Grant Agreement No.: 957216
Call: H2020-ICT-2018-2020

Topic: ICT-56-2020
Type of action: RIA

D4.4 Service orchestration at the edge

Revision: v1.0

Work package	WP 4
Task	Task 4.3
Due date	31/03/2022
Submission date	31/03/2022
Deliverable lead	Nextworks
Version	1.0
Authors	Pietro Piscione (NXW), Giacomo Bernini (NXW), Erin Seder (NXW), Francisco Javier Curieses Sanz (UPV), Nuria Molner (UPV), Roberto Bomfin (TUD), Jose Costa Raquena (CMC), Joe Cahill (iDR), Shane Bunyan (iDR), Christos Politis (SES)
Reviewers	Michael Roitzsch (BI), Nils Asmussen (BI), Jose Costa Raquena (CMC), Gino Ciccone (TEI), Roberto Bomfin (TUD), Nuria Molner (UPV), Francisco Javier Curieses Sanz (UPV), Tadeusz Puźniakowski (PJATK)

Co-funded by the Horizon 2020 Framework Programme of the European Union

Abstract	This document presents the design and preliminary prototype implementation of the iNGENIOUS service and network slice orchestration framework. The proposed solution is aligned with relevant 3GPP and ETSI standard architectures, and is described in terms of architectural principles, functional architecture, information models and APIs. The integration of AI/ML capabilities is also described to achieve full automation in the operation of 5G network slices supporting industrial IoT and supply chain services.
Keywords	5G, network slices, edge, orchestration, AI, ML

Document Revision History

Version	Date	Description of change	List of contributor(s)
V1.0	31/03/2022	EC version	NXW

Disclaimer

This iNGENIOUS D4.4 deliverable is not yet approved nor rejected, neither financially nor content-wise by the European Commission. The approval/rejection decision of work and resources will take place at the Mid-Term Review Meeting planned in June 2022, after the monitoring process involving experts has come to an end.

The information, documentation and figures available in this deliverable are written by the "Next-Generation IoT solutions for the universal supply chain" (iNGENIOUS) project's consortium under EC grant agreement 957216 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

Copyright notice

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to iNGENIOUS project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Executive Summary

This deliverable presents the design and preliminary prototype implementation of the iNGENIOUS network slice orchestration framework. The work reported covers specific functionalities within the iNGENIOUS Network Layer, and that relates to the management and orchestration features required for the coordination of the relevant 3GPP and non-3GPP network technologies developed in the project. The main aim is to provide automation in the deployment, provisioning and operation of end-to-end network slices in support of industrial IoT and supply chain 5G services.

The proposed orchestration framework is aligned with the relevant 3GPP and ETSI standard specifications for 5G network slice management. Specifically, the orchestration of 5G radio access, transport and core network domains allow to coordinate resources (network, computing, storage) and network functions to instantiate and compose vertical services and network slices over a shared infrastructure. The iNGENIOUS end-to-end network slice orchestration framework provide a seamless integration and coordination with edge computing infrastructures to support ultra-low latency applications. For this, the proposed architecture provides a truly end-to-end orchestration of resources that considers the edge of the network as key for the deployment of network functions and applications.

A detailed description of the iNGENIOUS end-to-end network slice orchestration framework architecture is provided in this document, which includes architectural principles, functional decomposition, applicable information models and exposed interfaces. Moreover, the pure slice management and orchestration functionalities are augmented with closed-loop capabilities, aiming at a high degree of automation in service and network slice operation leveraging on AI and ML techniques.

The document also reports on the preliminary software prototype of the iNGENIOUS end-to-end network slice orchestration framework, which makes use and enhances existing opensource tools and platforms. Moreover, the results of preliminary integration activities are reported, with the aim of validating the functionalities of the developed orchestration components against the relevant iNGENIOUS 5G-IoT network technologies (the 5G Core, the Flexible PHY/MAC and the O-RAN near real time controller).

Co-funded by the Horizon 2020
Framework Programme of the European Union

Table of Contents

1 Introduction 10
1.1 Objective of this deliverable 10
1.2 Positioning in the iNGENIOUS architecture 11
1.3 Structure of the document 12
2 Related Work 13
2.1 Management and orchestration of 5G networks 13
2.2 5G network slices 14
3 The 5G-IoT Network 21
3.1 Radio Access Network 21
3.2 5G Core 22
3.3 Backhaul and Transport Networks 29
4 End-to-end Network Slice Orchestration Architecture 35
4.1 Architectural Principles 35
4.2 Functional Architecture 36
4.3 AI/ML based network slice optimization 49
4.4 Network Slice Orchestration Workflows 52
4.5 Interfaces and APIs 56
5 Preliminary Prototype and Early Integration 62
5.1 Software prototype description 62
5.2 Integration with 5G Core 66
5.3 Integration with Flexible PHY/MAC 70
5.4 Integration with O-RAN 72
6 Relation to UCs and Main Innovations 75
6.1 End-to-end network slice orchestration innovations 75
6.2 Relation to UCs 76
7 Conclusions and Next Steps 79

Co-funded by the Horizon 2020 Framework Programme of the European Union

List of figures

Figure 1-1 - Overview of the iNGENIOUS architecture (D2.2) 11
Figure 1-2 - Highlight on the iNGENIOUS network layer components 12
Figure 2-1 - Mobile network management architecture – Interaction between 3GPP management system and NFV-MANO framework [2] 13
Figure 2-2 - Multiple instances of communication services deployed over different network slice instances [7] 15
Figure 2-3 - Example of network slice instances [7] 16
Figure 2-4 - Structure of network slices and network slice subnets in network services and virtual network functions [8] 17
Figure 2-5 - Information model of a Network Service Descriptor [9] 18
Figure 2-6 - Role of GST and NEST as input for the preparation of a NSI [11] 18
Figure 2-7 - High-level architecture of management functions for network slicing [12] 19
Figure 2-8 - Hierarchical interaction between NSMF and per-domain NSSMFs [13] 20
Figure 2-9 - Example of management system deployment with NSSMF interacting with NFVO [13] 20
Figure 3-1 - Flexible PHY/MAC frame structure. 21
Figure 3-2 - Network slicing for different machine type communications 23
Figure 3-3 - Network slicing management modules 23
Figure 3-4 - Network slice activation flow 24
Figure 3-5 - Network Data Analytic Function modules design. 27
Figure 3-6 - Interaction between 3GPP and TN management systems [7] 30
Figure 3-7 - Intra-Domain SDN Transport Network 31
Figure 3-8 - Inter-Domain SDN Transport Network 31
Figure 3-9 - Satellite backhaul connectivity architecture 33
Figure 3-10 - Distributed SDN Controller Architecture 34
Figure 4-1 - End-to-end network slice architecture functional high-level architecture supported by AI/ML platform 36
Figure 4-2 - High level software architecture end-to-end network orchestration framework 37
Figure 4-3 - Network Slice Simplified Class Diagram 39
Figure 4-4 - Translation of Vertical Service Requirements into end-to-end slice resource allocation 39
Figure 4-5 - High-level software architecture of VSMF 40
Figure 4-6 - High-level architecture of NSMF 42
Figure 4-7 - High-level architecture of NSSMF layer 44
Figure 4-8 - High-level architecture of generic NSSMF 44

Co-funded by the Horizon 2020 Framework Programme of the European Union

Figure 4-9 - AI/ML and monitoring platform functional architecture....47
Figure 4-10 - Closed-loop pre-emptive auto-scaling of UPF....50
Figure 4-11 - Data source available so far for the monitoring platform....51
Figure 4-12 - High-level sequence diagram of end-to-end network slice provisioning....53
Figure 4-13 - High-level workflow of end-to-end network slice termination....54
Figure 4-14 - High-level workflow of end-to-end network slice re-configuration 55
Figure 5-1 - VSMF implementation diagram....63
Figure 5-2 - Implementation diagram of NSMF....64
Figure 5-3 - High-level workflow of manual deployment of a 5G network....67
Figure 5-4 - Logs of day-1 configuration stages....68
Figure 5-5 - Snippet of Python script of day-1 configuration of 5G Core instance 69
Figure 5-6 - Edge-core deployment of 5G Core through the end-to-end Network Slicer....70
Figure 5-7 - Integration diagram between end-to-end network slice orchestrator and flexible PHY/MAC....71
Figure 5-8 - Flexible PHY/MAC integration with UERANSIM....71
Figure 5-9 - High-level architecture of O-RAN NSSMF....72
Figure 5-10 - Sequence diagram of network slice subnet instance....73
Figure 5-11 - Tree structure of source code of O-RAN NSSMF....74

Co-funded by the Horizon 2020 Framework Programme of the European Union

List of tables

Table 4-1 - Architectural principles of end-to-end network slice orchestration framework 35
Table 4-2 - Data model of inputs to the AI Engine 50
Table 4-3 - REST APIs for tenant and SLA management 57
Table 4-4 - REST APIs for VSB and VSD management 57
Table 4-5 - REST APIs for Vertical Service LCM 58
Table 4-6 - REST APIs for NST/GST management 60
Table 4-7 - REST APIs for end-to-end network slice management 60
Table 4-8 - REST APIs for end-to-end network slice subnet management 61
Table 5-1 - Technical description of VSMF components 63
Table 5-2 - Technical description of NSMF Components 64
Table 5-3 - Technical description of (O-RAN) NSSMF components 66
Table 6-1 - Network slice orchestration functionalities mapped to UCs 75

Co-funded by the Horizon 2020 Framework Programme of the European Union

Abbreviations

5GC	Fifth Generation Core
5QI	5G Quality of Service (QoS) Indicators
AD	Anomaly Detection
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AN	Access Network
API	Application Programming Interface
BS	Base Station
CN	Core Network
CNW	Cumucore Network Wizard
CRUD	Create Read Update and Delete
CS	Communication Service
CSMF	Communication Service Management Function
DSP	Digital Service Provider
EM	Element Manager
eMBB	Enhanced Mobile Broad Band
EPC	Evolved Packet Core
FPGA	Field Programmable Gate Array
GFDM	Generalized Frequency Division Multiplexing
gNB	gNodeB
GSMA	GSM Association
GST	Generalized Network Slice Template
GUI	Graphical User interface
GW	GateWay
IGW	Intelligent GW
LCM	Lifecycle Management
M2M	Machine-To-Machine
MANO	Management and Network Orchestration
mIoT	Massive Internet of Things
ML	Machine Learning
NF	Network Function
NFVO	NFV Orchestrator
NG-RAN	Next Generation RAN
NM	Network Manager
NR	New Radio
NRM	Network Resource Model
NEST	Network Slice Type
NSD	Network Service Descriptor
NSMF	Network Slice Management Function
NSMS	Network Slice Management Service

Co-funded by the Horizon 2020 Framework Programme of the European Union

NSSMF	Network Slice Subnet Management Function
NSSMS	Network Slice Subnet Management Service
NSI	Network Slice Instance
NSS	Network Slice Subnet
NSSI	Network Slice Subnet Instance
NSST	NSS Template
NST	Network Service Template
NWDAF	Network Data Analytics Function
OSS/BSS	Operations Support System / Business Support System
PCF	Policy Control Function
PDU	Protocol Data Unit
PNF	Physical Network Function
QoS	Quality of Service
RAN	Radio Access Network
RIC	Radio Intelligent Controller
RT	Real Time
SBA	Service Based Architecture
SDN	Software Defined Network
SDR	Software-Defined Radio
SLA	Service Level Agreement
SST	Slice/Service Type
TAC	Type Allocation Code
TAI	Tracking Area Identity
TAPI	Transport API
TN	Transport Network
TS	Traffic Steering
UC	Use case
UE	User Equipment
uRLLC	Ultra-Reliable and Low Latency Communication
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VNFM	VNF Manager
VS	Vertical Service
VSB	Vertical Service Blueprint
VSD	Vertical Service Descriptor
VSI	Vertical Service Instance
VSMF	Vertical Service Management Function
WAN	Wide Area Network
WIM	WAN Infrastructure Manager
WG	Working Group

Co-funded by the Horizon 2020 Framework Programme of the European Union

1 Introduction

1.1 Objective of this deliverable

The main objective of this document is to present the design and preliminary prototype implementation of the iNGENIOUS service and network slice orchestration functionalities. Indeed, with 5G, comprehensive Management and Orchestration (MANO) solutions are increasingly required to ease the deployment of heterogeneous vertical services and network slices across several technology domains, including radio, core and transport networks. In particular, the concept of network slicing allows to coordinate resources (network, computing, storage) and network functions (NFs) (virtualized or physical), which are managed and provisioned together to instantiate and compose network services over a shared infrastructure. As a fundamental concept, network slices can be tailored to the requirements of the vertical services that will run on top of them, mostly in terms of Quality of Service (QoS) needs. This is extremely relevant to iNGENIOUS, where heterogeneous IoT network technologies and devices are required to interoperate with the 5G network to provide smart and innovative supply chain and industrial IoT services. In the specific case of these iNGENIOUS services and use cases, a seamless integration and coordination with edge computing infrastructure is of high relevance to provide when needed ultra-low latency capabilities and thus reduce the response time of the deployed applications. Therefore, a key aspect to consider and to highlight is the need of delivering a truly end-to-end orchestration of resources that considers the edge of the network as key for the deployment of network functions and applications. In addition, iNGENIOUS aims at augmenting these orchestration and coordination functionalities with closed-loop capabilities, aiming at a high degree of automation in service and network slice operation leveraging on AI and ML techniques.

