINNI

3.4 Security concepts for O&M

3.4.1 Automation of Firewall Security Rules

As explained in Section 4.8 of the 5G-VINNI Deliverable D1.1 [4], the security design in 5G-VINNI MAY include the separation of the NFV environment in different security zones, to limit the communication between specific functions based on predefined security needs. In this section, a more detailed description on how this security zone model can be implemented by the use of a set of security rules, is provided. Figure 3.7 illustrates this concept, where such rules can be defined as a list of items that define specifically IP, ports and applications between the different zones that need to be allowed (communication matrix). Such lists are implemented and controlled by the Firewall, following manual configuration based upon the communication requirements of functions across different zones, and then by the allocation of such rules, one-by-one in the firewall.

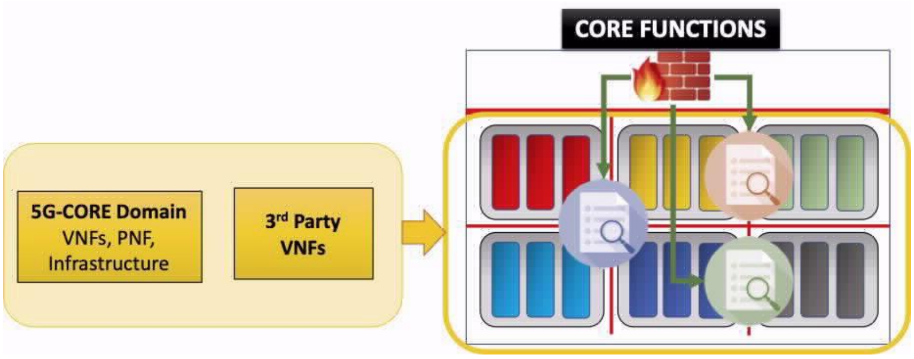


Figure 3.7 - Set of Rules among the different Security Zones

One of the new features from the management and orchestration point of view is that the creation and implementation of such list of rules can be automated and directed by the E2E-SO at the time of the deployment of the slice or at the deployment of individual VNFs. The respective rules are inserted in the firewall and put in operation in coordination with the overall new deployment as it is illustrated in Figure 3.8.

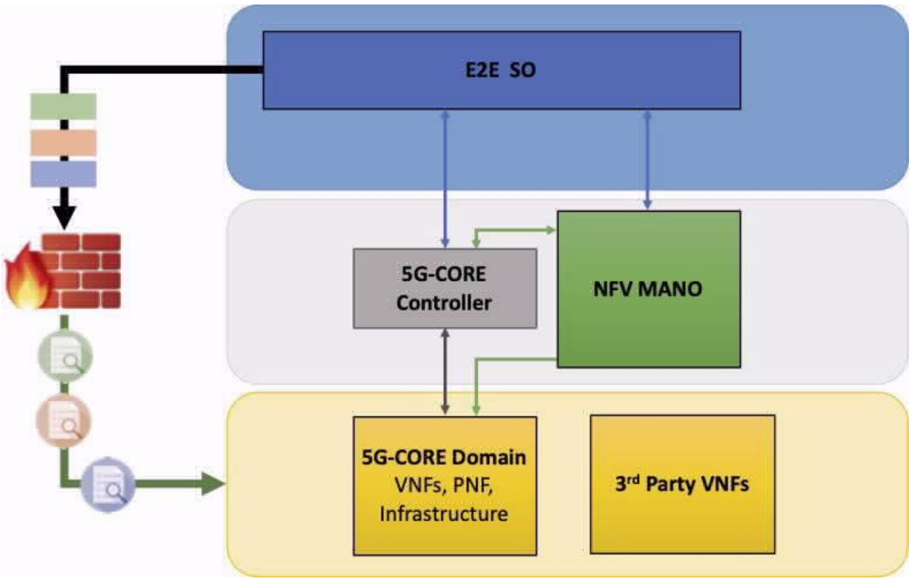


Figure 3.8 - Automation of the configuration of the Set of Rules among the different Security Zones

3.5 Run time management of NS and KPI monitoring

5G enables the concept of a distributed network across various edge nodes. One of the key limitations in the deployment of a data acquisition network across a distributed environment is the capability of automated management of such an infrastructure, especially during the runtime of the system. While the current deployment of the end-to-end system may be executed using the mechanisms described in Section 3.3, during runtime, large number of operations are executed by human administrators, which in turn leads to decreased deployment capabilities. The main functionality of Network Management is to provide system configuration, monitor and log its performance as well as to observe and mitigate faults, in order to orchestrate system-wide operations through the application of policies. In order to provide more automation to the end-to-end runtime management, a new structure for individual Network Service management is proposed following the concept of localized management in which specific operations can be executed at the edge of the network while some others have to be forwarded to the central entities. This is based on the same principles as described in Section 3.3, extended for the immediate application in the context of runtime.

Figure 3.9 illustrates the implementation of the Network Edge Management Infrastructure (NEMI) solution. This describes the implementation in a layered architecture divided into Active System, Data and control plane.

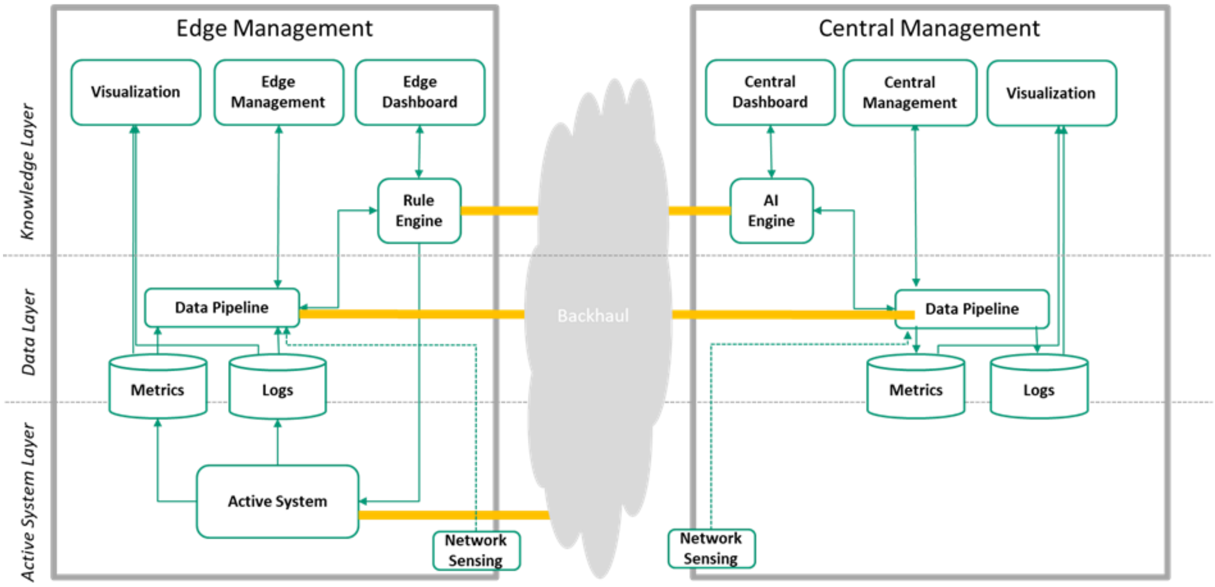


Figure 3.9 - Network Management Architecture

In Figure 3.10, the remote edge management is mapped one-to-one with the reference architecture described. Additionally, it demonstrates the components and technology selection of various layers and processes.

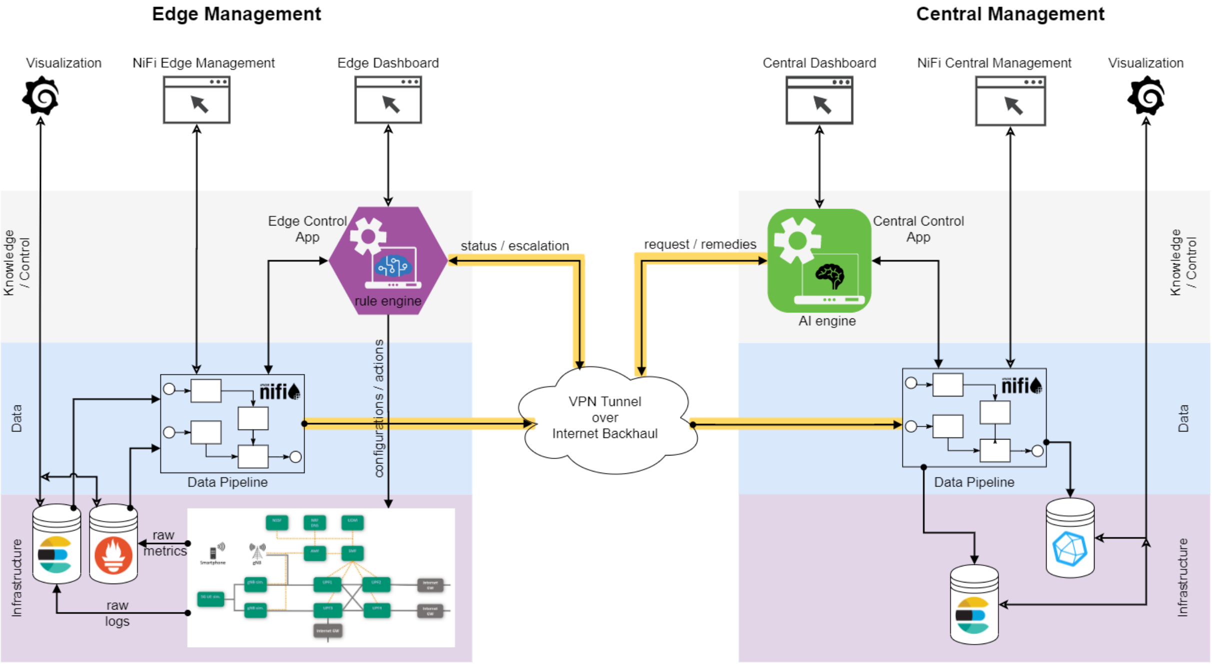


Figure 3.10 - NEMI Remote Node Management

At the edge, on the infrastructure/active system layer, the raw metrics and raw logs are collected from the active systems. The hardware, OS as well as process specific metrics are collected from the active systems and provided to Prometheus, an open source monitoring system tailored with an efficient time series database. Prometheus [55] works on pull-based mechanism, where the agents or exporters collect the metrics and Prometheus pulls those metrics with the help of the agents. Raw Logs are pushed to Elasticsearch [56] with syslog. This setup ensures that historical system data is archived and retrievable for both online and offline processing, as well as provide features to explore and visualize the metrics and logs. In the central location, the aggregated metrics and logs collected from several edges are stored in InfluxDB [57] and Elasticsearch repositories respectively. InfluxDB is also an efficient time series database with push-based approach.

Both in the edge and in central, the data layer is composed of a peer-to-peer based distributed data processing component, realized through Apache NiFi instances. Apache NiFi [58] provides a powerful and reliable data Flow management system with backpressure support that tracks the data flow and provides the provenance for the data it handles from the start to the end. Thus, in NEMI, Apache NiFi is used for defining the various data pipelines for collecting metrics and logs and for routing them reliably from the edge to the central. The setup and the internal setup is described as well in D1.5 [5].

Knowledge layer deals with the monitoring and observation of the active system and automated actions (either based on rules or AI) to manage and operate the system. On the edge-side, the reaction to system events must be near-real time. The control-loop at the edge handles cases that are autonomic in nature, i.e., without cognition / intelligence and the need for deeper analysis. Thus, on the edge-side, a rule-engine, is realized by Drools Fusion [59], a complex event processing solution. Drools allows the definition of rules in its own Domain Specific Language (DSL) that is both human readable and machine friendly.

On the central side, data and events escalated by all the edges are processed. Events escalated by the edges require cognitive actions, either in the form of an AI or human intervention. Thus, the edge-central control-loop is much slower than the edge-local control loop. The control app at the central is envisioned to be an AI-based knowledge engine. This was also discussed in D1.5 [5] along with the internal control loops for edge to central registration and communications.

3.6 Multi-domain network slicing

5G-VINNI is a large-scale E2E facility that provides 5G capabilities for advanced vertical experimentation in multi-domain environments. These capabilities are made available for consumption using Network Slice as a Service (NSaaS). This service delivery model allows a vertical to use the provided network slice for validation activities, deploying one or more use cases and validating their KPIs under different load conditions. Despite being formed through the combination of multiple smaller facilities (i.e. facility sites, each defining a single administrative domain), like in 5G commercial networks, orchestration in one facility site should be able to be executed in another facility site. This cross-domain orchestration procedure requires common standard interfaces and information models across those domains to enable the interoperability among the multi-vendor solutions adopted in each segment. This constitutes a real challenge, since most of the NFV, MEC and SDN solutions available today from vendors or open-source communities expose proprietary interfaces, which refer to non-standard information models.

A key enabler for vertical experimentation in 5G-VINNI facility is reproducibility, which can be defined as the ability to generate repeatable slice instances at multiple locations and at different time instants. Reproducibility allows any vertical to replicate experiments in controlled environments, assessing the variation of use case KPIs depending on selected capabilities. Different sites provide different 5G capabilities, not only in terms of resource capacity, but also in terms of functionality (e.g. edge support, telemetry/monitoring). To choose the capabilities that will support the use case execution, a vertical can decide where to deploy the slice: on one or another site, or across two or more sites. The latter is of particular interest for verticals, taking into account that many vertical services will span beyond the boundaries of a single administrative domain.

Multi-domain slice deployments brings several challenges in 5G-VINNI facility, since they require both data plane connectivity between the involved sites, and also interworking between their orchestration systems. For this interworking, two approaches can be followed: i) hierarchical orchestration; and ii) peer-to-peer orchestration. The first approach assumes the definition of a parent orchestrator, sitting on top of multiple child orchestrators, coordinating their workflows and providing translation of their information / data models. This introduces significant burdens in management scalability, as the number of sites connected to this master orchestrator increases. Additionally, the scenario of having a network operator taking the broker role is unrealistic for upcoming commercial, operational networks, as it would raise concerns with the rest of operators in terms of privacy, auditability and non-repudiation. For this reason, the peering approach is preferred for federating domains. This was also discussed in D1.5 [5].

Considering the facility site components, three options can be considered for federation:

- Federation at Service Orchestration level (SO-SO): the SOs from different sites exchange information and expose their capabilities across them.
- Federation at Network Orchestration level (NFVO-NFVO): the NFVOs from different sites exchange information and expose their capabilities across them.
- Federation at different orchestration levels (SO-NFVO): the SO from one site communicates with the NFVO from another site.

All the above options are technically feasible when the federated sites rely on the same orchestration solution. In such a case, the use of proprietary interfaces is enough to enforce the required communication and capability exposure across domains. However, this scenario is rather unrealistic, as is unlikely to be found in commercial networks, where federation may involve multiple sites from different network operators, each making usage of a different orchestration solution. In 5G-VINNI, though, there are multiple facilities each making usage of a different orchestration solution, and interoperability can only be achieved by means of standard interfaces. Table 3.3 gives an insight into the three federation options (available also in D1.5, repeated here for clarity), specifying their main features and the standard interfaces that can be used to fulfil these features. As seen, there exists at least one interface to implement every federation option.

Table 3.3 - Federation options

Option	Main Features	Standard interfaces
SO « SO	Information exchanged with external SO: list of on-boarded VINNI-SBs, selected configuration of deployed slice (subnet) instances. Operations exposed for external SO invocation: slice (subnet) provisioning; slice (subnet) performance assurance; slice (subnet) fault supervision; network functions application layer conf & mgmt.	MEF LSO Interlude
NFVO « NFVO	Information exchanged with external NFVO: list of on-boarded NSDs / VNFDs; records of deployed network service / VNF instances, with information on their resources. Operations exposed for external SO invocation: network service / VNF lifecycle mgmt; network service / VNF monitoring; network service / VNF resources mgmt.	Or-Or
SO « NFVO	Information exchanged with external SO: the same as for NFVO « NFVO, but without information on instances resources. Operations exposed for external SO invocation: the same as for NFVO « NFVO, but without resources mgmt. Information exchanged with external NFVO: slice (subnet) – network service mapping.	Os-Ma-nfvo

Considering this analysis of the options, SO « SO is considered the most realistic solution for future commercial networks, and thus it is the one explored in 5G-VINNI project.

3.6.1 Solution for Federation at the Service Orchestration Level

MEF specifies the requirements and capabilities of the INTERLUDE interface [26]. However, no data models or protocols have been defined for the interface implementation yet. Unlike Or-Or and Os-Ma-nfvo interfaces, based on SOL005 [40] and SOL011 [60], no normative solution has been defined for INTERLUDE interface. In this context, Tele Management Forum (TM Forum) open APIs can be used. These APIs are not tied to vendor-specific orchestration solutions, allowing rapid integration and easy interoperability across domains.

As of today, a wide variety of Open APIs can be found in the TM Forum portfolio [61]. For the INTERLUDE interface implementation in 5G-VINNI, the following APIs (specified in section 4) apply: Service Catalogue API (TMF633) [62], Service Ordering API (TMF641) [63], Service Inventory API (TMF638) [64], Service Configuration and Activation API (TMF640) [65].

The SO of every 5G-VINNI facility site needs to offer these open APIs, so they can be consumed by the SOs from federated sites. The integration of open APIs in each site depends on the selected solution for the SO. In 5G-VINNI facility, two types of orchestration solutions exist:

- Open Source MANO (OSM), deployed in 5G-VINNI Spain and Greece facility. Although it was originally defined as a NFVO, OSM currently implements enhanced data models (based on SOL006 [66]) for 3GPP slicing support, thus taking the SO role.
- Nokia’s orchestration toolkit, deployed in 5G-VINNI Norway site and UK site. This toolkit includes a SO (FlowOne [67]) and a NFVO (CloudBand [68]).

Figure 3.11 shows how the integration of Open APIs is done in OSM.

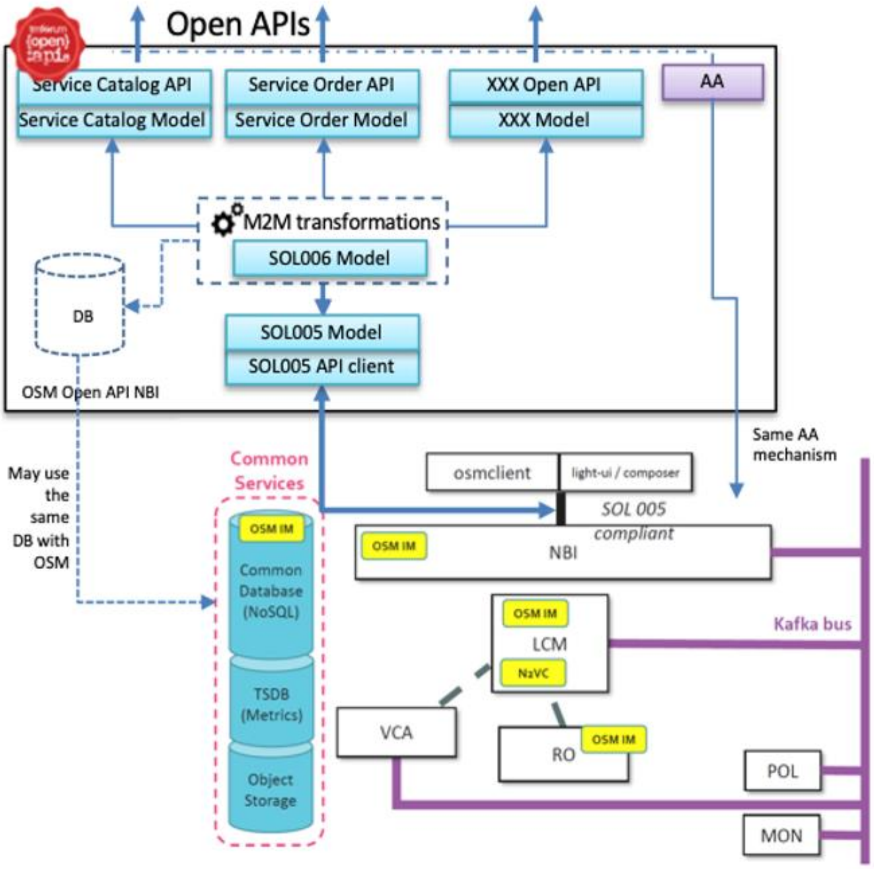


Figure 3.11 - TM Forum Open APIs in OSM

3.6.2 Example of Federation deployment and operation of an E2E slice instance across two facility sites

We explain here how federation enables the deployment and operation of an E2E slice instance across two facility sites upon vertical request. In this process, three phases can be envisioned: slice ordering, slice fulfilment and slice operation. To illustrate this, an example is considered of an eMBB slice instance across the 5G-VINNI Facility Sites in UK and Spain, with part of the slice also needing to be orchestrated into the Norway Facility Site

3.6.2.1 Slice ordering

In the first phase, the vertical gains access to the 5G-VINNI facility through the portal, browses the centralized service catalogue, selects one VINNI-SB and issues the corresponding service order. In this service order, the vertical provides a completed specification of the slice instance he wants, including information on slice topology (possibly extended with 3rd party VNFs), slice attributes (filled in with values fitting use case requirements) and slice location. We assume the following: i) the vertical wants the slice deployed across two facility sites, each having a different orchestration solution; and ii) the selected VINNI-SB was retrieved from the local catalogue of one of these sites. For our example, the vertical orders the provisioning of an eMBB slice instance across UK and Spain, by selecting a VINNI-SB with SST=1 from the 5G-VINNI service catalogue, retrieved from Spain's OSM catalogue.