Due to the aforementioned aspects, iNGENIOUS has put efforts to define and implement an end-to-end network slice orchestration framework. The work reported in this deliverable presents a comprehensive service and network slice orchestration solution suitable for the iNGENIOUS 5G-IoT network infrastructure and at the same time aligned with the current solutions and trends for network slice management and orchestration as defined by main relevant standardisation bodies (i.e., 3GPP and ETSI). Specifically, the main target of the deliverable is to provide a detailed functional design of the orchestration components, including their integration with closed-loop functionalities enabled by Artificial Intelligence (AI) / Machine Learning (ML). Moreover, as a preliminary development milestone, it also aims at describing the initial software prototype of the proposed end-to-end network slice orchestration framework.

Co-funded by the Horizon 2020 Framework Programme of the European Union

1.2 Positioning in the iNGENIOUS architecture

This deliverable covers a specific functionality within the iNGENIOUS Network Layer, and relates to the management and orchestration features required for the coordination of the relevant network technologies developed in the project. It is worth to mention that, while in other iNGENIOUS deliverables the terms MANO and cross-layer MANO are used, in this document the term-end-to-end network slice orchestration framework is used.

As anticipated above, the traditional and pure MANO functionalities are augmented with AI/ML enabled closed-loop features to achieve a full automation in services and network slice management, control and operation. Figure 1-1 highlights the end-to-end network slice orchestration framework within the iNGENIOUS architecture as presented in D2.2 [1]. As shown in the picture, the end-to-end network slice orchestration framework encompasses the MANO functional block, including part of the orthogonal AI/ML functionalities within the iNGENIOUS architecture. In practice, the end-to-end network slice framework described in this deliverable provides a set of overarching management and orchestration features to glue together different network and computing technology domains, spanning from the radio, edge, transport up to the core network.

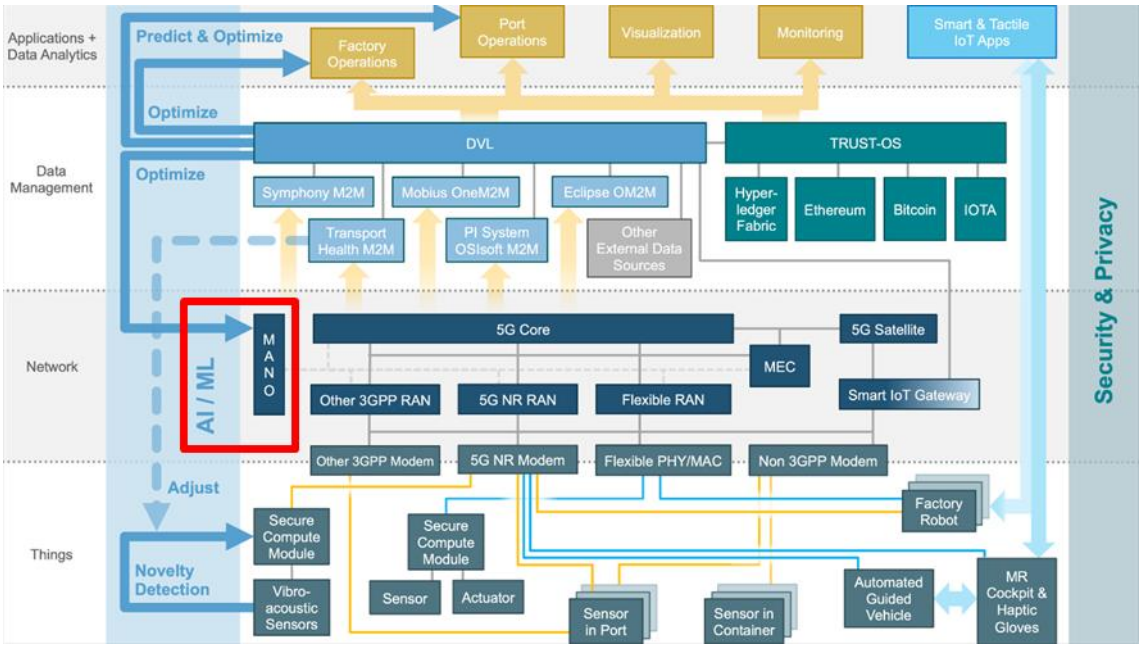

Figure 1-1 - Overview of the iNGENIOUS architecture (D2.2)

This end-to-end overarching approach is more clearly depicted in Figure 1-2, where it shown the interaction for resource management, orchestration, provisioning and operation in the different technology domains and layers. As deeply detailed in the next sections of this document, the main interactions supported by the iNGENIOUS end-to-end network slice orchestration framework (in terms of actual implementation, integration and validation) refer to the Radio Access Network (RAN), the 5G Core and the edge (shown as MEC in the figure), with this latest as specific case of compute location for

Co-funded by the Horizon 2020 Framework Programme of the European Union

deploying virtualized network functions and applications that impose ultra-low latency requirements.

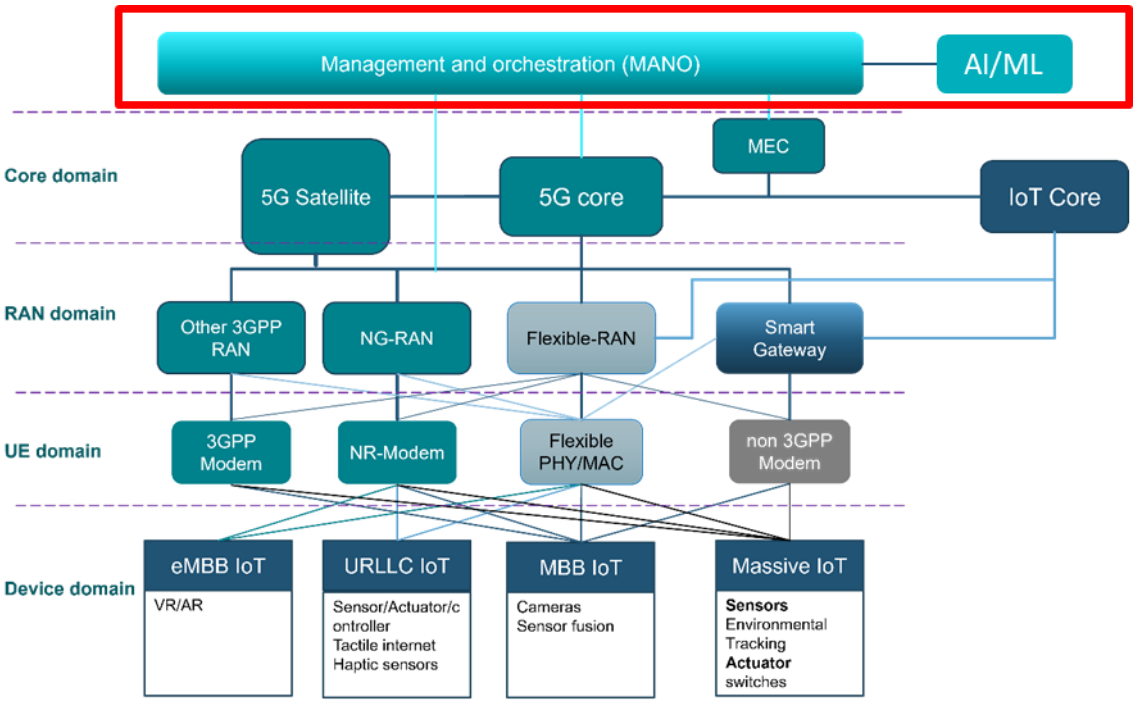

Figure 1-2 - Highlight on the iNGENIOUS network layer components

1.3 Structure of the document

This deliverable is structured as follows:

- Section 2 provides a brief overview of relevant state-of-the-art and related work for the design and implementation of the iNGENIOUS end-to-end network slice orchestration. This mostly includes 5G network slicing, information models and management approaches defined by 3GPP.
- Section 3 provides an overview of the iNGENIOUS 5G-IoT network, focusing on the solutions and technologies integrated with the end-to-end network slice orchestration framework. It briefly summarizes the content described in previous WP4 deliverables helping to better contextualize the work reported in this document.
- Section 4 is the core part of the document, and describes the end-to-end network slice orchestration framework architecture, highlighting the functional components, data models, workflows and interfaces.
- Section 5 describes the preliminary software prototype of the iNGENIOUS end-to-end network slice orchestration framework, including a report of the initial integration activities carried out within WP4.
- Section 6 identifies the main innovations that the iNGENIOUS end-to-end network slice orchestration framework brings and how these innovations are mapped against the project Use Cases.
- Section 7 provides some concluding remarks and highlights the next steps foreseen for the work reported in the document.

Co-funded by the Horizon 2020 Framework Programme of the European Union

2 Related Work

This chapter provides an overview of the major outcomes from 3GPP and ETSI standardization activities in the areas of 5G network management and orchestration, including network slicing in 5G infrastructures. The main references in this context are the 3GPP technical specifications from the TSG-SA Working Groups (WG) for the 5G system Architecture (TSG-SA WG 2) and the Management, Orchestration and Charging (TSG-SA WG 5), with particular reference to Release 16 and the latest Release 17. This 3GPP work is complemented with the ETSI NFV specifications for management and orchestration of Virtual Network Functions (VNFs) and network services, which are considered the reference baseline for 5G infrastructures exploiting the concept of network virtualization.

2.1 Management and orchestration of 5G networks

The 3GPP TSG-SA WG 5, responsible for the aspects related to Management, Orchestration and Charging of 5G networks, has defined a generalized mobile network management architecture in the 3GPP TS 28.500 specification [2]. The architecture, depicted in Figure 2-1, involves a 3GPP Management System with a Network Manager (NM) and Element Managers (EM) that control the elements composing a 5G network, where each of them can be deployed as a Physical Network Function (PNF) or a VNF. The presence of VNFs in the 5G mobile network introduces the need of a Management and Orchestration (MANO) framework responsible for their provisioning, configuration and, more in general, for their lifecycle management (LCM), in cooperation with the 3GPP management system.

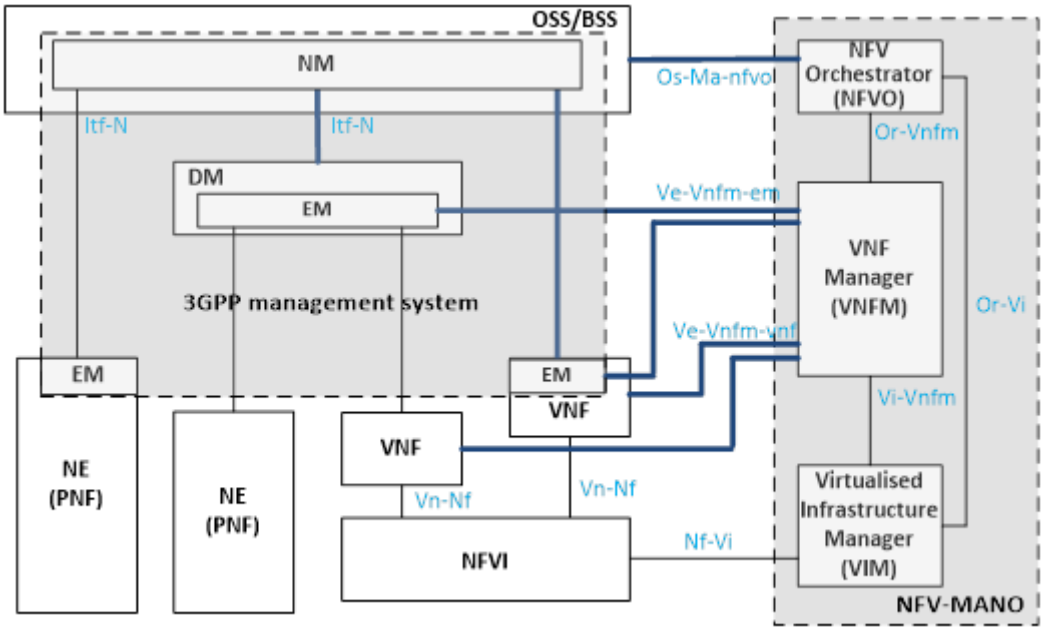

Figure 2-1 - Mobile network management architecture – Interaction between 3GPP management system and NFV-MANO framework [2]

Co-funded by the Horizon 2020
Framework Programme of the European Union

The MANO framework is based on the architecture defined by the ETSI NFV ISG for the NFV-MANO [3] and includes the three elements of the NFV Orchestrator (NFVO), VNF Manager (VNFM) and Virtual Infrastructure Manager (VIM). In this scenario, the NM of the 3GPP management system is part of the Operations Support System / Business Support System (OSS/BSS) and interacts with the NFVO to request the provisioning and drive the management of the NFV Network Services composed by the VNFs that build the mobile communication network.

The adoption of virtualized functions as elements of the mobile network brings higher degrees of dynamicity and flexibility in the 5G network deployment. It also enables a number of features in its management and operation, including dynamic instantiation, automated scaling, optimization and healing. Such functionalities can be driven by an external management logic and actuated through the NFV Orchestrator, with the cooperation of VNFM and VIM for the configuration of virtual functions and the control of the virtual resource allocation, respectively.

The interaction between the 3GPP management system and the ETSI NFV MANO components is further specified by the 3GPP TS 28.528 specification [4], which identifies the ETSI NFV SOL 005 interface [5] as the reference for the communication between the NM and NFVO. This interface is used to request provisioning and lifecycle management actions (e.g., termination, scaling, healing, re-configuration) on the NFV network services related to a 5G mobile network including virtualized elements. Additional FCAPS-related functionalities are also supported on this interface, including monitoring, policy management, failures, and alarms notifications, etc.