The service order with the above setup is captured by the portal's order manager, which validates the order and send it to the Spain site. Then, the slice fulfilment phase begins.

3.6.2.2 Slice fulfilment

In the second phase, upon receiving the service order, Spain site checks it, realizing that part of the ordered slice needs to be deployed at Norway site. This means that federation between the SOs of both sites (OSM and Nokia's FlowOne) is needed. From this point, the event workflow is as follows. First, OSM on-boards the VINNI-SB into FlowOne's catalogue, using TMF633 [62]. Then, OSM decomposes the service order received from the portal, identifying the subnets that will be deployed on Spain and Norway sites. Finally, it triggers a service order towards FlowOne, using TMF641 [63]. With this order, OSM informs FlowOne about the topology and attributes of the slice (subnet) instance to be deployed on Norway site.

After the above actions, the slice can be commissioned. To this end, each SO first deploys the slice subnet at its site, providing day-0 and day-1 configuration on the different VNFs. Then, OSM and FlowOne exchange connectivity information of their slice subnets (e.g. IP addresses of VNF instances at the edge of each subnet) to set up a L2/L3 VPN connectivity service across these subnets, establishing an E2E data plane for the slice. The exchange of information is done with TMF641, while the VPN connectivity service instantiation is done with TMF640 [65].

3.6.2.3 Slice operation

At the operation time, the cross-domain slice can be made available to the vertical for advanced experimentation activities. As part of these activities, advanced lifecycle management operations (e.g. scaling) can be issued. In this case, cooperation between SOs is needed by means of TMF641 and TMF640.

3.7 Management and Orchestration of Edge Clouds

This section is focused on 3 options to manage and orchestrate Edge Clouds, which SHALL be compliant with SDOs (ETSI MEC and 3GPP SA6), to enable the use of edge infrastructure. The 3 options are: centralized, distributed and hierarchical orchestration. The focus will be on the first two, described in section 3.3.1. One of the main advantages of using hierarchical orchestration architecture is service scalability. Therefore, not all available options that can be used to manage and orchestrate an Edge Cloud are covered. This is addressed in D1.5. The challenge on the options selected, taking into account the need to be aligned to ETSI MEC, is about the relation of an ETSI MEC App with a service. There is a need to have an Application Descriptor, which SHALL be used by the NSD (this is out of scope for this deliverable).

The option selected shall be in accordance to Service Providers / Verticals / Consumers requirements. In Figure 3.12, the deployment models shown are linked to options that can be offered by the Service Provider and the decision on the option to pursue SHALL be based on the Service Provider Operational Model and, most importantly, the use case. The same figure provides a high level view of Service attributes (primary ones) that can be used to support a deployment model decision (RTT/Round Trip Time vs Distance vs Latency). Therefore, it is important to raise the awareness that the closer the computing power is to the service consumer, the lower is the latency, but the greater number of sites require equipment to be installed, as one can observe in Figure 3.12.

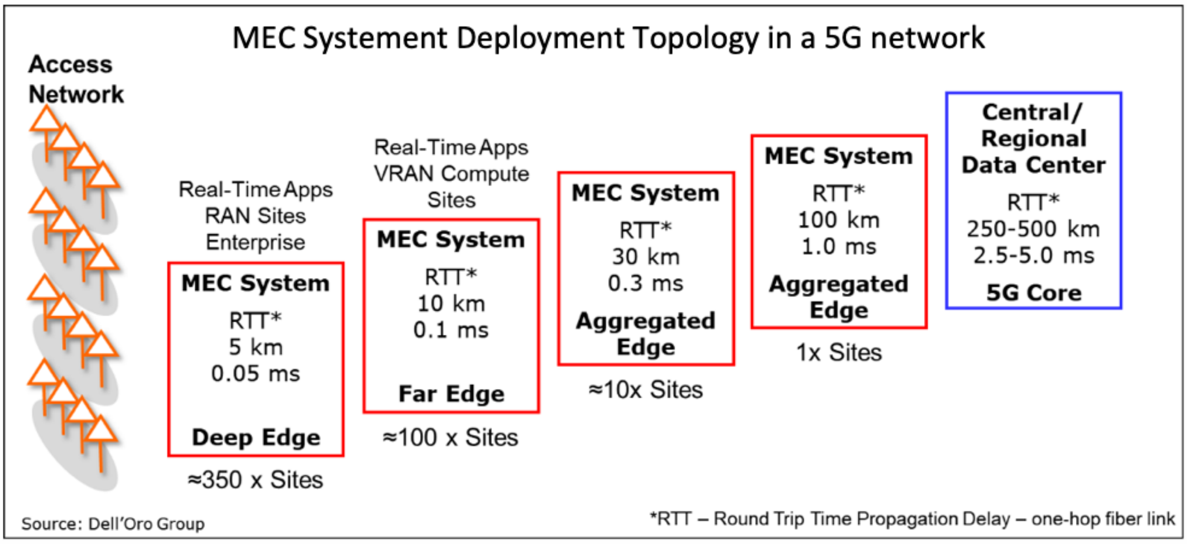


Figure 3.12 - Edge and Central Management Deployment Topology (RTT vs Latency vs Distance) (Source: 5G Core – Are We Ready? [69])

The following section describes, in more detail, the main characteristics of the centralized and distributed deployment models, and in particular how Third Party clouds are integrated in an architecture that supports the Edge.

There will be focus on the deployment models 1 and 5, according to the categorisation shown in Figure 3.13. Deployment mode 1 is considered a centralized model, in which the edge can be directly integrated to E2E orchestration and all components are running at the management plane. Deployment model 5 has several components deployed at the edge (Edge Application Orchestrator, Edge Server), and the respective UPF, which in the end is in compliance with use cases that demand very low latency.

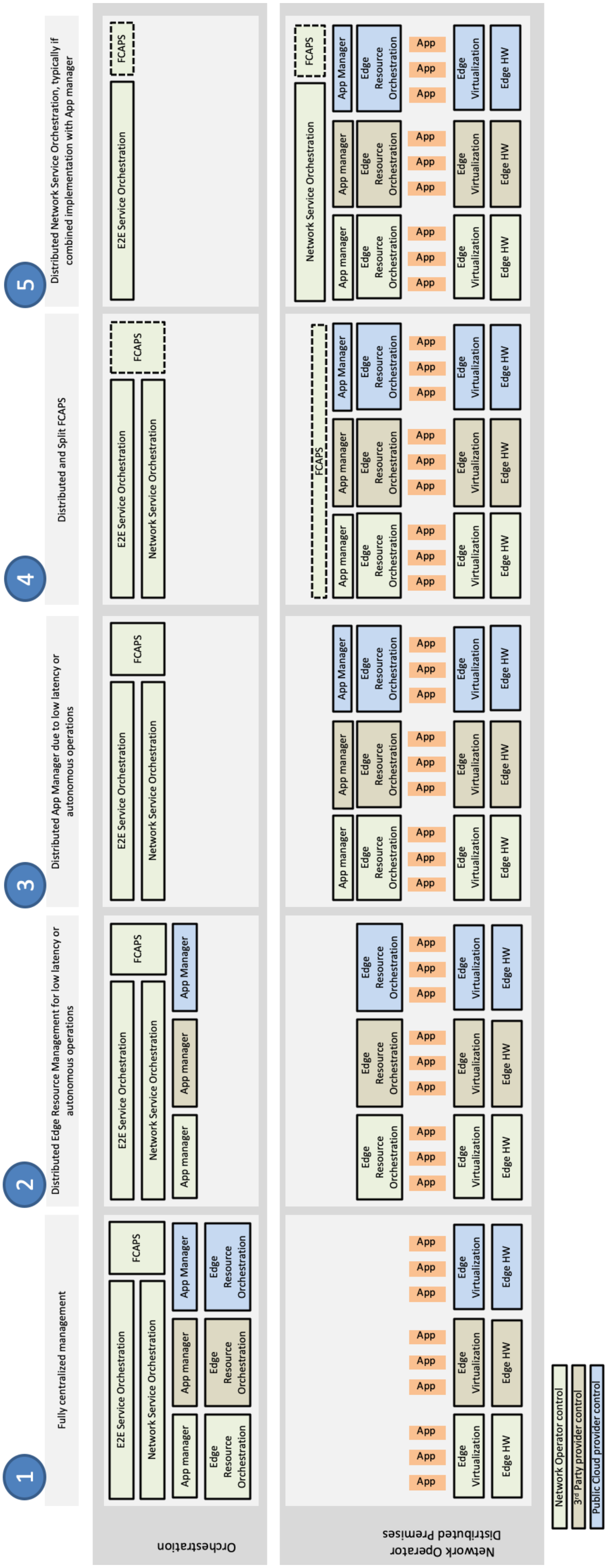

1
Fully centralized management
2
Distributed Edge Resource Management for low latency or autonomous operations
3
Distributed App Manager due to low latency or autonomous operations
4
Distributed and Split FCAPS
5
Distributed Network Service Orchestration, typically if combined implementation with App manager
Orchestration
Network Operator Distributed Premises
E2E Service Orchestration
Network Service Orchestration
FCAPS
App manager
App Manager
Edge Resource Orchestration
App
Edge Virtualization
Edge HW
Network Operator control
3rd Party provider control
Public Cloud provider control

Figure 3.13 - Deployment Models (Service Provider Example)

3.7.1 Centralized vs. distributed orchestration

The dichotomy between centralized and distributed orchestration has implications in business and operational models of Service Providers. This section addresses the reasoning for opting for one or the other. It is important to note that this selection depends on the use case and respective requirements. There is an opportunity with a 5G ecosystem to use the base stations that can support several characteristics usually demanded once an edge application is orchestrated and delivered:

- Real-time operations
- Offloading workloads that demand high computation resources
- Multimedia caching.

Main application areas and/or advantages of a centralised orchestration model are:

- eMBB use cases
- Used in scenarios with limited resources available to deploy workloads linked to the availability of a service (e.g. Edge Server; Edge Application Orchestrator)
- Limited number of edge infrastructure, and orchestration delegation operations are not required.