2.2 5G network slices

The network slicing concept has been introduced in 5G networks to allow the operators to effectively share their own infrastructure creating multiple concurrent logical partitions, i.e., the network slices. Network slices can be easily customized according to the business requirements of their customers or the technical requirements of their services. Network slices can be differentiated and updated independently, offering various degrees of isolation, and they can be adapted in terms of mobile connectivity, virtual functions or computing and storage resources.

Network slices can be easily configured to offer dedicated communication services to the verticals, e.g., customized on the basis of their production requirements. For example, ultra-Reliable and Low Latency Communications (uRLLCs) meet the requirements of the production lines automated control in Industry 4.0 and smart factory scenarios. Massive Internet of Things (mIoT) communications are particularly suitable to manage huge amounts and high density of IoT sensors and actuators. Enhanced Mobile BroadBand (eMBB) communications are the enablers for video producing and broadcasting, offering high data rates in both uplink and downlink directions. Vehicle to Everything (V2X) communications support high-bandwidth, low-latency and high-reliable interactions among moving (autonomous) vehicles and different entities such as other vehicles, pedestrian, etc.

Co-funded by the Horizon 2020 Framework Programme of the European Union

The concept of network slicing is introduced in the 3GPP TS 23.501 [6] specification, where a network slice is defined like a logical network that provides specific network capabilities and characteristics. A Network Slice Instance (NSI) consists of a set of Network Functions (NFs), with their own computing, storage and networking resources. A NF can be implemented within a NSI as a PNF running on dedicated hardware or as a VNF instantiated over a computing shared infrastructure, e.g., on the cloud. In a 5G network, an end-to-end NSI includes the NFs related to control and user planes of the 5G Core Network, as well as Next Generation RAN (NG-RAN) functions for the 3GPP mobile access network.

Network slices can be differentiated in terms of network functions and network capabilities, according to a number of major categories defined through a Slice/Service Type (SST) including eMBB, uRLLC, mIoT and V2X. A network operator can thus instantiate multiple NSIs with their specific SST to differentiate the business offer towards its own customers. Moreover, multiple NSIs with the same SST can be instantiated and reserved to different customers to better guarantee their traffic QoS, isolation and security.

The 3GPP TS 28.530 specification [7] defines the major concepts related to the management of a network slice to support specific types of Communication Services (CS), or vertical services. The network slice is presented as a logic network, including the related pool of resources, which enables the delivery of a CS on the basis of its characteristics and requirements (e.g., maximum latency and jitter, minimum data rates, density of UEs, coverage area, etc.). As shown in Figure 2-2, different types of CS can be supported through dedicated NSIs.

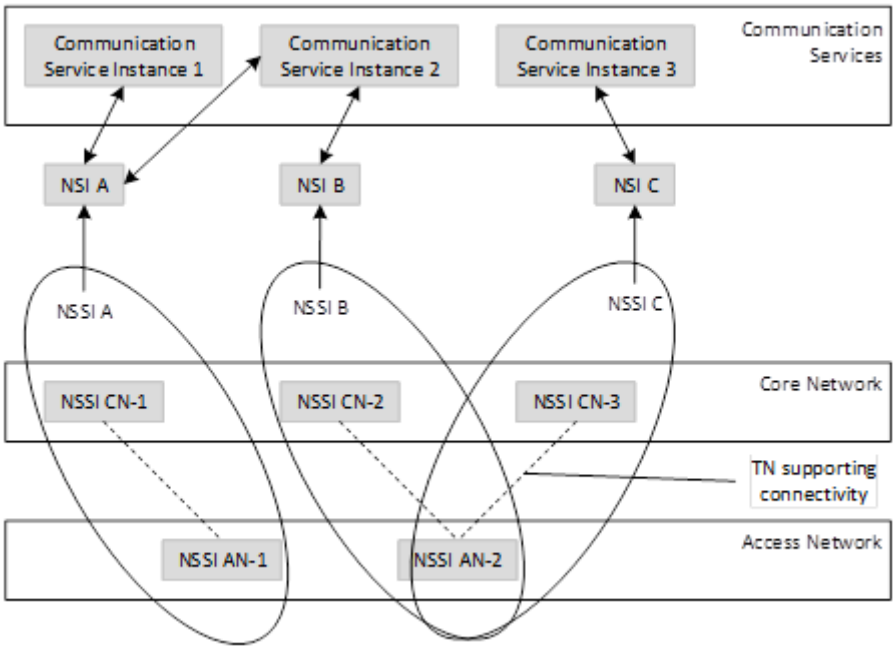

Figure 2-2 - Multiple instances of communication services deployed over different network slice instances [7]

As highlighted in the picture, a NSI can support one or more instances of CS. Moreover, a NSI is formally modelled as an end-to-end Network Slice Subnet Instance (NSSI), which in turn can include multiple NSSIs (see the network slice information model in Section 2.2.1 for further details). In particular, the

Co-funded by the Horizon 2020 Framework Programme of the European Union

figure shows a common pattern of network slice modelling, with the end-to-end NSSI composed of two lower level NSSIs: the former related to the 5G Core Network (CN) and the latter related to the Access Network (AN). Each of them includes the related NFs, which communicate through the underlying connectivity provided by the Transport Network (TN). In other terms, a NSSI represents a group of NF instances (together with their own resources) that implement a part of the NSI. Through this concept, it is possible to manage the set of NF and related resources as an atomic element, independently on the rest of the NSI.

A key concept of the network slices is the possibility to share one or more NSSIs among multiple NSIs. This approach allows to share NFs (physical or virtual) and their resources among different end-to-end NSIs and, consequently, multiple service instances. Therefore, the operator can adopt strategies for resource allocation and sharing, in compliance with the isolation requirements of the target services, in order to optimize the utilization of its own physical infrastructure.

Clearly, there is a strong relationship between the lifecycle of a CS instance and the lifecycle of the NSI hosting the service. Whenever a new CS instance is created, a new NSI may need to be established or an existing one may be activated, re-configured, or up-scaled according to the requirement of the additional service. Similarly, when the CS instance is terminated, the related NSI may be deactivated, down-scaled, re-configured or entirely terminated, depending on the presence of additional services still running. iNGENIOUS implements a similar approach, creating on demand NSIs which will be shared among multiple services on a per-tenant basis and modifying their configuration and deployment according to the service requests and the real-time network conditions.

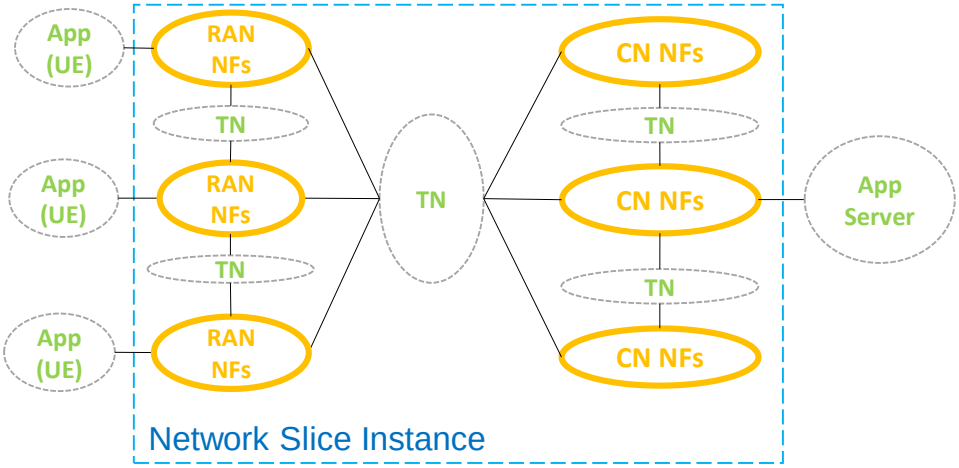

Figure 2-3 - Example of network slice instances [7]

Figure 2-3 provides an example of NSI. NFs related to the RAN and the CN are interconnected through a TN. iNGENIOUS assumes the same modelling for the network slices. It is worth to mention that the management of the NFs related to AN and CN is covered by the management system defined in the 3GPP specifications. On the other hand, the TN management is usually delegated to a "non-3GPP" system that interacts with the 3GPP management system to receive the requirements in terms of transport network topology, QoS and network connectivity (see 3.3).

Co-funded by the Horizon 2020 Framework Programme of the European Union

2.2.1 INFORMATION MODELS FOR 5G NETWORK SLICES

The information model of a NSI is defined in the 3GPP TS 28.541 [8], as part of the 5G Network Slice Resource Model (NRM). The model, represented in Figure 2-4, highlights how an end-to-end network slice, composed of several network slice subnets, can be deployed through a number of NFV network services and (virtual) network functions.

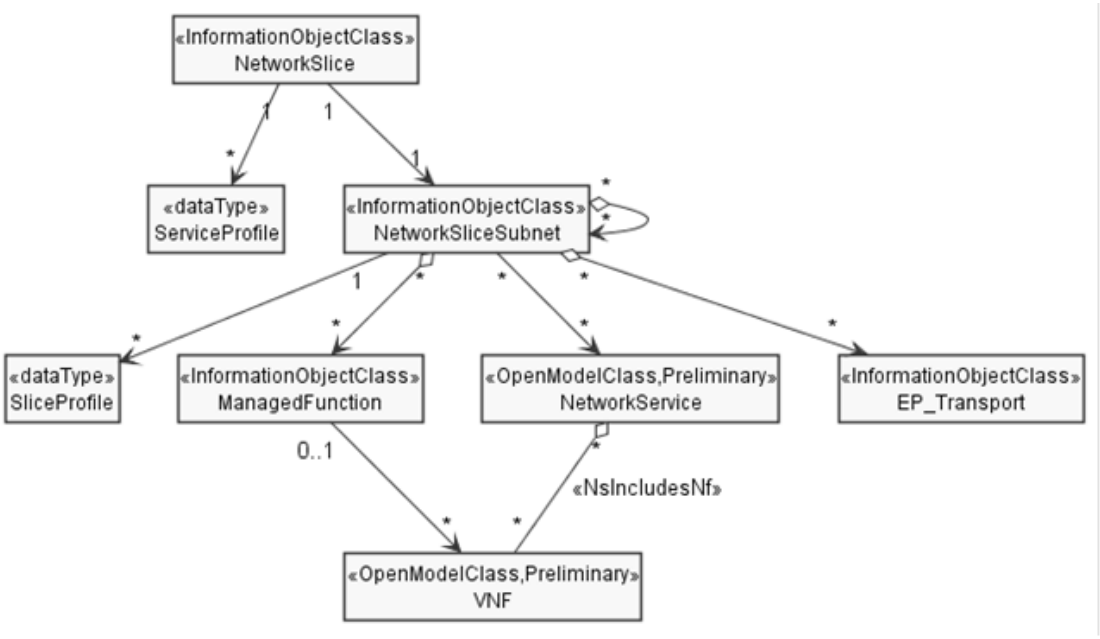

Figure 2-4 - Structure of network slices and network slice subnets in network services and virtual network functions [8]

A Network Slice is associated to an end-to-end Network Slice Subnet that defines the slice's internal elements and their interconnectivity, together with a set of Service Profiles describing the service requirements. Example of Service Profile parameters includes maximum number of UEs, service coverage area, maximum latency, per-slice and per-UE throughput in uplink and downlink, maximum number of allowed connections, jitter, UEs' maximum speed, etc.
The information model of a Network Slice Subnet includes the following main elements:

- Slice Profiles defining the requirements of the slice subnet (similar to the Service Profile at the network slice level);
- Managed Functions, i.e., the network functions managed within the network slice subnet which can corresponds to one or more VNFs;
- Network Service, representing a group of interconnected VNFs and/or PNFs composing (part of) the slice subnet. The Network Service concept corresponds to the NFV Network Service defined in the context of the ETSI NFV ISG and, in particular, in the ETSI GS NFV-IFA 014 [9] and ETSI GS NFV-IFA 013 [10] that define the models for the Network Service Descriptor (NSD) and for the Network Service instance, respectively.

As shown in Figure 2-5, an NSD represents the topology of a network service, identifying its internal network functions (through references to the VNF

Co-funded by the Horizon 2020 Framework Programme of the European Union

and/or PNF descriptors) and describing how they are interconnected through the Virtual Links. Moreover, the NSD also defines the logic of the communications among the network functions, describing how the traffic should be forwarded through the sequence of functions. This aspect is defined through the "VNF Forwarding Graph", which indicates the sequence of VNFs, and the related "Network Forwarding Path", which describe the traffic flows and their L2/L3 classifiers.

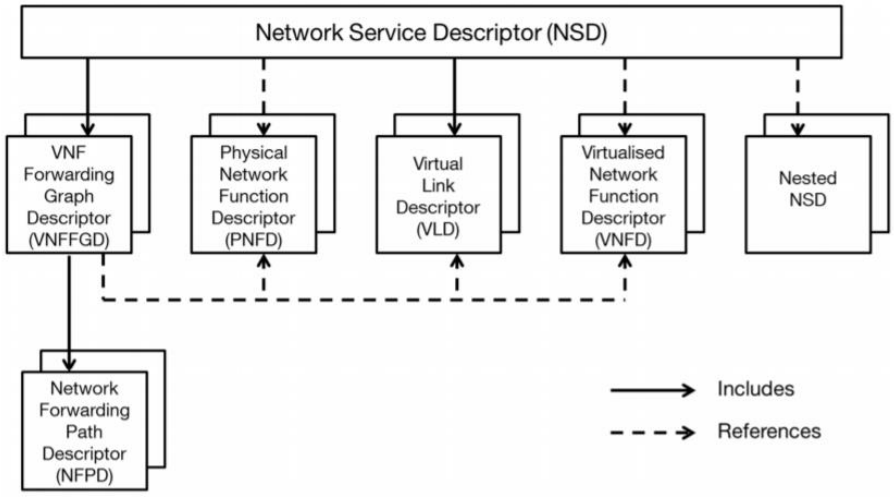

Figure 2-5 - Information model of a Network Service Descriptor [9]

The 3GPP information model reported in Figure 2-4 captures the internal technical details of a network slice instance, identifying its components and their connectivity. However, when exposing the generic characteristics of a network slice towards external entities (for example in case of network slice offers to potential customers), it is useful to refer to a "Network Slice Template" that describes the slice capabilities through a more abstract model that hides its internal details and the operator implementation choices. In this case, the slice can be defined through the "Generalized Network Slice Template" (GST) [11] defined by the GSM Association (GSMA).