Main application areas and/or advantages of a distributed orchestration model are:

- URLLC & mMTC use cases
- Service/Application lifecycle operations are delegated to the Edge
- Real-time operations
- No “limitation” on the number of Edge Infrastructures (dependent on the business model)

3.7.2 Management and Orchestration of 3rd Party Edge Clouds

One of the most important requirements that shall be tackled is the possibility of deploying (containerized) workload on 3rd party Clouds. There are some challenges in supporting such a requirement with the current ETSI MEC architecture, which is the reason why there is an ongoing work which would fulfil such a demand, to enable the possibility of supporting a MultiCloud capability which is being demanded by several service providers. This is also shown in the deployment model - public cloud provider.

A common approach is recommended for such an architecture, which would enable the workload deployment across 3rd party Edge Clouds. This is also the result of ETSI MEC “Study on MEC support for alternative virtualization technologies” [70]. The introduction of the Container Infrastructure Service Management (CISM) by ETSI NFV [71] [54] facilitates deployment, management and/or orchestration. The CISM exposes the OS Container Manager Services which ultimately can be used to manage and orchestrate services which are deployed on different NFV-MANO and even on an external entity (e.g. 3rd party Edge Cloud).

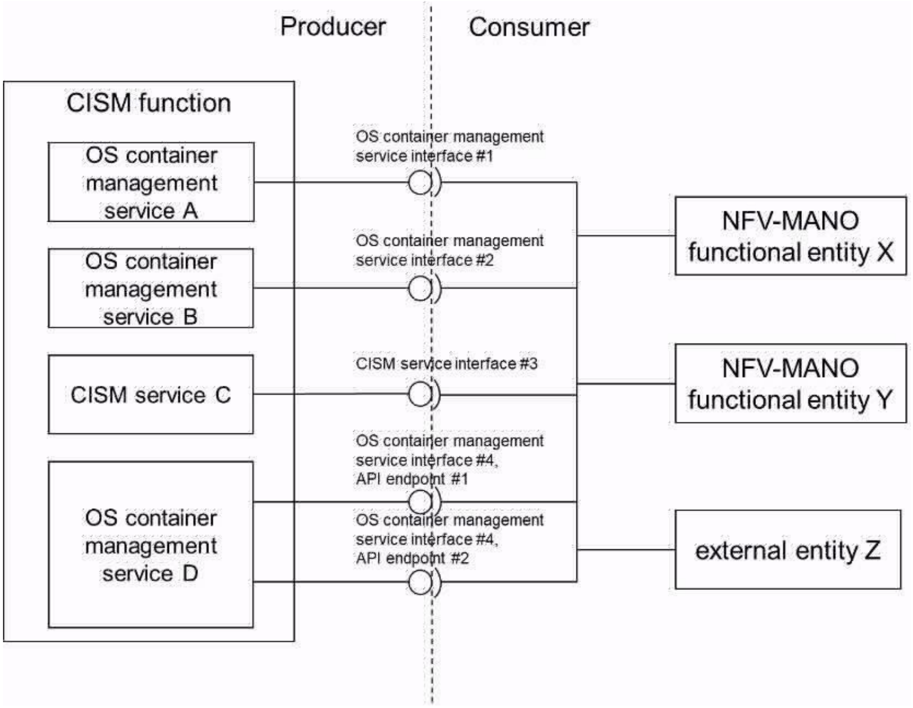

Producer
Consumer
CISM function
OS container management service A
OS container management service B
CISM service C
OS container management service D
OS container management service interface #1
OS container management service interface #2
CISM service interface #3
OS container management service interface #4, API endpoint #1
OS container management service interface #4, API endpoint #2
NFV-MANO functional entity X
NFV-MANO functional entity Y
external entity Z

Figure 3.14 - Relation overview between multiple infrastructure instances and CISM

Figure 3.15 describes in detail how such an object relation can be created, although it is necessary to design a service with such requirement (e.g., deployable at 3rd party Edge Cloud).

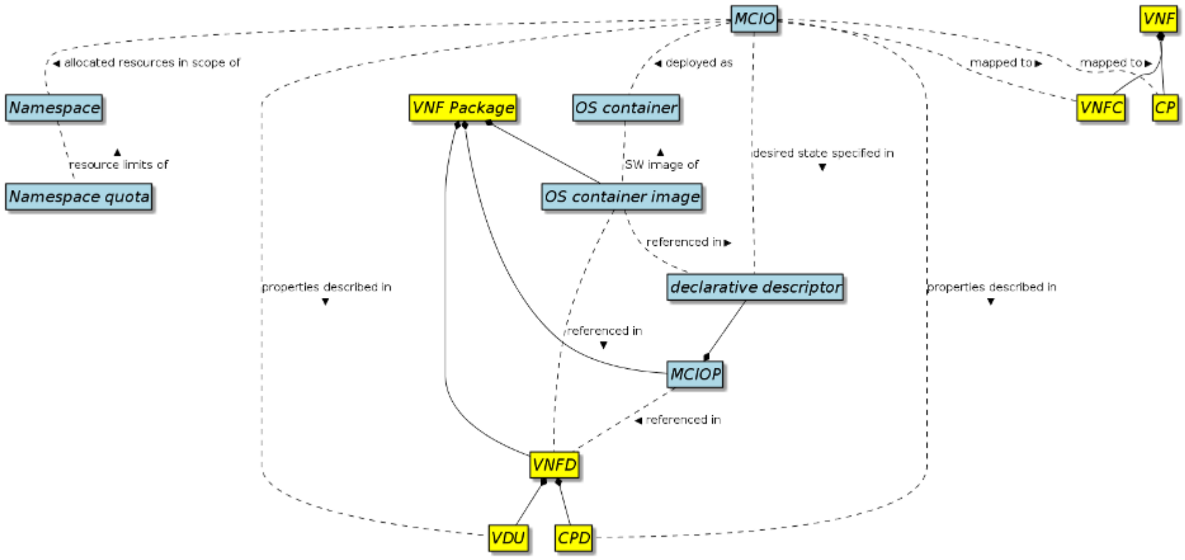


Figure 3.15 - Relation between NFV objects and OS Container Manager and Orchestration

3.8 Lessons learned from the 5G-VINNI facility sites

This section provides a view from the 5G-VINNI facilities on the experiences they have had in managing and operating network slices in support of their vertical customers.

3.8.1 Greece Facility

Currently, the Greek facility supports a single slice operation for 5G NSA as the radio nodes do not have the ability to connect to multiple MMEs (as in DÉCOR). So for 5G NSA, we cannot have slices using same radio and core. However we can have multiple slices when we offer to a vertical its own radio and its own EPC.

For 5G SA things are more difficult. Apart from the technical issues with UE registration, our 5GC does not include the NSSF. So there is only again one slice with radio and 5G core. A possibility of slicing via separating AMF, SMF and UPF per slice/vertical customer is being investigated.

We offer however slicing for the 3rd party VNFs and Network Services, since they are installed on the tenant's environment and they are attached on the user-plane path.

3.8.2 Spain Facility

The Spain facility has tested both OpenSource platforms and Ericsson's 5G NSA deployment to achieve Network Slicing at different levels. This text will be focused on the OpenSource platforms as the Ericsson's deployment is a shared resource at 5TONIC and it is not always available for verticals.

Nowadays, the Spain facility is using an OpenAir Interface radio testbed with a vEPC orchestrated by OSM SEVEN, which allows the experimentation with slices that share the same radio devices and dedicated EPC resources (dedicated MMEs). The Spain facility can provide verticals with its own EPC (Core Network Slicing with resource isolation) and a shared 5GRAN.

It is important to remark that the OpenAir Interface gNB is still under development and when configured with certain parameters the deployment becomes unstable.

On the other hand, for 5G SA the Spain facility will use Ericsson's 5G SA deployment at 5TONIC as a shared resource but the finalization date for this resources to be available is expected to be by the end of October 2020. Consequently, no experimenting has been carried out with 5G SA.

Finally, as the Greek facility, the Spain facility also offers slicing for 3rd party VNFs and Network Services.

3.8.3 UK Facility

The UK facility is currently operating NSA core supporting the 3.6GHz RAN. We are currently not able to offer true network slicing to customers as we are not able to deploy the additional VNFs needed to do so. Instead, we are currently developing a 'pseudo-slicing' option, in which separate access point names (APNs) (e.g. for eMBB, URLLC, etc.) are offered to customers. Any differentiation in these allowed network 'slices' for a particular device would be carried out by defining an appropriate profile for that IMSI at the HSS and using a particular APN on the device. There is currently no interaction between the Samsung NFVO and the Nokia Flowone application for the orchestration of these 'slices' and therefore the setup, management and control are handled manually.

For release 1, when the migration to SA core is completed, the system will be defined in a more flexible way, using NSDs and lower layer VNFs, deployed to support network slicing in a more proper sense. Service chaining to outline how the VNFs combine to provide particular slice types in given in 5G-VINNI document D3.2 [34]. It is expected also that once this capability is in place, the interaction between the NFVO and Flowone will be completed and slice instances can be set up, controlled and managed from this application.

3.8.4 Norway Facility

The Norway facility is operating three slices in NSA and two slices in SA. The NSA implementation follows the DECOR architecture and the slices successfully provided there are eMBB, MTC and, URLLC. There are some lessons learned regarding the differentiation of those NSA slices. From the computational resources, the differences can be allocated and updated to the current differentiation needs without any major issue. But from the operation point of view, the differences are more challenging to highlight. For the MTC slice, the core is provided with special IoT modules. For the

URLLC the key difference is the separation of the SPGW user and control plane that enables a more flexible deployment of potential edge sites. However, in order to provide the latest URLLC features expected by the industry, it needs to wait for the implementation of release 16 features. Therefore, current deployment and operational differentiation of slices is still susceptible to the improvements that will come with new releases.

For SA, the functions developed follow a container approach. However, such containers implementation was made on top of VMs, keeping open the question on and implementation on bare-metal which by the time constraints of the project will be difficult. One of the main challenges found and learnt in the SA deployment is the integration of orchestration systems that provide the same level of automation achieved with the NSA implementation. Basically, in the Norway SA implementation, orchestration and high level of automation is an open question, specially having as reference the good results obtained in NSA.

The Norway facility uses FlowOne as the E2E-SO according to the VINNI general architecture. At the same time this component is integrated with OpenSlice-Norway server. This tool has shown to be very useful for the automation of internal deployments. However, when it comes to the deployment of customer slices the functionality has been limited. The main reason for that is that the Slices in the Norway facility are relatively big and pre-established components. In other words, not all the coming customers will have a dedicated slice, but instead, depending on the customers' requirements, the service requested will be allocated in a sharing approach, in one of the 5 or 6 prefixed network slices of the Norway facility. Still, we want to highlight the advantages that the OpenSlice and Flowone integration will offer for the scenarios where the coming customers would need to integrate their third party VNFs with the pre-established slice assigned.

From the RAN point of view, the radio components used in 5G-VINNI Norway are based on Antenna Integrated Radios for mid-band (3600 MHz) and high-band (24.5-27.25 GHz). The radios supports 3GPP standardized 5G carrier bandwidths ranging from 20 MHz to 100 MHz for the mid-band frequencies and up to 400 MHz for the high-band frequencies. 3GPP standardized Time Division Duplex (TDD) patterns are supported. For the non-standalone deployment in 5G-VINNI, it has been decided to use 3GPP B1 (2100 MHz) as the LTE anchor band. One of the open question that still remain open is the possibility of having better slicing features at the RAN (RAN slicing), which in current 3GPP releases seems to be still limited and we hope that will be improved in the releases to come.

3.8.5 Berlin Facility

The Berlin Facility is currently operating a 5G NR SA complex testbed supporting 3.7GHz RAN. The Berlin facility concentrated on the edge-central split of the end-to-end network offering rather static slicing to the different experiments. At the current moment, a large part of the development went into the support of Kubernetes based infrastructures which enable easier deployments of infrastructure as needed by the customers of the Berlin Facility interested into its copying at different locations around the world.

For release 1, the following major unexpected elements were learned. First, the interoperability between the base stations and the core network was rather smooth, as the third party base station providers as well as the Fraunhofer FOKUS Open5GCore were implementing the 3GPP standards with rather similar interpretation. However, not the same happened with the interoperability with the UEs, where there is a very large variation in the interpretation of the different standard optional and mandatory fields as well as a less extended means to verify partial features (testing is more complicated when the different features cannot be tested in isolation).

Most of the phones using 5G NR SA require an interoperable IP Multimedia Subsystem (IMS) as to be able to automatically provide integrated voice over 5G services. This presumes not only the installation of an IMS in the testbed system (Kamailio IMS is currently used) as well as its integration with the UDR and the PCF using Diameter interfaces. This requires a rather large amount of work in backporting the specific interfaces from the 4G core. At the current moment, it is assumed that a separate IMS will be deployed for each slice. This is an easier solution as it provides the means to separate concerns. If the same IMS would be deployed, then there is a need for a more detailed development of a common IMS slice stitched to multiple core network slices.

3.8.6 Portugal Facility

The Portuguese facility site sits on top Fraunhofer Fokus' Open 5G Core (O5GC), with three instances instantiated by a SONATA Service Platform (SP), one as the 'core' and the two remaining as 'edges', and an ASOCS RAN.

SONATA's SP supports network slices as an aggregation of interconnected network services, deployed in one or more Virtual Infrastructure Managers (VIMs), either OpenStack- (VMs, which we have used in this instance) or Kubernetes-based (containers). We have described both services (core and edge) and a slice interconnecting the core with the two (similar) edges fairly quickly, taking advantage of the knowledge on SONATA previously acquired. To be able to exploit scenarios holding two edges, we had to upgrade O5GC to a more recent version. We should note that use cases to be ran in this infrastructure to not impose strict requirements in providing distinct slices to serve them.

Work with the RAN part started a bit latter, and is still in progress. Different Radio Unit (RU), Data Unit (DU) and Control Unit (CU) configurations are being tested and KPIs are being measured. The interconnection with the O5GC has also been tested and enhanced, with the help of both providers.

3.8.7 Munich Facility

The Munich experimentation facility site uses an experimental core network along with a Huawei experimental radio access. The core network implement slicing and orchestration based on network virtualization and containerization. The core network has mainly two slice modes implemented to realize traffic differentiation. One slice is for high priority traffic and another for low priority traffic. The slice two slices are able to deliver traffic and steer it based on the requirement of the application layer. In addition, a graphical user interface has been implemented to provide a method to control the different network functions and realize the core network. Using this graphical user interface it is possible to assign resources to the different function as need by the network architect. An important lesson learned in this context relates to the importance of keeping different software component up to date and in sync with the others. It is generally not easy to achieve this as the core network component relies on different open source technologies (docker, mininet, open VSwitch etc). Evidently, one has to be careful since behaviour of different complex components can be hard to troubleshoot.

3.8.8 Luxembourg Facility

The Luxembourg Experimentation Facility Site uses the Fraunhofer FOKUS' 5G Core Network solution "Open5GCore" and implements an edge-central network split where the 5G Core Network is deployed with a functional split between the edge and the core network. In particular, four slice models have been implemented for the Luxembourg Experimentation Facility Site support: centralized (direct connectivity), local offload with centralized control plane, autonomous edge node, and proxy node slides models. For further details on the experiences in managing and operating network slices based on this approach, see "Berlin Facility" section above.

4 Network slicing management interfaces

4.1 Introduction

D1.3 [2] provided a preliminary description of network slicing management interfaces, including interfaces for slice lifecycle management, interfaces for intra-slice management and inter-domain interfaces and testing interfaces. An update to that information is provided in this section.

4.2 Service Orchestration / MANO APIs

In order to fulfil the need for interoperability within a multi-vendor system, it was chosen to follow ETSI MANO standard interfaces. Several of these standard interfaces define RESTful APIs for the implementation (e.g. SOL011 [60], SOL005 [40]). Nevertheless, there are a few cases where there are no RESTful APIs specified for the interfaces. Therefore, TM Forum Open APIs were used, which also enable the interoperability between different vendor components. As an example, some of the TM Forum API were developed to guarantee the interoperability across domains, and facilities (for some of the use cases).

- Service Catalogue API (TMF633 [62]), providing artefacts for the registration and discovery of VINNI-SBs in the service catalogue, as well as capabilities for their lifecycle management (e.g. registration, deletion, updating, etc.)
- Service Ordering API (TMF641 [63]), for issuing a service order. This order conveys the information required to deploy a slice instance: selected VINNI-SB and instantiation parameters. In some cases, this instance can be modelled as a network slice subnet instance
- Service Inventory API (TMF638 [64]), which defines standardized mechanisms for CRUD operations over the records providing run-time information about the deployed slice (subnet) instances
- Service Configuration and Activation API (TMF640), providing capabilities to allow the operation of a deployed slice (subnet) instance. This includes the ability to trigger lifecycle management actions (e.g. creation, modification, update, deletion) over that instance, and the ability to define rules to collect monitoring data from that instance (e.g. using threshold-based alarms or periodic notifications).

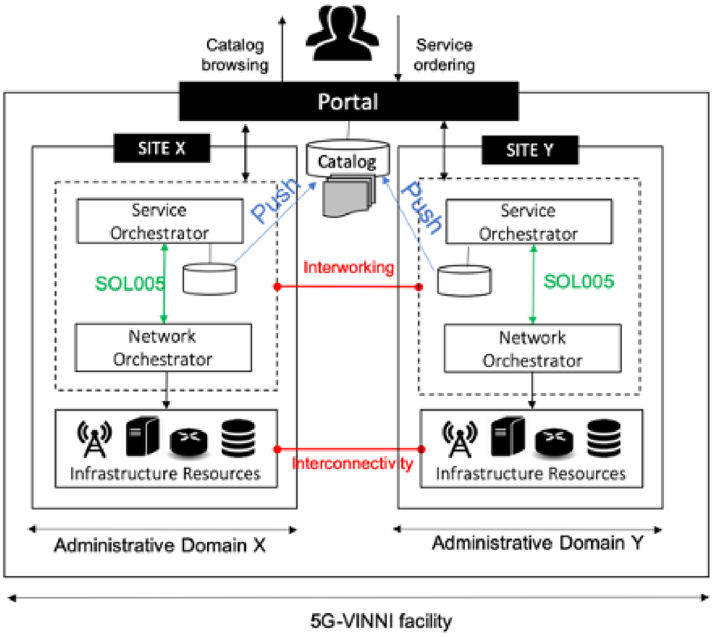


Figure 4.1 - 5G-VINNI Facility Architecture

4.3 Testing interfaces

The 5G-VINNI project is currently working on the definition of a Test as a Service (TaaS) API that will allow the integration of the testing system within the network orchestration. The purpose of the effort is allowing a direct and continuous validation of the deployed slices in an automated fashion.

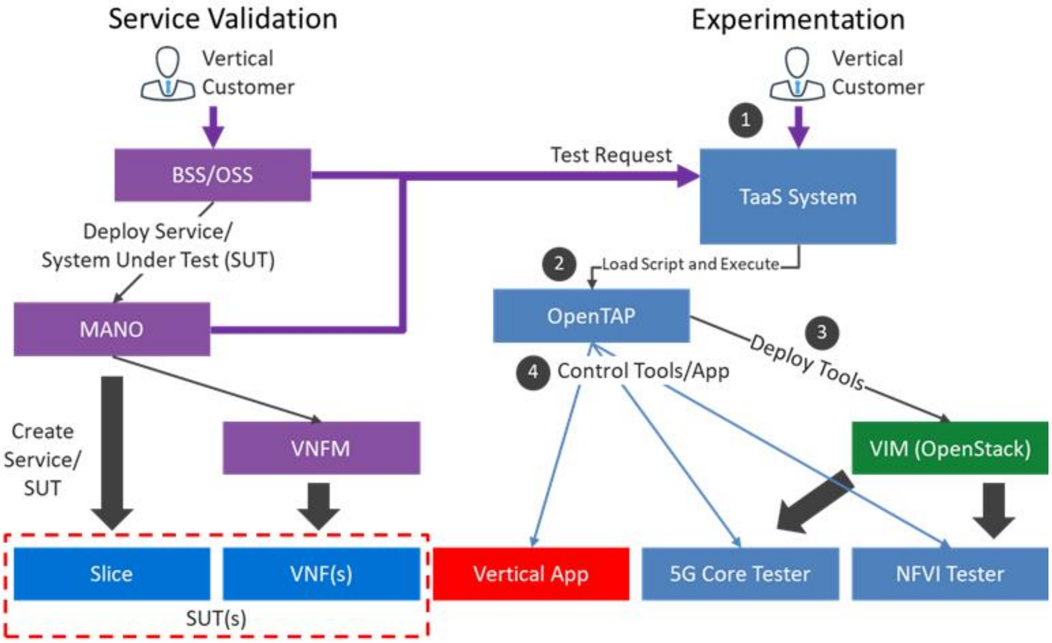


Figure 4.2 - TaaS service flow

Figure 4.2 illustrates the typical consumption flow of TaaS service in a programmatic fashion. The flow is designed in the following way:

1. A programmatic Test Request is made via API to the TaaS System. The request can be made by either the OSS/BSS or the MANO indifferently, according to the management architecture of the network. The request should contain information about the environment to be tested (or System Under Test) and which Test Case or Test Campaign needs to be executed.
2. The TaaS system, that is a coordination and management for testing services, creates an OpenTAP session and load the necessary configurations and scripts into it. OpenTAP [72] is the choice in 5G-VINNI for test sequencing and automation.
3. OpenTAP starts executing the test sequence by creating the necessary test infrastructure. In the example in Figure 4.2, the VIM is OpenStack, and it used to deploy virtual testing tools such as NFVI testers or 5G Core Network tester.
4. Finally OpenTAP starts commanding the tools according to the pre-defined test sequence.