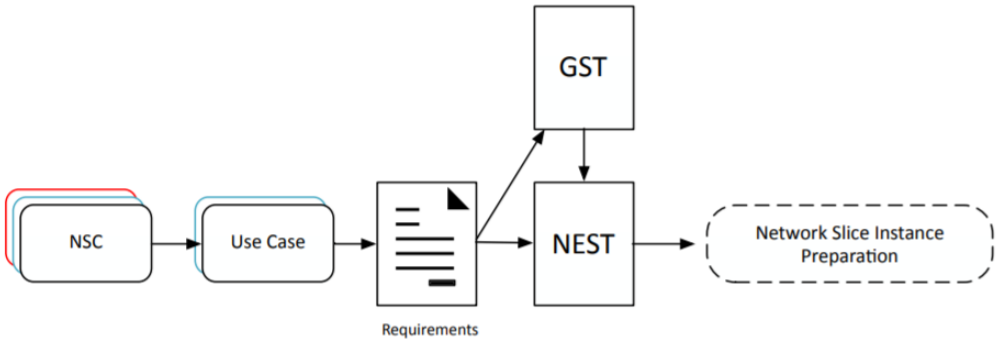

Figure 2-6 - Role of GST and NEST as input for the preparation of a NSI [11]

The GST consists of a set of attributes that characterize a given type of network slice and it is *generic*, i.e., independent on the specific implementation of the slice. When the GST attributes are associated to specific values, the result is a Network Slice Type (NEST), which constitutes the main input for the preparation of a Network Slice Instance, as shown in Figure 2-6.

Co-funded by the Horizon 2020 Framework Programme of the European Union

2.2.2 MANAGEMENT OF 5G NETWORK SLICES

The 3GPP TR 28.801 specification [12] defines the high-level functional architecture for the management of network slices in support of communication services, identifying the three functional elements of the Communication Service Management Function (CSMF), Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF), represented in Figure 2-7.

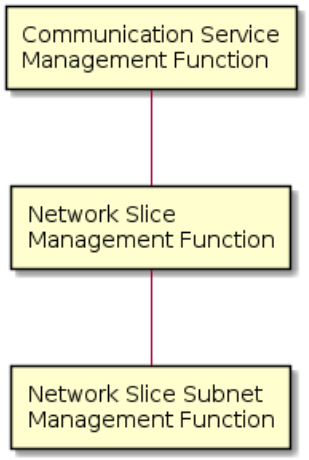

Figure 2-7 - High-level architecture of management functions for network slicing [12]

At the upper layer, the CSMF is responsible of processing the requests for new CS and manages the CS instances provided by a network operator. The CSMF translates the CS requirements into a set of network slice characteristics, e.g., defining the SST, the required capacity of the mobile connectivity, the QoS requirements, etc., and interacts with the NSMF to request the creation of the related NSI.

The NSMF is responsible for the management and end-to-end orchestration of NSIs, on the basis of the requests received from the CSMF. The NSMF splits the NSI into its internal NSSIs, according to the NEST, and manages their lifecycle. Therefore, the NSMF is the entity which takes decisions about the composition of a NSI, including the re-usage of pre-existing NSSIs that can be shared among multiple NSIs, and the coordination of their provisioning, scaling and/or configuration. The actuation of these decisions is then delated to the NSSMFs, which are the final responsible for the management and orchestration of each NSSI.

As analysed in the 3GPP TS 28.533 specification [13], which defines an architecture of the 3GPP management system designed following the Service Based Architecture (SBA) pattern, a typical deployment of the 3GPP management system is structured with domain-specific NSSMFs, related to the RAN, the CN or TN domains. Such NSSMF are customized according to the specific requirements and technologies adopted in their own target domain. As detailed in Section 4, the iNGENIOUS end-to-end network slice orchestration architecture follows a similar approach introducing dedicated NSSMF to handle the RAN, 5G core and transport domains, as shown in Figure 2-8.

Co-funded by the Horizon 2020 Framework Programme of the European Union

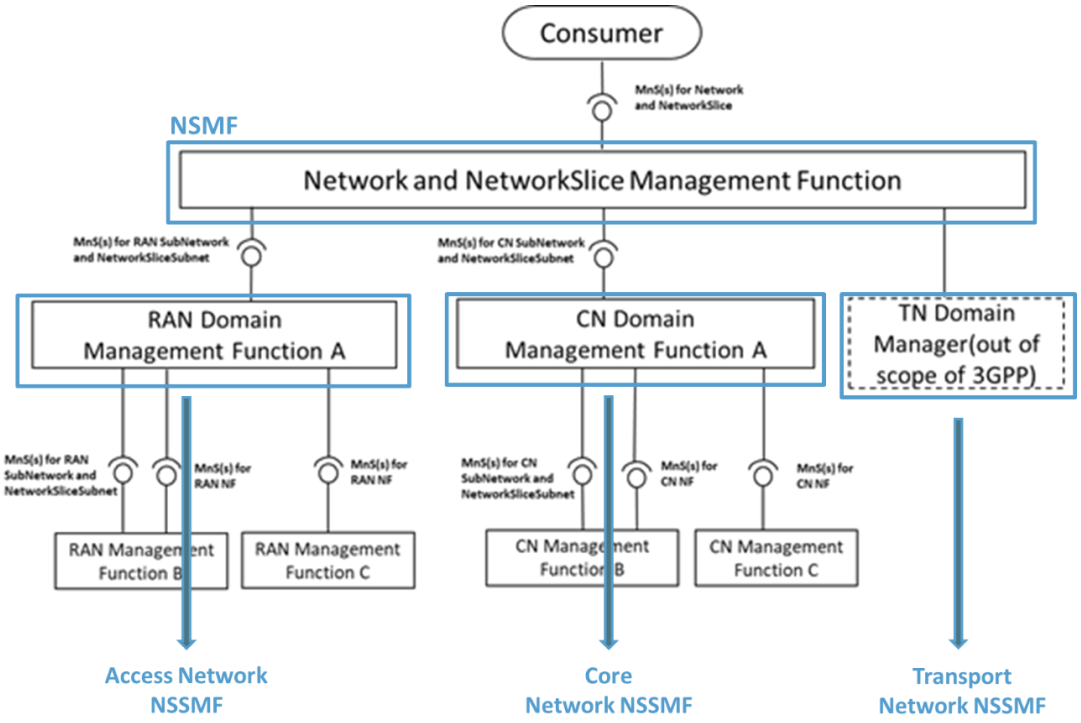

Figure 2-8 - Hierarchical interaction between NSMF and per-domain NSSMFs [13]

3GPP standards do not mandate any specific implementation of the NSMF and NSSMF components. However, the 3GPP TR 28.533 specification [13] proposes a deployment option, widely used in production infrastructures, where the management of the network slices and slice subnets lifecycle is handled through an interaction with the NFV MANO system, where the NFV Orchestrator is responsible for the lifecycle of the NFV Network Services associated to the NSSIs (see Figure 2-9).

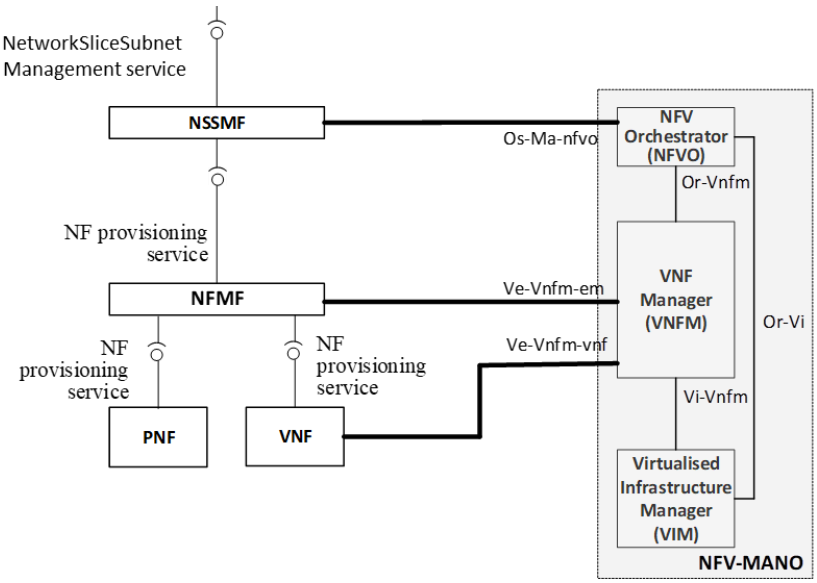

Figure 2-9 - Example of management system deployment with NSSMF interacting with NFVO [13]

iNGENIOUS is aligned to this approach and relies on the NFV MANO for the instantiation and lifecycle management of the virtual functions related to the 5G Core Network and to the application services within the end-to-end network slices.

Co-funded by the Horizon 2020 Framework Programme of the European Union

3 The 5G-IoT Network

This chapter is intended to provide a brief overview of the iNGENIOUS 5G-IoT network technologies that are targeted to be managed, at least at the design and concept level, by the end-to-end network slice orchestration framework described in this document. The aim is to provide a summary description of such technologies for the sake of completeness of this deliverable.

3.1 Radio Access Network

This section provides an introduction on two specific aspects of the work carried out as part of the iNGENIOUS Network Layer and that are going to be integrated with the iNGENIOUS MANO functionalities, namely, the flexible PHY/MAC implementation and O-RAN. Both items have been described in detail in the Deliverable D4.2 [14].

3.1.1 FLEXIBLE PHY/MAC

The flexible PHY/MAC implementation consists of a centralized multiple access scheme based on a polling mechanism, which allows the base station (BS) to have a full control of the traffic such that the PHY parameters can be adjusted according to the application needs. The goal of this implementation is to provide a dynamic RAN deployment with Software Defined Radio (SDR) devices, such that applications with diverging needs are able to share the wireless medium with efficient use of the radio resources in the time domain.

In order to implement this MAC protocol, a control channel has been defined to transmit control data and coordinate the medium usage. At the PHY level, the implementation is based on the generalized frequency division multiplexing (GFDM) waveform, which is advantageous from the flexibility perspective, i.e., GFDM can be flexibly configured depending on the data type and user's needs. A basic frame structure is given in the Figure 3-1. It can be seen that the payload length is variable, meaning that users can have different PHY configuration.

Figure 3-1 - Flexible PHY/MAC frame structure.

Moreover, this implementation uses SDR platforms, with field programmable gate array (FPGA) based implementation for real-time signal processing. As described in the Deliverable D4.1 [15], the flexible PHY will be applied in the iNGENIOUS Factory UC, where different applications will be deployed in a dynamic environment. For example, applications that require relatively high throughput, e.g., video stream will have more resources allocated than applications requiring less data, e.g., remote control of AGV.

The relevant aspects of the integration of flexible PHY/MAC and iNGENIOUS MANO layer is related to the management and coordination of radio resource allocation according to application requirements, with the aim of integrating

Co-funded by the Horizon 2020 Framework Programme of the European Union

the solution within (private) 5G networks. The status of the current activities and planned development in this regard is described in Section 5.3.

3.1.2 O-RAN

The O-RAN Alliance is working together with hundreds of vendors and operators to transform the radio access network industry towards open, intelligent virtualized and fully interoperable RAN. One of the most relevant elements in the architecture is the RAN Intelligent Controller (RIC), which is designed to take intelligent decisions and provides the possibility manage the radio resources, is divided into the non-real time (Non-RT) and the near-real-time (Near-RT) components.

To improve RAN efficiency and performance, dedicated applications are used, the called rApps in the Non-RT RIC and xApp in the Near-RT RIC, that work independently of the RIC. The main differences between rApps and xApps are that rApps are used to help to create policies, and xApps directly control a real-time function within the RAN element, and act at different times. In iNGENIOUS, the Near-RT RIC has been deployed to connect with the end-to-end network slice orchestration framework through the O-RAN A1 interface, and validate xApps defined in the O-RAN community.

One of the main application use cases is Anomaly Detection, with the objective of detecting and correcting anomalies in the UEs, using quality metrics and throughput values. For this, RIC makes use of three xApps: Anomaly Detection (AD), Traffic Steering (TS) and QoS Prediction (QP). When the AD xApp detects an anomalous UE (due to degradations of RSRP, RSRQ, SINR and/or throughput), sends the UE ID to the TS xApp, which asks for a throughput prediction to the QP xApp. QP xApp predicts the throughput based on signal quality parameters and cell load data, and sends the data to TS xApp, who checks if a neighbour cell can provide better throughput to the UE sends a handover request. With this application we can take decision in base of, i.e., the UE priority, and offer the best possible throughput.

The interconnection between the end-to-end network slice orchestration framework and the Near-RT RIC of O-RAN allows to set policies following the specifications (in terms of APIs, data models) of the A1 interface. These policies define to which context the policy statement will be applied and the goals for the Near-RT RIC. The full workflow of the policy statement is explained in the Deliverable D4.2 [14]. O-RAN has enabled a new release (E), that is currently being deployed and will test once the A1 standard policy is completely available to use. This integration will allow to automatize the exchange of policies between the Near-RT RIC and the end-to-end network slice orchestration framework.

3.2 5G Core

The 5G Core architecture compared to previous generations is designed to support UE and IoT communications. For this reason, Network Slicing creates virtual instances of the core to separate traffic with different high-level requirements. For instance, IoT data flow can be isolated from consumer data flow, each one of them belonging to different 5G slices. Moreover, different

Co-funded by the Horizon 2020 Framework Programme of the European Union

flavours of IoT traffic can have their own slice depending on their requirements in terms of latency, reliability, or bandwidth capacity.

Network Slicing feature enables to deliver several virtual networks relying on the same physical network as depicted in Figure 3-2. Virtual networks can have different operational characteristics among themselves. For instance, these characteristics can be maximum number of simultaneous users, maximum capacity, access to advanced features like 5GLAN and integration to external Data Networks, etc.

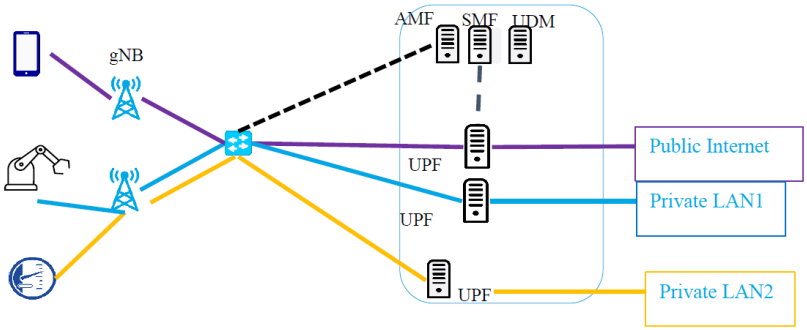

Figure 3-2 - Network slicing for different machine type communications

As depicted in Figure 3-3, Cumucore Network Wizard (CNW) consists of a user-friendly Graphical User interface (GUI) that interacts with the Cumucore Network Controller (CNC) through a RESTful interface to create and manage Network Slices and use of network slices on data flow level.

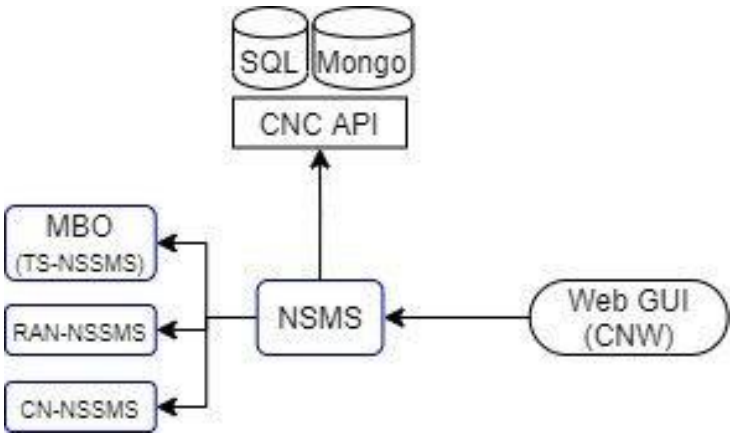

Figure 3-3 - Network slicing management modules

There are two separate APIs to manage network slices and dataflows:

- There is an API provided by the CNC that can be used from CNW to create, manage and terminate Network Slices.
- An API provided by the Policy Control Function (PCF) that is used to generate dataflows with specific QoS settings and related SLAs.

The Network Slice Management is done through an Application Function (AF) named informally NSM-AF. Network Slice Management Service (NSMS). or Network Slice Subnet Management Service (NSSMS) are defined in 3GPP

Co-funded by the Horizon 2020 Framework Programme of the European Union

specifications to support different use cases defined in TS 28.531 [16]. The CNC has been implemented as the 3GPP standard NSMS and CNW as the NSM-AF. A slice subnet is considered a different segment of the end-to-end system e.g., RAN subnet, Transport subnet and Core subnet.

3.2.1 NETWORK SLICE INSTANCE CREATION

To satisfy requests for allocation of a network slice instance with certain characteristics, the request shall include the network slice related requirements.

As depicted in Figure 3-4, the network slice NSMS receives the request for allocation of the network slice instance with certain characteristics. The request contains network slice related requirements and the information indicating whether the requested NSI could be shared with other consumers.

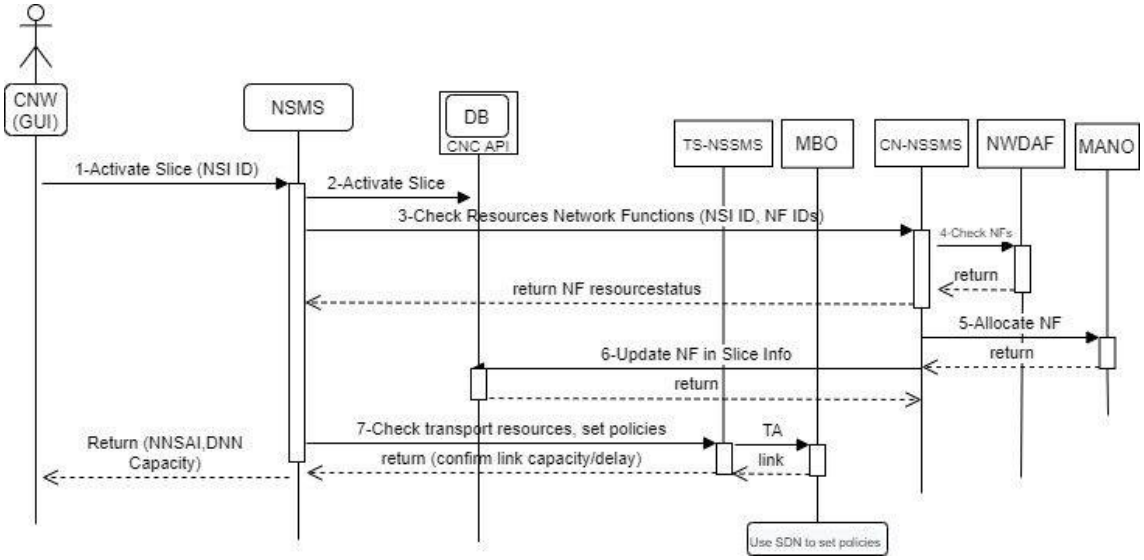

Figure 3-4 - Network slice activation flow

1. The network slice provisioning management service provider triggers the creation of a new Network Slice Instance (NSI).
2. For the required NSSI(s), the network slice provisioning management service provider sends network slice subnet related requirements to the network slice subnet provisioning management service provider to request allocation of the required NSSI(s).
3. The network slice provisioning management service provider receives the information of the allocated NSSI(s).
4. The network slice provisioning management service provider, via the network slice subnet provisioning management service provider, sends the transport network related requirements (e.g., external connection point, latency and bandwidth) to the Transport Network (TN) manager. The TN manager reconfigures the TN accordingly and responds to the network slice provisioning management service provider via the network slice subnet provisioning management service provider.

Co-funded by the Horizon 2020 Framework Programme of the European Union

5. The network slice provisioning management service provider receives the response from TN manager via the network slice subnet provisioning management service provider.
6. The network slice provisioning management service provider associates the NSSI(s) with the corresponding NSI (e.g., allocation of the management identifier of NSI and mapping the management identifier of NSI with the received management Identifier of NSSI(s)) and triggers to establish the links between the service access points of the NSSI(s).

3.2.2 NETWORK SLICE SUBNET INSTANCE CREATION

Create a new network slice subnet instance or use an existing network slice subnet instance to satisfy the network slice subnet related requirements.

The steps are:

1. Based on the network slice subnet related requirements received, the network slice subnet provisioning management service provider decides to create a new NSSI or use an existing NSSI.
2. The network slice subnet provisioning management service provider triggers to create a new NSSI, the following steps are needed.
3. The first network slice subnet provisioning management service provider receives the constituent NSSI information from the other network slice subnet provisioning management service provider (s) and associates the constituent NSSI(s) with the required NSSI.
4. Based on the network slice subnet related requirements received and Slice Profile, the network slice subnet provisioning management service provider decides that to satisfy the NSSI requirements. The network slice subnet provisioning management service provider determines the NS related requirements (i.e. information about the target NSD and additional parameterization for the specific NS to instantiate).
5. Based on the NS related requirements, the network slice subnet provisioning management service provider triggers corresponding NS instantiation requests to Network Function (NF).
6. The network slice subnet provisioning management service provider associates the NS instance with corresponding network slice subnet instance (e.g., allocation of the management identifier of NSSI and mapping with the corresponding identifiers).
7. The network slice subnet provisioning management service provider is using the NF provisioning service to configure the NSSI constituents.

Co-funded by the Horizon 2020 Framework Programme of the European Union

3.2.3 NETWORK SLICE AND NETWORK SLICE SUBNET INSTANCE ACTIVATION

This section provides the steps for network slice and network slice subnet instance activation.

To activates an NSI based on the received network slice related request from a customer after on boarding, the following steps are needed:

1. The NSMS checks whether NSSIs associated with the NSI are all in active state; if there is an inactive NSSI, the network slice provisioning management service provider requests the network slice subnet provisioning management service provider to activate the corresponding NSSI.
2. The NSMS receives a response from the network slice subnet provisioning management service provider indicating that the NSSI is active.
3. The network slice provisioning management service provider sets the state of the NSI as active and sends a response to the requesting consumer.

To activate an existing network slice subnet instance which is in inactive state, the required steps are:

1. The network slice subnet provisioning management service provider identifies inactive constituents (e.g., NSSI, NF) of the NSSI and decides to activate those constituents.
2. If the constituent of NSSI is managed directly by the network slice subnet provisioning management service provider, the network slice subnet provisioning management service provider activates the NSSI constituent directly.
3. If an NSSI constituent is an NF managed by NF related provisioning management service provider, the network slice subnet provisioning management service provider requests the NF related provisioning management service provider to activate the NF.
4. The network slice subnet provisioning management service provider receives responses indicating that NSSI constituents are all activated.
5. The network slice subnet provisioning management service provider sets the state of the network slice subnet instance as active and sends a response to its authorized consumer.

Co-funded by the Horizon 2020 Framework Programme of the European Union

3.2.4 NETWORK DATA ANALYTICS FUNCTION

The 5G Core architecture also includes network functions that collect data from other network functions for monitoring. This information is made available to external orchestrator to take some corrective actions and launch or terminate some instances. The NetWork Data Analytics Function (NWDAF) represents operator managed network analytics logical function.

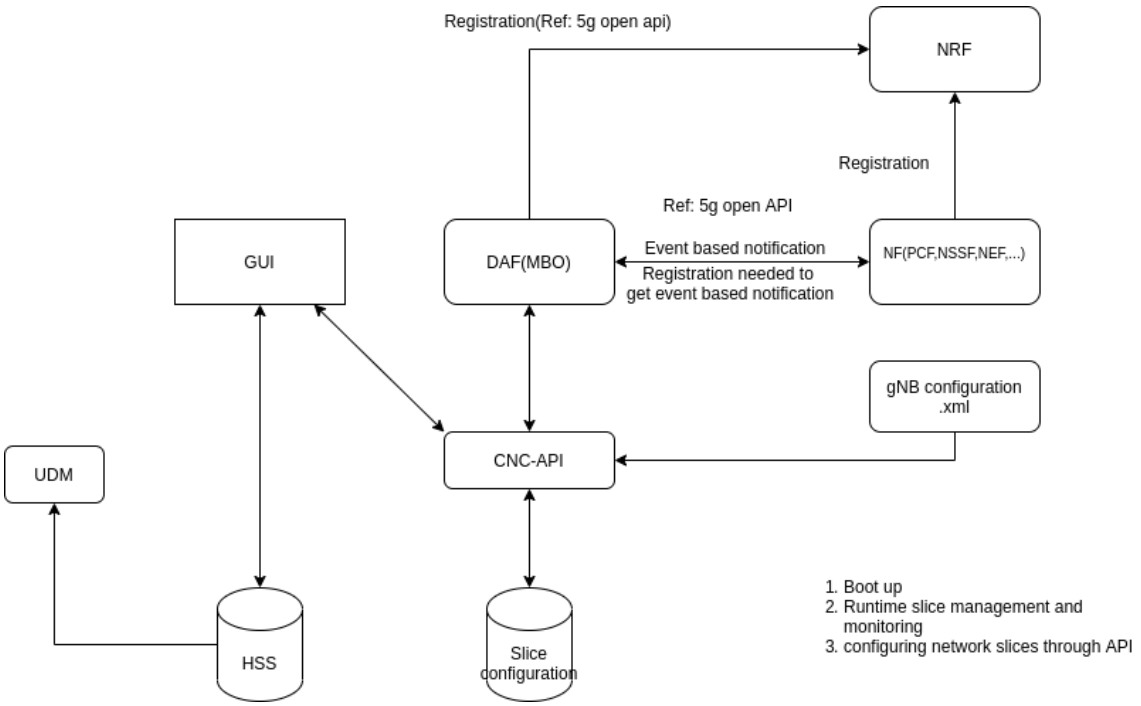

Figure 3-5 - Network Data Analytic Function modules design.

As depicted in Figure 3-5, the NWDAF includes the following functionality:

- Support data collection from NFs and AFs;
- Support data collection from OAM;
- NWDAF service registration and metadata exposure to NF/AFs;
- Support analytics information provisioning to NF or AFs.