The Testing Interface allows to make a test request (and of course receive in response information about the outcome of the test), as well as providing ancillary services such as browsing and management of the Test Case and Test Campaign repositories available in the TaaS System.

While the general flow and type of services that the interface should support are settled, the specific API design is still in progress and will be reported in D4.5. More information about the architecture and design of the TaaS system can be found in D4.1 [73] and updated in D4.2 [74].

5 Future Topics and Research

5.1 NFV MANO towards PaaS interoperability at the Edge

The *Platform-as-a-Service (PaaS)* model was defined for Cloud Computing by NIST [75] as the one of the 3 service models (Infrastructure/Platform/Software-as-a-Service) that raises management and control concerns of the underlying cloud infrastructure (compute, networking, storage etc.) for the cloud consumer. In PaaS, the consumer only controls the application deployment over an available, pre-configured environment, also setting the hosting environment configurations and the application provisioning parameters. Thus, PaaS fosters reusability of services and deployment speed, by raising the abstraction of service modelling to a usable yet flexible level.

The enhancement of NFV MANO towards incorporation of a Platform-as-a-Service model can provide significant benefits in terms of automation, interoperability and maintainability [15]. PaaS services can be VNF services, either common or dedicated to specific Consumer VNFs. Through their integration to NFV MANO, it is possible to achieve rapid service instantiation and automation of service management, while respecting tenant isolation. In this model, PaaS services can expose common and open APIs and can be invoked and utilized by different NFVI providers and administrative domains eliminating the setup/integration/management overhead for NFVI platforms. Service registration and discovery mechanisms can be employed to manage and maintain consumption relationships, providing the means for seamless PaaS and Consumer VNF service binding. Moreover, PaaS services' flexibility and adaptability can be exploited towards development of mechanisms for effectively responding to fluctuating and diverse service demand.

In our approach, we consider an interoperable scheme of Cloud/Core NFV MANO entities with Edge Platforms, the latter being responsible for the management of Cloud-native Network Functions (CNFs), including Edge PaaS services. Hence, NFV MANO and Edge Platforms are linked to different orchestration scopes, but through their coordination, it is possible to manage Network Slices that extend end-to-end from Cloud/Core NFVI, to Edge Points of Presence (PoP). This is depicted in Figure 5.1, where Network Slice Templates (NST) include Cloud/Core VNFs, as well as Edge PoP CNFs that are either custom services of the Network Slice (MEC App CNF) or instantiations of PaaS services (PaaS service CNF).

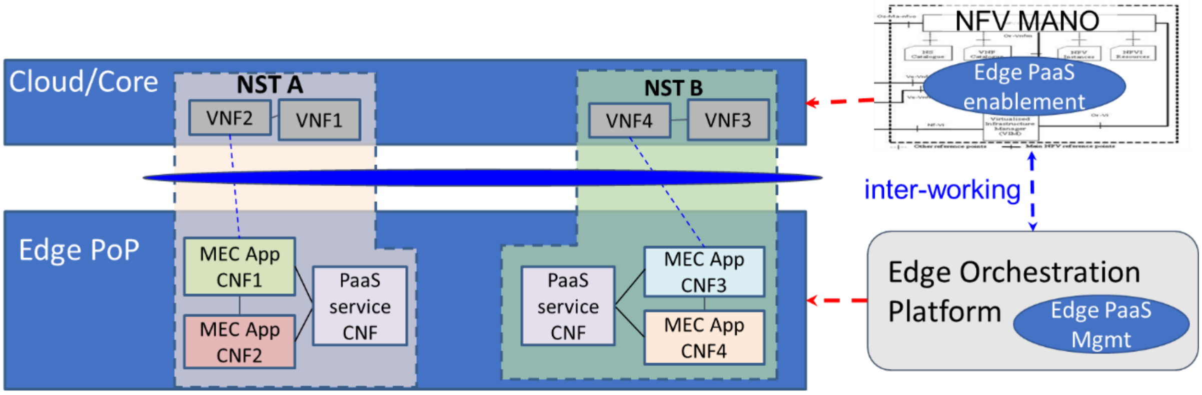


Figure 5.1 - Edge PaaS-aware NFV MANO

According to our approach, Edge PoPs retain autonomy and can interoperate with multiple MANO entities, allowing flexibility, efficiency and scalability in cross-domain Network Slicing that extends to the Edge. In addition to PaaS automation benefits at the Edge, the main advantage of this scheme is that Edge infrastructure and service layer management is decoupled from Cloud/Core-level MANO, simplifying operations and enabling dynamic producer-consumer relationships between orchestration entities. Moreover, entry barriers are lowered to 3rd party edge resource providers, who can expose Edge resources to MANO as platform services with usable, common and open APIs.

At the same time, Total Cost of Ownership is minimized for M(V)NOs, who can seamlessly consume common PaaS edge services from different edge resource providers at different edge PoPs, without having to worry about underlying edge infrastructure management.

As an interesting use case (illustrated in Figure 5.2), we consider Edge Function-as-a-Service scenarios, in which Edge PoPs are equipped with libraries of reusable Edge Services (Function Stores), which are vertical-specific (e.g. common libraries/functions for Smart Factory and Smart city domains). In this setup, end-to-end Network Slices can be devised with VNF services at the Cloud/Core, combined with mixed compositions of PaaS services and custom slice-specific network services at the Edge. Thus, Service Function Chaining is achieved at the Edge in an automated fashion, with services being both reusable functions of concrete domains (customization domains) and custom service functions that cover the slice-specific requirements.

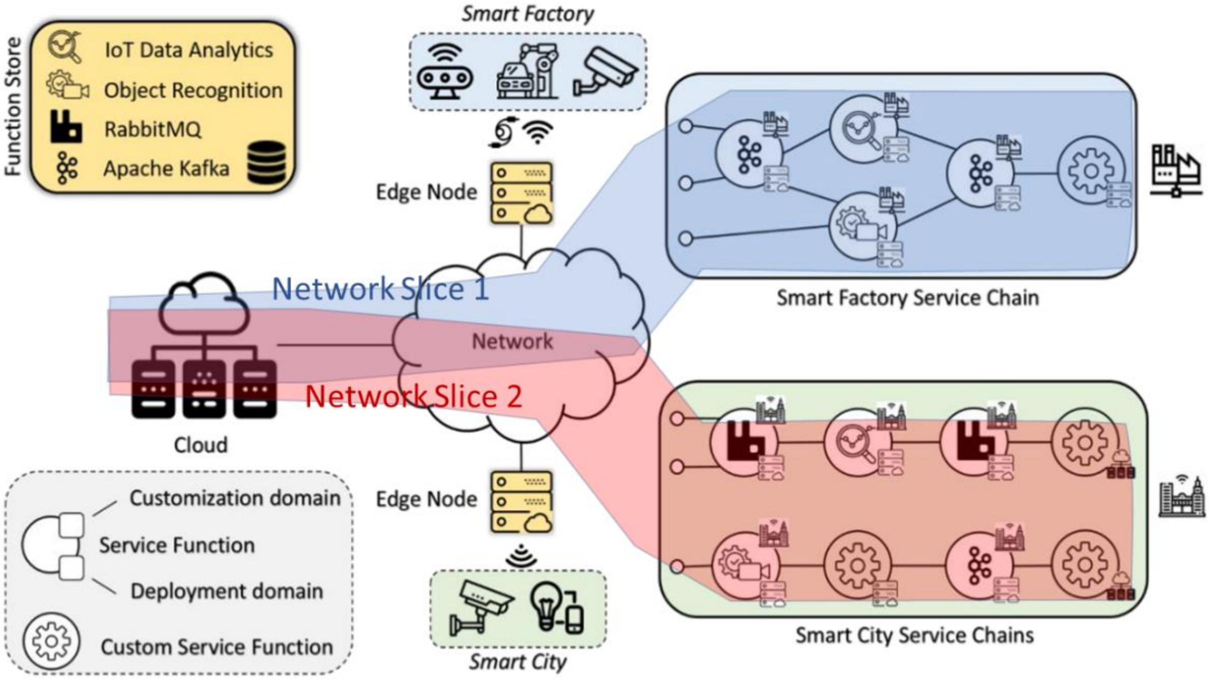


Figure 5.2 - Edge Function-as-a-Service with Edge PaaS CNFs

6 Evolution roadmap and future work

This section uses the same evolution reference presented in D1.3 and in Figure 6.1 below. However there are some updates regarding the scope of the implementation of such schemes in 5G-VINNI, as well as in pointing more specifically the challenges and implementations that may come in the future in this context.

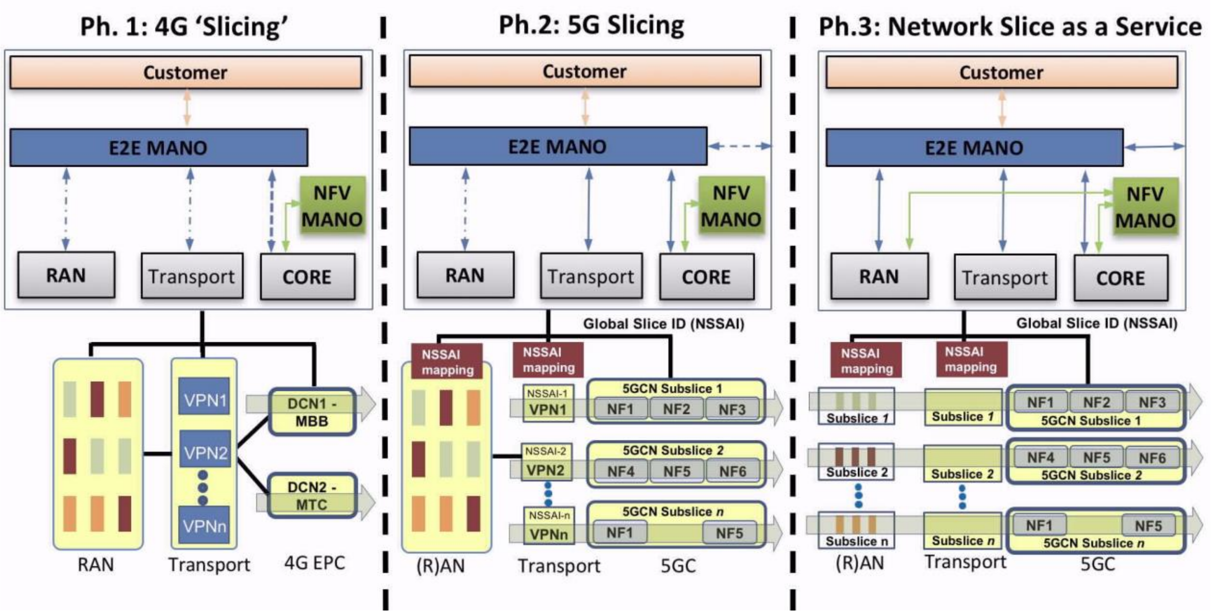


Figure 6.1 - 5G-VINNI Evolution Roadmap (LCM and Orchestration emphasis)

Starting with Phase2, the main change is the introduction of Stand Alone (SA) operation, and at the management level, the link between E2E-SO and CORE, and between E2E-SO and Transport respectively. Although the implementation of Stand Alone in the facilities presents some challenges, its implementation can be achieved. However, from the management and orchestration part, an open the full integration of SA with the entire VINNI management suite, composed by Domain Controllers, NFV-MANO and E2E-SO Orchestration, is still uncertain. The main reason is that most SA implementations are container-based, where the Orchestration is mainly based on Kubernetes. The integration of Kubernetes with the already defined management and orchestration is an ongoing work, which most likely won't be implemented in VINNI, but it represents one of the main future tasks to keep in target.

In phase 3, the main features introduced are a more mature management and orchestration, and better slicing at the RAN and Transport, where management and orchestration play again a big role.

The support of slicing at the RAN is still an evolving issue, with some progress in current 3GPP release 16, as mentioned in the 5G-VINNI Deliverable D1.4 [3]. In addition to this, the support of "bandwidth parts" enables a single carrier to divide bandwidth into a set of bandwidth parts, each defining a specific numerology [76]. This was proposed in Release 15 and enhanced in Release 16. It may be tested now, and it is in the plan of 5G-VINNI. However it will depend on the available time after some prioritization policies in the last phase of 5G-VINNI.

Another important feature that will be studied (but most likely not implemented) in 5G-VINNI is the RAN location based slicing, in which slicing is executed based on a specific tag and specific area. Finally, slices need to be differentiated by the security needs, where the Subscription Concealed Identifier (SUCI) may be useful. This feature will be initially be implemented in a static way, but it is expected that this feature can be automated by the 5G management and orchestration suite.

In the context of this document, the integration of RAN management (including slicing features, bandwidth parts, location based, SUCI, etc) in order to enable control from a central entity such as the E2E-SO, is still an open issue that will present big difficulties in the implementation during the last part of 5G-VINNI. However, this represents at the same time one of the most relevant topics to be addressed by future works.

Another important remaining part is the transport network. Support of enhanced manageability in the transport network, as defined in D1.4 Section 6.2 “Backhaul Automation”. As mentioned there, the network automation of transport network can be harmonized with RAN and CORE using SDN. However, the scope goes beyond, since it is mentioned that the programmability and flexibility of SDN is not enough and further enhancements are needed, such as those mentioned in D1.4. Work to address all these challenges has been undertaken in several standards organizations, such as ETSI Experiential Networked Intelligence Industry Specification Group (ISG ENI). For a perfect integration of such enhancements in the transport network, the evolution and adaptation of the management and orchestrations systems is fundamental.

7 Conclusions

Network slicing is one of the key ingredients of 5G, particularly to meet the wide range of requirements of next generation of mobile network services. In particular, network slicing will enable dynamically customised (and private/isolated) services for vertical markets in a cost efficient way. In this environment, orchestration and management will surely represent a key challenge. For 5G-VINNI this has been an important part of the work conducted in facility sites.

In this report, several aspects of orchestration and management of network slicing have been analysed. Some of the key takeaways are:

- In 5G-VINNI, network slice lifecycle follows the 3GPP vision, except for the preparation phase, in which the experiment-driven definition of 5G-VINNI network slices requires additional testing and validation activities before making the service offerings publicly available in the 5G-VINNI service catalogue.
- Multiple deployment models are possible in relation to management and orchestration of Edge Clouds, ranging from fully centralized to fully distributed management. The choice of the most appropriate orchestration model depends on several factors, including the type of service (eMBB, URLLC, mMTC) and the level of dispersion of edge infrastructure components.
- Integration of Cloud Native environments is a fundamental requirement to be tackled by CSPs and vendors for 5G. This has been already under study by relevant standardization groups, especially ETSI NFV and new approaches are required in relation to aspects such as instantiation, management and orchestration of network services. Thus, the compliance of 5G-VINNI Ecosystem with ETSI NFV raises new requirements, including M&O of virtualised containers and services exposed to the NFVO.
- The integration of 5G-VINNI vertical customers should enable the ability to securely expose management capabilities of 5G-VINNI facility towards authorized customers, based on different capability exposure levels, exposing a different set of operational capabilities from the 5G-VINNI facility. For a fine-grained control of this capability exposure across public and private NOPs, 5G-VINNI may use token-based authentication to define the set of management services that can be consumed at operation time.
- To enable the implementation of flexible security policies and the separation of the NFV environment in different security zones that limits the communication between specific functions based on predefined security needs, one of the new features required from management and orchestration is the automated creation and enforcement of rules in the firewalls by the E2E-SO at the time of the slice deployment or individual VNF deployment.
- The automated management of the distributed network infrastructure across a potentially high number of edge nodes requires a new network management approach, in which AI is likely to play an increasingly significant role.
- Cross-domain orchestration is required whenever 5G capabilities for advanced vertical experimentation require network components placed in multiple facility sites. Three federation options have been considered by 5G-VINNI: (i) SO-SO: the SOs from different sites exchange information and expose their capabilities across them; (ii) NFVO-NFVO: the NFVOs from different sites exchange information and expose their capabilities across them; (iii) SO-NFVO: the SO from one site communicates with the NFVO from another site. Out of these three options, SO-SO is considered the most realistic solution for 5G commercial networks, and thus it is the one explored in 5G-VINNI.
- The 5G-VINNI evolution roadmap includes three main phases, already outlined in previous 5G-VINNI deliverables, namely 4G slicing, 5G slicing and Network Slice as Service. Management and orchestration represent a fundamental ingredient of this evolution, in relation to which a number of challenges will have to be handled - e.g. container orchestration, RAN slicing, backhaul automation.

In addition, the practical deployment of 5G-VINNI facility sites has provided useful lessons about the management and operation of a 5G network infrastructure and how to overcome the challenges related to immaturity of certain components.

At the time of writing, work is ongoing at the 5G-VINNI facility sites and will continue in the next few months. It is expected that the management and orchestration principles outlined in this report will be adopted and possibly extended by the different 5G-VINNI facility sites, either main or experimental, until the end of the project.

References

[1] 5G-VINNI, “Grant Agreement 815279 - 5G Verticals INNovation Infrastructure,” 2018.

[2] 5G-VINNI, “Deliverable D1.3 - Design for systems and interfaces for slice operation v1,” 2019. [Online]. Available: https://zenodo.org/record/3345599#.X8p5K2j7Q_4. [Accessed November 2020].

[3] 5G-VINNI, “Deliverable D1.4 - Design of infrastructure architecture and subsystems v2,” 2020. [Online]. Available: https://zenodo.org/record/4066381#.X8p6sGj7Q_4. [Accessed November 2020].

[4] 5G-VINNI, “Deliverable D1.1 - Design of infrastructure architecture and subsystems v1,” 2018. [Online]. Available: https://zenodo.org/record/2668754#.X8p7Vmj7Q_4. [Accessed November 2020].

[5] 5G-VINNI, “Deliverable D1.5 - 5G-VINNI E2E Network Slice Implementation and Further Design,” 2020. [Online]. Available: https://zenodo.org/record/4067793#.X8p7q2j7Q_4. [Accessed November 2020].

[6] 5G-VINNI, “Deliverable D1.2 - Design of network slicing and supporting system v1,” 2019. [Online]. Available: https://zenodo.org/record/2668763#.X8p8UGj7Q_4. [Accessed November 2020].

[7] 5G-VINNI, “Deliverable D3.1 - Specification of services delivered by each of the 5G-VINNI facilities,” 2019. [Online]. Available: https://zenodo.org/record/3345612#.X2Sp9i-w1TY. [Accessed September 2020].

[8] Bell Labs Consulting , “Future X Network Cost Economics —A network operator’s TCO journey through virtualization, automation, and network slicing,” 2018.

[9] J. Ping, “Network Resource Model for 5G Network and Network Slice,” 2019. [Online]. Available: https://www.riverpublishers.com/journal/journal_articles/RP_Journal_2245-800X_724.pdf. [Accessed December 2020].

[10] ETSI ZSM ISG, “ETSI GS ZSM 003; Zero-touch network and Service Management (ZSM); End to end management and orchestration of network slicing,” 2020.

[11] 3GPP, “TS 28.531 V16.1.0, Technical Specification Group Services and System Aspects; Management and orchestration; Provisioning;,” 2019.

[12] 3GPP, “TS 28.550 V16.1.0 Management and orchestration; Performance assurance (Release 16),” 2019.

[13] 3GPP, “TS 28.545 V16.1.0 Management and orchestration; Fault Supervision (FS); (Release 16),” 2020.

[14] ETSI ZSM ISG, “ETSI GS ZSM 002 V1.1.1; Zero-touch network and Service Management (ZSM); Reference Architecture,” 2019.

[15] ETSI NFV ISG, “ETSI GR NFV-IFA 029 V3.3.1 - Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS",” 2019.

[16] Analysis Mason, “Acceleration technologies: realizing the potential of network virtualization,”

2019.

[17] R. Rokui, S. Homma, K. Makhijani, L. Contreras and J. Tantsura, “IETF Definition of Transport Slice, IETF Draft, draft-nsdt-teas-transport-slice-definition-04,” 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-nsdt-teas-transport-slice-definition/. [Accessed October 2020].

[18] E. Gray and J. Drake, “IETF Draft, draft-nsdt-teas-ns-framework-04, Framework for Transport Network Slices,” 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-nsdt-teas-ns-framework/. [Accessed October 2020].

[19] L. M. Contreras, S. Homma and J. Ordonez Lucena, “Considerations for defining a Transport Slice NBI, IETF Draft draft-contreras-teas-slice-nbi-00,” 2019. [Online]. Available: https://datatracker.ietf.org/doc/draft-contreras-teas-slice-nbi/. [Accessed October 2020].