The details of the NWDAF functionality are defined in TS 23.288 [17] and it can expose the analytics to both internal or external AFs The NWDAF analytics may be securely exposed by NEF for external party, as specified in TS 23.288. The AFs can subscribe to following events published by the NWDAF:

- Service Experience information, as defined in clause 6.4.2, TS 23.288.
- UE Mobility information, as defined in clause 6.7.2.2, TS 23.288.
- UE Communication information, as defined in clause 6.7.3.2, TS 23.288.
- Exceptional information, as defined in clause 6.7.5.2, TS 23.288.

Co-funded by the Horizon 2020 Framework Programme of the European Union

3.2.4.1 Retrieval of data from external party by NWDAF

Data provided by the external party may be collected by NWDAF via NEF for analytics generation purpose. NEF handles and forwards requests and notifications between NWDAF and AF, as specified in TS 23.288.

NWDAF will collect information from all Network Functions (NFs) and network switches using SDN controller to fetch counter information and will store into CNC DB. NWDAF uses CNC to access the network statistics stored in the DB.

The NWDAF provides three services:

- Event subscription service which allows NFs to subscribe/unsubscribe for different analytic information found in 3GPP TS 29.520 [18]
- Event notification service provides analytic notification for NFs which subscribe for a certain event based on subscription ID
- The NWDAF provide Analytic Info service which allows any NF to request and get specific analytic information.

3.2.4.2 NWDAF discovery and selection

Multiple instances of NWDAF may be deployed in a network. The NF consumers shall utilize the NRF to discover NWDAF instance(s) unless NWDAF information is available by other means, e.g., locally configured on NF consumers. The NWDAF selection function in NF consumers selects an NWDAF instance based on the available NWDAF instances. The following factors may be considered by the NF consumer for NWDAF selection:

- S-NSSAI
- Analytics ID(s).
- NWDAF Serving Area information, i.e., list of Tracking Area Identities (TAIs) for which the NWDAF can provide analytics.

3.2.4.3 NWDAF Data collection procedure

The NWDAF collects analytics from core NF and expose the results using the *Namf_EventExposure* service endpoint defined in TS23.502 [19]. This service enables an NF to subscribe and get notified about an Event ID. Following UE access and mobility information event that can be collected from the AMF:

- Location Report (TAI, Cell ID, N3IWF/TNGF node, UE local IP address and optionally UDP source port number);
- UE moving in or out of a subscribed "Area Of Interest" as described in clauses 5.3.4.4 and 5.6.11 in TS 23.501
- Number of UEs served by the AMF and located in "Area Of Interest";
- Time zone changes (UE Time zone);
- Access Type changes (3GPP access or non-3GPP access);

Co-funded by the Horizon 2020 Framework Programme of the European Union

- Registration state changes (Registered or Deregistered); - Connectivity state changes (IDLE or CONNECTED);
- UE loss of communication;
- UE reachability status;
- UE indication of switching off SMS over NAS service;
- Subscription Correlation ID change (implicit subscription);
- UE Type Allocation code (TAC);
- Frequent mobility re-registration;
- Subscription Correlation ID addition (implicit subscription); and

Other AF can also subscribe to the *Nsmf_EventExposure* Service to receive notifications whenever some event is trigger as defined in TS23.502. Following are the events that can be subscribed by the NF consumer to get information from the SMF:

- UE Ip address
- DNN
- PDU session type
- QoS monitoring for URLLC
- change of access type
- PLMN change
- QFI allocation

3.3 Backhaul and Transport Networks

Backhaul and transport networks are typically responsible for connecting radio access/edge networks to the core network and can be wired, wireless or a mix of both. As described in Section 2.2, according to 3GPP, end-to-end network slices have to take into account that 5G NFs could be interconnected through transport networks, that can be either intra-RAN and intra-core domain, or inter-domain (in the sense that they interconnect RAN and core domains). This is shown in Figure 3-6, where indeed different types of transport networks are highlighted. In the RAN domain, a transport network can interconnect 5G RAN NFs, while in the core domain, it can interconnect 5G Core NFs.

From a management and orchestration perspective, transport networks are considered (by 3GPP) as “non-3GPP” systems that require specific management functionalities to interact with the 3GPP management system to receive the requirements in terms of transport network topology, QoS and

Co-funded by the Horizon 2020 Framework Programme of the European Union

network connectivity. It is worth to mention that, in the iNGENIOUS case, the 3GPP management system is mapped to the end-to-end network slice orchestration framework described in this document. Therefore, the 3GPP management system coordinates with the management systems of the non-3GPP parts when preparing a network slice instance. This coordination can be either direct or mediated by a MANO system.

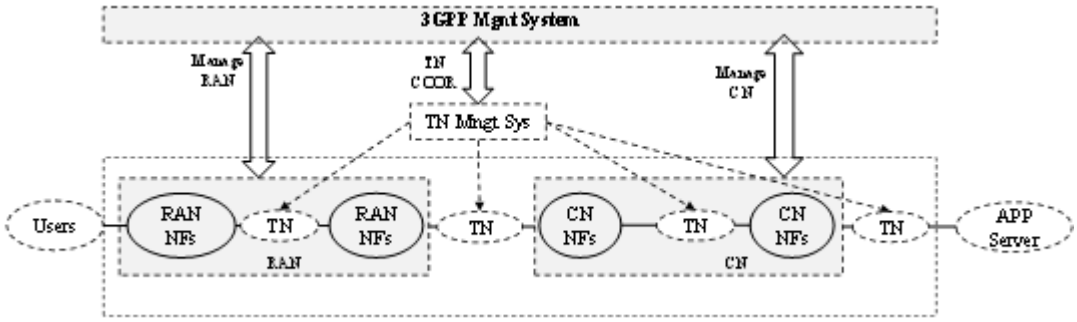

Figure 3-6 - Interaction between 3GPP and TN management systems [7]

The following subsections focus on a brief analysis of backhaul and transport network options considered in iNGENIOUS, and that from a conceptual and design perspective can be integrated with the proposed end-to-end network slice orchestration solution.

3.3.1 FIXED BACKHAUL AND TRANSPORT

In the context of end-to-end network slices to be provided across several network domains, including as mentioned above RAN, Core and transport, a common approach for the control and management of fixed transport networks (e.g., based on optical technologies) is to make use of SDN solutions. However, according to the differentiation of transport networks identified above and shown in Figure 3-6 (i.e., intra-domain and inter-domain), different approaches can be followed.

First, in the intra-domain case, the 3GPP management system (i.e., the iNGENIOUS end-to-end network slice orchestration framework), when RAN or 5G Core NFs are virtualized, leverages on NFV orchestration solutions to manage those NFs as VNFs and NFV Network Services. In this case, the 3GPP management system delegates to the NFV MANO the 5G NFs deployment and configuration into virtualized infrastructures that can span across several compute locations interconnected through a transport network. To properly manage the interconnection of the various 5G RAN or 5G Core VNFs across different compute locations the NFV MANO defines and uses a WAN infrastructure Manager (WIM), that is a particular case of the NFV MANO VIM component (see Section 2.1) [3]. While the VIM is responsible for controlling and managing the compute locations resources (for allocation of networking and computing resources required to execute the virtualized NFs), the WIM is used to establish connectivity between them and the given NFs running in it. In this case, the WIM can be implemented by a dedicated Transport SDN controller (e.g., based on opensource), and from a 3GPP management system perspective, the control and management of the intra-domain transport network is hidden and mediated by the NFV MANO system. Specifically, there is no need to export towards the 3GPP management system any transport

Co-funded by the Horizon 2020 Framework Programme of the European Union

network detail or specific requirement, as it is the NFV MANO system that translates the specific network slice requirements imposed by the 3GPP management system into transport network provisioning and configurations through the SDN controller.

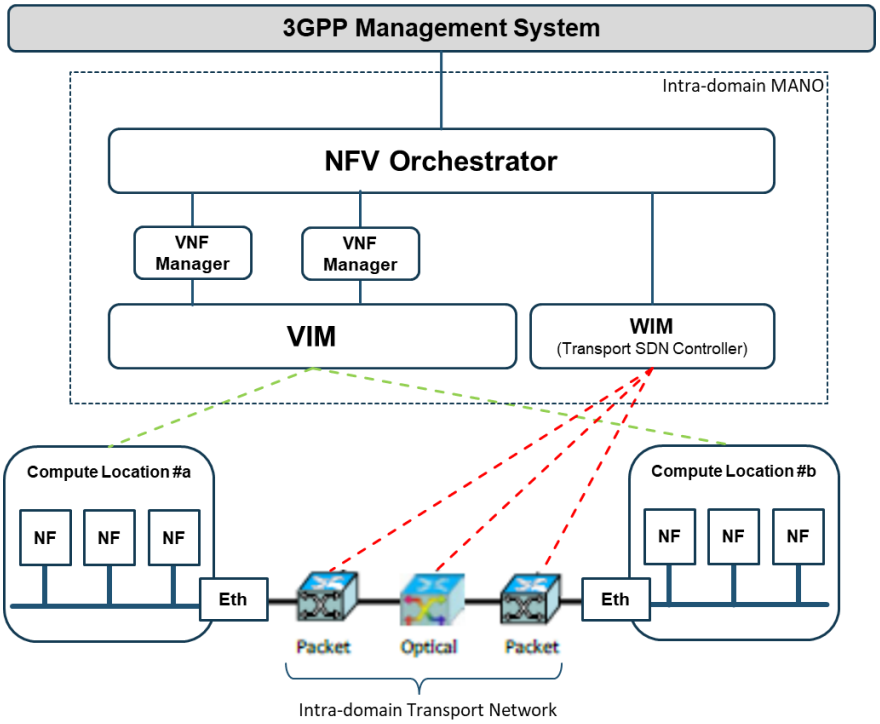

Figure 3-7 - Intra-Domain SDN Transport Network

In the second option, i.e., the inter-domain transport network used for interconnecting RAN/edge and core network domains, the 3GPP management system should directly interact with a Transport SDN control framework.

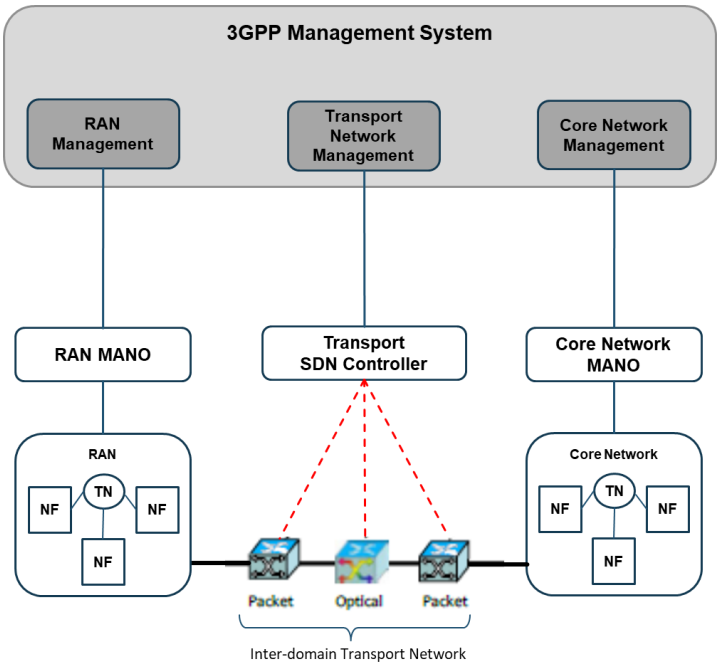

Figure 3-8 - Inter-Domain SDN Transport Network

Co-funded by the Horizon 2020 Framework Programme of the European Union

As depicted in Figure 3-8, the 3GPP management system (i.e., the iNGENIOUS end-to-end network slice orchestration framework), should provide specific transport network management functionalities capable to interact with Transport SDN controllers and thus retrieve information on available topologies, QoS capabilities, and enforce transport network connectivity services in support of end-to-end network slice performance requirements.

However, due to the traditional complexity of transport networks, either in the intra-domain or inter-domain case, which are normally composed by multiple network nodes featuring diverse technologies (i.e., packet, optical) and provided by different vendors, the case of having a single Transport SDN controller is not realistic. Current transport networks are indeed fragmented into multiple vendor domains with their own control plane technology. For this reason, the Open Networking Foundation Transport API (ONF TAPI) is a good candidate solution to overcome this fragmentation [20]. Indeed, the ONF TAPI enables to abstract a set of common SDN control plane functions (e.g., for path computation, topology, connectivity services provisioning) and defines a common data model and protocol based on YANG [21]. Moreover, it defines common APIs to interact with heterogeneous SDN controllers, independently of the specific SDN proprietary logic implemented. In particular, the ONF TAPI approach facilitates the deployment of hierarchical SDN architectures, with the specific common APIs used as the northbound interface of the lower layer SDN controller and as southbound interface of the upper layer SDN controller, enabling a recursive approach with cascades of SDN controllers.

3.3.2 SATELLITE BACKHAUL AND TRANSPORT

A satellite backhaul connectivity deployment includes UEs (e.g., IoT devices) connected to an edge node which is connected to, or integrated with, a satellite terminal. The satellite terminal communicates with the central node over a satellite link. The satellite backhaul is seen as a transport layer for the messages between the edge and the central node. Because of this, the backhaul should be as transparent as possible, while at the same time being able to assure a guaranteed communication quality depending on the requirements of the use case.

IoT devices send regular status updates to an IoT Gateway (GW) which processes and optimizes the data before forwarding via the satellite backhaul link to the IoT cloud/data centre as shown in Figure 3-9.