[20] X. Liu, J. Tantsura, I. Bryskin, L. Contreras, Q. Wu, S. Belotti and R. Rokui, “Transport Network Slice YANG Data Model, IETF Draft draft-liu-teas-transport-network-slice-yang-01,” 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-liu-teas-transport-network-slice-yang/. [Accessed October 2020].

[21] B. Wu, D. Dodhy, L. Han and R. Rokui, “A YANG Data Model for Transport Slice NBI, IETF Draft draft-wd-teas-transport-slice-yang-02,” 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-wd-teas-transport-slice-yang/. [Accessed October 2020].

[22] S. Peng, R. Chen, G. Mirsky and F. Qin, “IETF Draft draft-peng-teas-network-slicing-04 - Packet Network Slicing using Segment Routing,” 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-peng-teas-network-slicing/. [Accessed October 2020].

[23] Z. Zhang , S. Hegde and A. Gulko, “IETF Draft draft-zzhang-teas-network-slicing-with-flex-te-00 - Network Slicing with Flexible Traffic Engineering,” 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-zzhang-teas-network-slicing-with-flex-te/. [Accessed October 2020].

[24] Broadband Forum, “SD-406 End-to-End Network Slicing,” 2018. [Online]. Available: https://wiki.broadband-forum.org/display/BBF/SD-406+End-to-End+Network+Slicing. [Accessed October 2020].

[25] Broadband Forum, “SD-407 5G Fixed Mobile Convergence Study,” 2020. [Online]. Available: https://wiki.broadband-forum.org/display/BBF/SD-407+5G+Fixed+Mobile+Convergence+Study. [Accessed September 2020].

[26] Metro Ethernet Forum, “MEF 55; Lifecycle Service Orchestration (LSO): Reference Architecture and Framework,” 2019.

[27] MEF, “Slicing for Shared 5G Fronthaul and Backhaul,” 2020. [Online]. Available: https://www.mef.net/wp-content/uploads/2020/04/MEF-white-paper-Slicing-for-Shared-5G-Fronthaul-and-Backhaul.pdf. [Accessed November 2020].

[28] ONF, “OpenNetworkingFoundation/TAPI,” 2020. [Online]. Available: https://github.com/OpenNetworkingFoundation/TAPI. [Accessed November 2020].

[29] MEF, “MEF 60, Network Resource Provisioning Interface Profile Specification,” 2018. [Online]. Available: https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_60.pdf. [Accessed November 2020].

[30] GSMA, “An Introduction to Network Slicing,” 2017. [Online]. Available:

https://www.gsma.com/futurenetworks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf. [Accessed September 2020].

[31] GSMA, “Network Slicing Use Case Requirements,” 2018. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2018/03/Network-Slicing-Use-Cases-Requirements-Wrapper.pdf. [Accessed September 2020].

[32] GSMA, “Generic Network Slice Template v3.0,” 2020. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads/NG.116-v3.0.pdf. [Accessed September 2020].

[33] GSMA, “From Vertical Industry Requirements to Network Slice Characteristics,” 2018. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2018/09/5G-Network-Slicing-Report-From-Vertical-Industry-Requirements-to-Network-Slice-Characteristics.pdf. [Accessed December 2020].

[34] 5G-VINNI, “Deliverable D3.2 - Publication of first version of service catalogues to vertical consumers,” 2019. [Online]. Available: https://zenodo.org/record/3345620#.X47Bw9BKjGg. [Accessed September 2020].

[35] O-RAN Alliance, “O-RAN Alliance,” 2020. [Online]. Available: https://www.o-ran.org/. [Accessed October 2020].

[36] O-RAN Alliance, “O-RAN Use Cases and Deployment Scenarios,” 2020.

[37] O-RAN Alliance, “O-RAN.WG1.Slicing-Architecture-v01.00 O-RAN Working Group 1 Slicing Architecture,” 2020.

[38] RCRWireless News, “Open RAN 101–Role of RAN Intelligent Controller: Why, what, when, how? (Reader Forum),” [Online]. Available: https://www.rcrwireless.com/20200730/opinion/readerforum/open-ran-101-role-of-ran-intelligent-controller-why-what-when-how-reader-forum.

[39] Open Source MANO, “OSM#9 Hackfest, Hack 0: Introduction to NFV and OSM,” 2020. [Online]. Available: http://osm-download.etsi.org/ftp/osm-7.0-seven/OSM9-hackfest/presentations/OSM%239%20Hackfest%20-%20HD0.0%20Introduction%20to%20NFV%20and%20OSM.pptx.pdf. [Accessed October 2020].

[40] ETSI NFV ISG, “ETSI GS NFV-SOL 005 V2.4.1 Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point,” 2018.

[41] OSM community, OSM Technical Steering Committee, “OSM Release EIGHT – Release Notes,” 2020. [Online]. Available: https://osm.etsi.org/wikipub/images/5/56/OSM_Release_EIGHT_-_Release_Notes.pdf. [Accessed September 2020].

[42] Open Source MANO, “"OSM#9 Hackfest, Hack 1: Architecture & Installation”,” 2020. [Online]. Available: http://osm-download.etsi.org/ftp/osm-7.0-seven/OSM9-hackfest/presentations/OSM%239%20Hackfest%20-%20HD0.1%20OSM%20Architecture%20&%20Installation.pdf. [Accessed October 2020].

[43] ONAP, “ONAP - Open Network Automation Platform,” 2020. [Online]. Available: https://wiki.onap.org/. [Accessed October 2020].

[44] ONAP, “ONAP Use Cases and Blueprints,” 2020. [Online]. Available: https://www.onap.org/architecture/use-cases-blue-prints. [Accessed October 2020].

[45] ONAP, " ONAP Developer Wiki," 2020. [Online]. Available: https://wiki.onap.org/. [Accessed October 2020].

[46] 3GPP, "3GPP TS 28.530 V16.3.0; Technical Specification Group Services and System Aspects; Management and orchestration; Concepts, use cases and requirements (Release 16)," 2020.

[47] ETSI NFV ISG, "ETSI GS NFV 002 V1.2.1 Network Functions Virtualisation (NFV); Architectural Framework," 2014.

[48] Ericsson, "NFVI evolution," 2019. [Online]. Available: https://www.ericsson.com/4a46af/assets/global/qbank/2019/08/23/nfvi-evolution-110530resize1520847crop001500844autoorientquality90stripbackground23ffffffextensionjpgid8.jpg?w=1212. [Accessed October 2020].

[49] ETSI NFV ISG, "GS NFV-MAN 001, "Network Functions Virtualisation (NFV); Management and Orchestration", v1.1.1," 2014.

[50] ETSI NFV ISG, "ETSI GS NFV-EVE 004 V1.1.1 Network Functions Virtualisation (NFV); Virtualisation Technologies; Report on the application of Different Virtualisation Technologies in the NFV Framework," 2016.

[51] OSM, "Open Source MANO's documentation," 2020. [Online]. Available: https://osm.etsi.org/docs/user-guide/15-k8s-installation.html. [Accessed December 2020].

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

[53] ETSI NFV ISG, "ETSI GS NFV-IFA 036 V0.0.4; Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Specification of requirements for the management and orchestration of container cluster nodes," 2020.

[54] ETSI NFV ISG, "ETSI GS NFV-IFA 040 V4.1.1; Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification," 2020.

[55] Prometheus, "Prometheus Overview," 2020. [Online]. Available: https://prometheus.io/docs/introduction/overview/. [Accessed October 2020].

[56] Elasticsearch, "The heart of the free and open Elastic Stack," 2020. [Online]. Available: https://www.elastic.co/elasticsearch/. [Accessed October 2020].

[57] Influx , "Influx DB 1.8 documentation," 2020. [Online]. Available: https://docs.influxdata.com/influxdb/v1.8/. [Accessed October 2020].

[58] Apache NiFi , "Apache NiFi Overview," 2020. [Online]. Available: https://nifi.apache.org/docs.html. [Accessed October 2020].

[59] Drools Fusion, "Drools Fusion User Guide," October 2020. [Online]. Available: https://docs.jboss.org/drools/release/5.3.0.Final/drools-fusion-docs/html_single/.

[60] ETSI NFV ISG, "ETSI GS NFV-SOL 011 V3.3.1 Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Or-Or Reference Point," 2020.

[61] TM Forum, "TM Forum, GB992 Open API Map R18.0.1," 2018.

[62] TM Forum, "TMF633 Service Catalog API REST Specification R18.5.1," 2019.

[63] TM Forum, “TMF641 Service Ordering API REST Specification R18.5.1,” 2020.

[64] TM Forum, “TMF638 Service Inventory Management API REST Specification R16.5.1,” 2017.

[65] TM Forum, “TMF640 Activation and Configuration API REST Specification R15.5.1,” 2016.

[66] ETSI NFV ISG, “ETSI GS NFV-SOL 006 V2.7.1 Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; NFV descriptors based on YANG Specification,” 2019.

[67] Nokia, “Flowone - Service fulfillment and orchestration,” 2020. [Online]. Available: https://www.nokia.com/networks/solutions/flowone/. [Accessed October 2020].

[68] Nokia, “CloudBand Network Director - Automate network services delivery and operation,” 2020. [Online]. Available: https://www.nokia.com/networks/products/cloudband-network-director/. [Accessed December 2020].

[69] Dell'Oro Group, “5G Core – Are We Ready?,” 2020. [Online]. Available: https://www.delloro.com/5g-core-are-we-ready/. [Accessed December 2020].

[70] ETSI MEC ISG, “ETSI GR MEC 027 V2.1.1¸Multi-access Edge Computing (MEC); Study on MEC support for alternative virtualization technologies,” 2019.

[71] ETSI NFV ISG, “ETSI GR NFV-IFA 029 V3.3.1; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS",” 2019.

[72] OpenTap, “OpenTap - Test Automation Project,” 2020. [Online]. Available: https://www.opentap.io/. [Accessed December 2020].

[73] 5G-VINNI, “Deliverable D4.1 - Initial report on test-plan creation and methodology, and development of test orchestration framework,” 2019. [Online]. Available: https://zenodo.org/record/3345626#.X5WZC4j7Q_4. [Accessed October 2020].

[74] 5G-VINNI, “Deliverable D4.2 - Intermediate report on test-plan creation and methodology, and development of test orchestration framework,” 2020.

[75] NIST, “NIST Special Publication 800-146 Cloud Computing Synopsis and Recommendations,” 2012. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-146.pdf. [Accessed October 2020].

[76] C. Sexton, N. Marchetti and L. DaSilva, “Customization and Trade-offs in 5G RAN Slicing,” *IEEE Communications Magazine ,* pp. 116-122, April 2019.