As illustrated in Figure 3-9, the satellite network architecture used in iNGENIOUS also incorporates major concepts and components of the 5G architecture to provide the end-to-end satellite connectivity. Firstly, the space segment is provided by SES' multi-orbit and multi-band transparent (bent pipe) satellite fleet which provides connectivity between the satellite remotes and the hub platform located at the SES' teleport in Betzdorf, Luxembourg. The satellite network deployed at the SES Teleport is built using IDR's 5G-enabled Velocity™ Intelligent Gateway (IGW) system and uses satellite capacity provided by SES's Ku Band satellite. This architecture is captured by ETSI SES in ETSI SES - *DTR/SES-00405 - TR 103 611: Satellite Earth Stations and Systems (SES); Seamless integration of satellite and/or HAPS (High Altitude*

Co-funded by the Horizon 2020 Framework Programme of the European Union

Platform Station) systems into 5G systems specifically “Scenario A3 - Indirect mixed 3GPP NTN access with bent-pipe payload”.

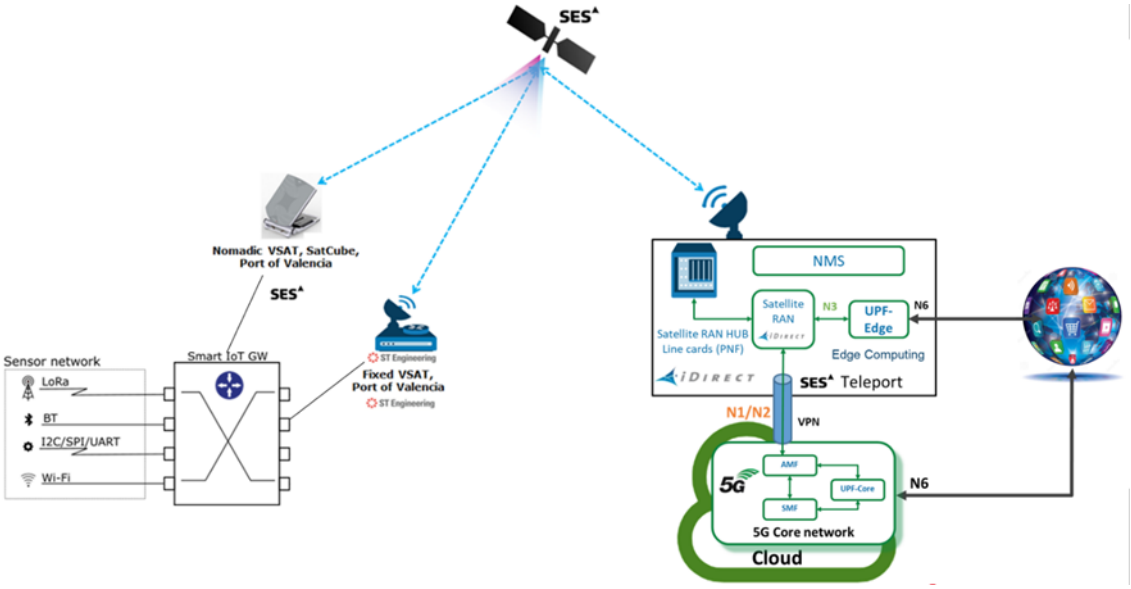

Figure 3-9 - Satellite backhaul connectivity architecture

A satellite system can be used as a transport network within the IoT network to provide connectivity between remote and central nodes. Typically, the satellite system is not managed by the same entity that manages the end-to-end IoT network.

Management and Orchestration

Reaching across a network to provision an end-to-end service is certainly challenging in any network with diverse control across a multi-layer and heterogeneous service network. However, in satellite networks there is the constant challenge of variable performance and reliability combined with greater latencies of GEO and Non-GEO satellite links that can add hundreds of milliseconds of latency. Centralized Orchestration or SDN Controller architectures are vulnerable to a high frequency of command exchanges across the space segment. Furthermore, synchronized message exchanges suffer on high latency and variable availability links. These demanding operating conditions need a resilient and recoverable control architecture that provides durability in the face of variable performance that a physically centralized architecture does not sustain. A distributed and possibly federated SDN Controller architecture puts greater intelligence and autonomy at the edge beyond the space segment, minimizing the dependency and use of space segment resources using distributed intelligence to maintain a rich control interface with local devices where terrestrial networks are typically available [22]. Figure 3-10 illustrates specialized controllers for various layers and segments of the network where each controller can support either the centralization or distribution of network control functions. These functions can typically be implemented as virtualized network functions that are placed at the network edge in devices that support a virtualization infrastructure.

Co-funded by the Horizon 2020 Framework Programme of the European Union

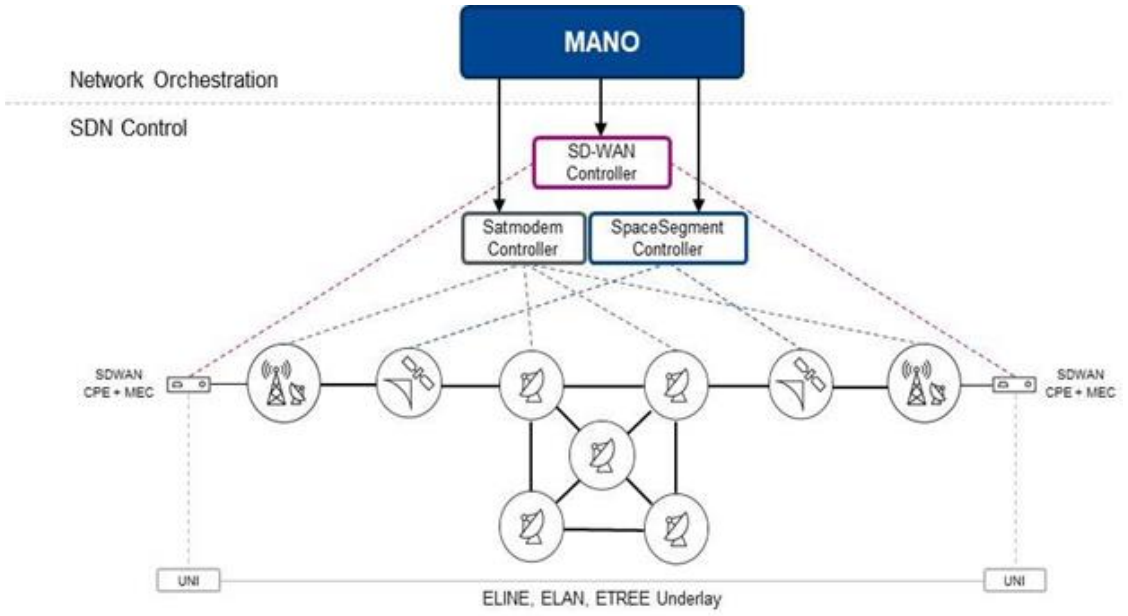

Figure 3-10 - Distributed SDN Controller Architecture

When the MANO system requests a service from the satellite system, the internal network controller (e.g., SDN) will "spin up" the necessary virtual functions and perform the necessary checks to provide the service before responding to the MANO system indicating if the service is available or not.

It is important to note that, typically, the MANO system itself is not allowed to spin up resources within the satellite network. The satellite system needs to be responsible for this as it has limited resources and the satellite system is the only one with a full view of the resources available.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4 End-to-end Network Slice Orchestration Architecture

This chapter represents the core part of this document, and provides a detailed description of the iNGENIOUS end-to-end network slice orchestration framework architecture. In particular, it highlights the architectural principles that are the foundation of the design of the orchestration framework, that is then detailed in the form of a functional architecture with the identification of the main building blocks required to achieve the needed management and orchestration capabilities withing the iNGENIOUS Network Layer. In addition, operation workflows, information models and APIs are also detailed to provide a full design picture for the proposed end-to-end network slice orchestration framework.

4.1 Architectural Principles

The iNGENIOUS end-to-end network slice orchestration framework has been designed considering the overall requirements defined in Deliverable D4.1 [15], and satisfy the architectural principles gathered in Table 4-1:

Table 4-1 - Architectural principles of end-to-end network slice orchestration framework

Principle #1: *The end-to-end network slice orchestration architecture should follow the global structure of the 5G system defined in the 3GPP specifications and make use of the latest technologies and architectures in the area of Network Function Virtualization*
Principle #2: *The end-to-end network slice orchestration architecture should be aligned with the major 3GPP and ETSI standards in terms of functional architecture and interfaces with aim of facilitating interoperability and integration with 5G infrastructure deployments.*
Principle #3: *The design of the end-to-end network slice orchestration framework should maximize the re-use of existing architectural components from 3GPP and ETSI NFV specifications, e.g., in terms of management functionalities, MANO components, etc. When new functions or components are required, their interfaces should be designed to facilitate their integration with the existing standard frameworks.*
Principle #4: *The end-to-end network slice orchestration should be augmented with closed-loop functionalities to achieve a high degree of automation in service and network slice operation. The integration of AI/ML solutions and technologies should be considered to go beyond current reactive closed-loop approaches in favor of proactive optimization solutions.*
Principle #5: *The end-to-end network slice orchestration architecture should enable the implementations of its components as cloud-native services, easing the deployment in edge and cloud environments, in a modular, dynamic and orchestrated way.*
Principle #6: *The end-to-end network slice orchestration framework should make use of open interfaces and APIs to facilitate its integration with third party systems and avoid vendor lock-ins.*

Co-funded by the Horizon 2020 Framework Programme of the European Union

> **Principle #7:** *The design of the end-to-end network slice orchestration architecture should follow a modular pattern that enables its applicability to multiple use cases and deployment scenarios. It should facilitate composition and customization of the functional blocks according to accommodate specific requirements of the target use case domains and required features.*

4.2 Functional Architecture

The main principles and motivations described in the previous section led to the specification of the iNGENIOUS end-to-end network slice orchestration framework. In Figure 4-1 is available a mapping between the functional architecture described the 3GPP TR 28.801 specification (Figure 4-1a) and the proposed high-level architecture of the end-to-end network slice framework, which is assisted by cross-layer AI/ML functionalities in support of the network slice operations. (Figure 4-1b).

Figure 4-1b shows the three main functional blocks, namely Vertical Service Management Function (VSMF), Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF), which play a specific and crucial role in the proposed orchestration framework. In particular:

- The VSMF layer is in charge of the lifecycle of vertical service instances, i.e., a service with high-level requirements. The VSMF translates the vertical service requirements into end-to-end network slice requirements.
- The NSMF layer is in charge of the lifecycle of end-to-end network slices. Furthermore, the NSMF interacts with different NSSMFs.
- The NSSMF layer is in charge of managing the specific lifecycle of the network slices subnet. This layer can include multiple instances of NSSMFs, one specific for each network domain (e.g., RAN, transport, core).

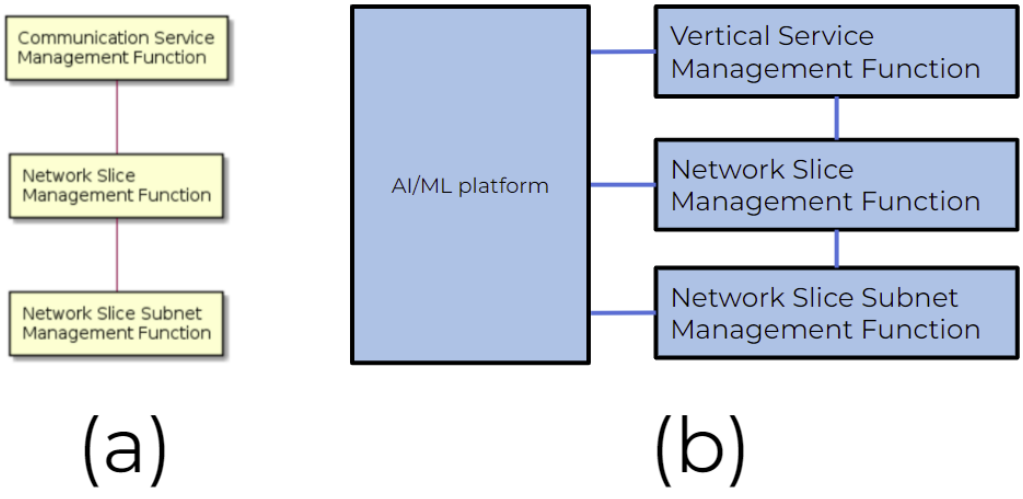

Figure 4-1 - End-to-end network slice architecture functional high-level architecture supported by AI/ML platform

Co-funded by the Horizon 2020 Framework Programme of the European Union

The number and type of end-to-end network slices applicable and suitable for a given vertical service strictly depend on its high-level requirements and application scenario. For instance, a uRLLC and eMBB end-to-end slices can coexist on the same physical network infrastructure. The former can be referred to as an industry 4.0 scenario (e.g., robot communication service), while the latter as a video streaming communication service with a fixed QoS (e.g., video resolution).

From an architectural perspective, the orchestration framework uses a cross-layer approach, meaning that each functional component described above is dedicated to manage and coordinate specific service, network slice and resource operations, with tight cooperation to fulfill end-to-end and cross-layer consistency. The information available at the VSMF level is kept at the service level only, with abstraction in terms of network slice and resource details. On the other hand, at the NSSMF level the information managed is technology and vendor-specific. Therefore, the end-to-end network slice orchestration framework implements different mechanisms for translating the high-level requirements into technology and vendor-specific requirements. The end-to-end orchestration framework is also supported by an AI/ML platform to execute some automatic decisions in the operation of vertical services and network slices.

In the next sections, the main components of the proposed architecture (already briefly described above) are detailed. In particular, for each component the related functional decomposition is presented, including the information managed and the interaction with other layers.

4.2.1 ORCHESTRATION COMPONENTS

This section describes the internal components of the end-to-end network slice orchestration framework.

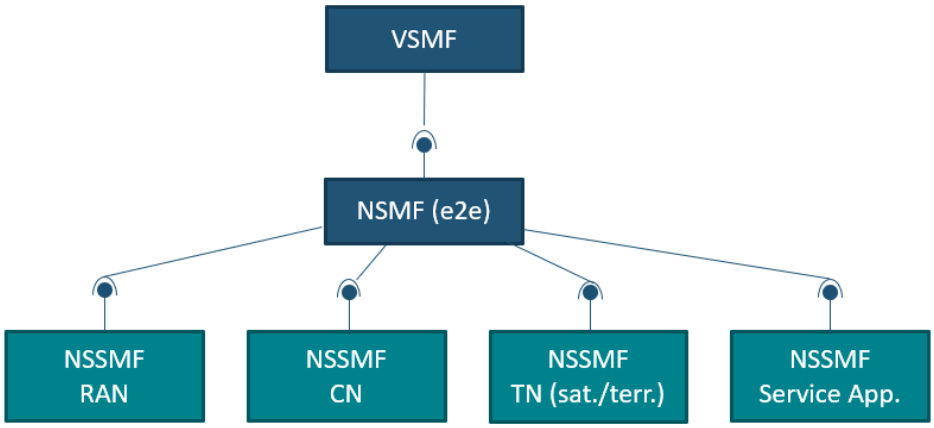

Figure 4-2 - High level software architecture end-to-end network orchestration framework

Figure 4-2 depicts the functional architecture of the end-to-end network orchestration framework, derived from the high-level view of Figure 4-1. In particular, the 3GPP CSMF functionalities are realized by the Vertical Service Management Function (VSMF), the 3GPP NSMF functionalities are realized by the end-to-end NSMF and finally the 3GPP NSSMF layer is mapped into multiple specific technology-tailored NSSMFs.

Co-funded by the Horizon 2020 Framework Programme of the European Union

After a brief description of the network slice related data models supported by the end-to-end orchestration framework (which is key to capture how the various entities managed are modelled), the following sub-sections detail the functional decomposition and internal design of the VSMF, NSMF and NSSMF components.

4.2.1.1 Data models

The end-to-end network slice orchestration stack introduced above supports a multi-layered data model. This is used by each orchestration component to drive the lifecycle management operations and derive any requirement concerning services and network slices, and thus enforce the proper actions and invoke primitives in the lower layer components.

At the upper layer of the orchestration stack, the VSMF implements two different data models: the Vertical Service Blueprint (VSB) and the Vertical Service Descriptor (VSD). Both data models are based on a non-standard information model defined as part of the Vertical Slicer (the baseline Nextworks software stack used for the end-to-end network slice orchestrator [23]) and represent respectively a class of vertical services (VSB) and a specific vertical service belonging to a certain class (VSD). The VSB describes a vertical service through service parameters defined according to digital/communication service providers' knowledge. Indeed, it provides a high-level description of the service that does not include infrastructure-related information. The VSD is obtained from a VSB, when a vertical consumer selects a class of service (i.e., a VSB) and produces a Vertical Service description by specifying certain value of the VSB parameters, that may include resource specifications, QoS and geographical constraints, number of users consuming the service, and also reference to specific Vertical Functions.

As anticipated above, at the NSMF and NSSMF levels two different network slice data models are supported: the 3GPP Network Slice Template (NST) and the GSMA GST. The latter is then called Network Slice Type (NEST) once its attributes have been assigned proper values for a given service. The GSMA NEST allows the description of a Network Slice through value assignment according to the GSMA GST (GSMA, 2020). The main requirements expressed through the NEST consist of a list of 5G Quality of Service (QoS) Indicators (5QI), which are subsequently mapped into NST's parameters that determine the type of the Network Slice. In particular, such 5QIs are used in the GST-to-NST translation process to determine the 3GPP-based Service Profile specified in the NST.

The 3GPP NST describes a Network Slice according to the attributes defined by the 3GPP Network Slice NRM [8], that provides network requirements and related resources' configuration. In particular, the NST, whose simplified class diagram is shown in Figure 4-3, contains a list of Service Profiles, each of them specifying the Network Slice type and the related QoS and service attributes (e.g., the latency, the maximum number of UE, the maximum supported packet size etc.). In addition to the list of Service Profiles, the NST contains a reference to a Network Slice Subnet (NSS) that can be represented through a NSS Template (NSST) as part of the overall NST data model. The NSST contains a list of Slice Profiles, each of them representing the required properties of the NSS. The Slice Profile contains QoS attributes, similar to the Service Profile, and

Co-funded by the Horizon 2020 Framework Programme of the European Union

a list of Performance Requirements. The attributes included in the Performance Requirements depends on the type of the NSS. For instance, if the Network Slice type is uRLLC, the list of Performance Requirements will contain parameters like the E2E latency, the jitter, the message size byte and the communication service availability target. For eMBB network slices, the list of Performance Requirements can contain attributes like the experienced data rate, the area traffic capacity downlink and the area traffic capacity uplink. Finally, two other attributes contained inside the NSST are a reference to a list of NSSTs and a Network Service Descriptor (NSD) Info field, which refers to the NFV Network Services that may be included into the NSS.

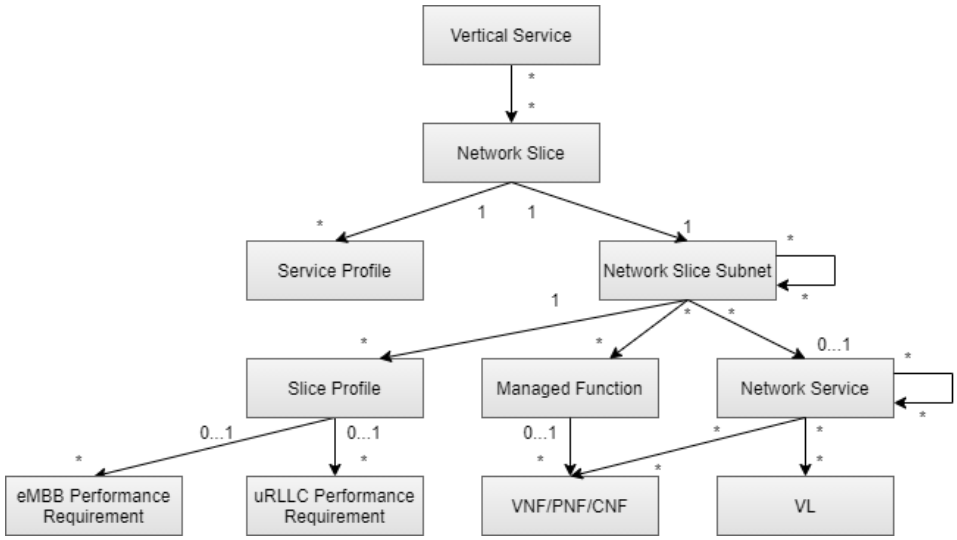

Figure 4-3 - Network Slice Simplified Class Diagram

Translation process

As part of the end-to-end network slice orchestration steps, together with the slice resource allocation operations, the iNGENIOUS cross-layer orchestration stack implements a stepwise procedure where the vertical consumer intent expressed in the VSB and VSD to describe the desired service is gradually translated into the slice and related resources technical requirements that follow standard definitions.

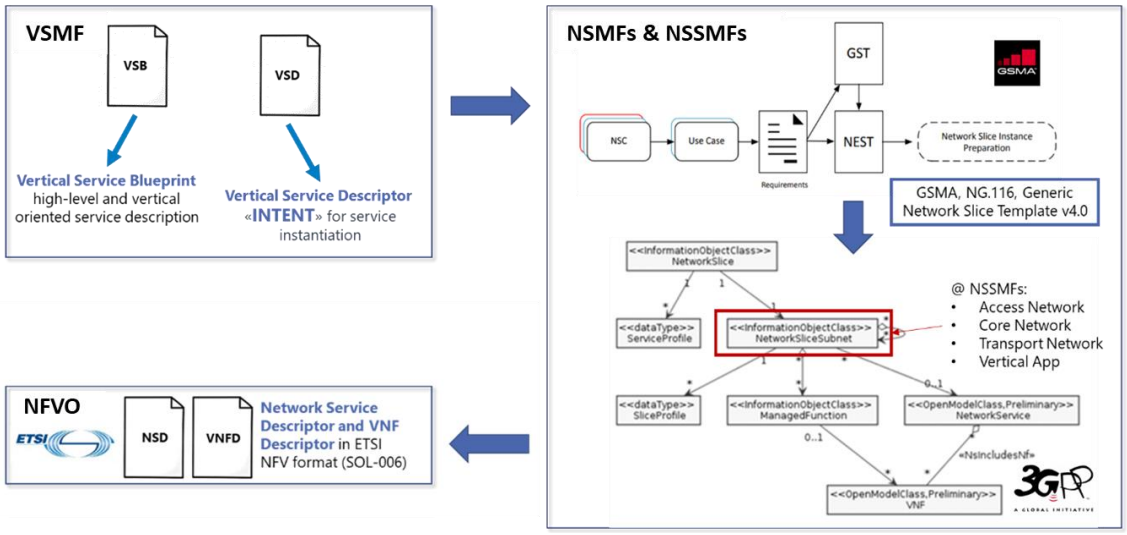

Figure 4-4 - Translation of Vertical Service Requirements into end-to-end slice resource allocation

Co-funded by the Horizon 2020 Framework Programme of the European Union

As anticipated above, the first step consists in the vertical consumer creating a VSD specifying tailored values for the VSB parameters (e.g., through a dedicated User Interface offered by the VSMF). After this, the vertical may request the deployment of the vertical service intent expressed in the VSD, thus triggering the translation process to standard network slice technical models. From the VSD are extracted information to build a slice specification following the GSMA-GST model. At this point, first, a NEST is produced, then the Network Slice is defined following the 3GPP NRM model, that incorporates the concepts of end-to-end slice and network subnet slices, in order to model and manage the slices at each network segment. For the core/edge segment the slice subnet can include a set of NFV Network Services and VNFs (e.g., to model either the 5G Core NFs or specific vertical virtual service applications), defined by following the ETSI NFV SOL-006 [24] information models for the VNF Descriptor (VNFD) and the Network Service Descriptor (NSD).

4.2.1.2 Vertical Service Management Function

As already mentioned, the VSMF is in charge of managing the requests of vertical service lifecycle management exploiting the related data model, i.e. the Vertical Service Blueprint (VSB) and Vertical Service Descriptor (VSD). Specifically, the VSB is a template used for representing a class of services. It contains parameters like the number of users, covered geographical area by the service and so on. VSD is the parametrization of the defined VSB, specifying for instance the actual number of users the service, the actual geographical area where the service would be deployed and so on.

In general, each vertical service is associated with a Tenant that represents the vertical consumer/customer of the orchestration platform. However, each Tenant has a maximum amount of resources for the vertical service provisioning defined within a Service Level Agreement (SLA). Therefore, the VSMF implements operations to manage the Tenant according to its specific SLAs information.

Figure 4-5 depicts the functional decomposition of the VSMF, including the interactions among the internal components and with the NSMF.

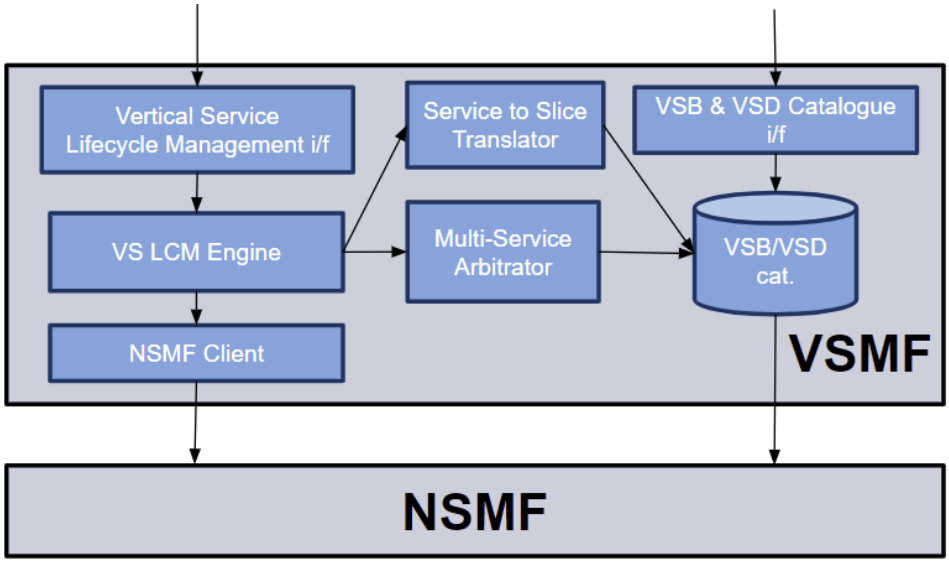

Figure 4-5 - High-level software architecture of VSMF

Co-funded by the Horizon 2020 Framework Programme of the European Union

In general, the main aim of the VSMF is to manage the lifecycle of multiple vertical services in a seamless way. For this reason, different functionalities are supported by its internal components. The two main entities that interact with the VSMF at its northbound are:

- the network/admin operator for managing the onboarding of VSBs, and the configuration of the Tenants and related SLAs
- the vertical consumer/customer (i.e., the Tenant) for requesting lifecycle operations of vertical services (e.g., instantiation, modification, termination)

A brief description of each VSMF internal component (with reference to Figure 4-5) is reported below.

VSB & VSD catalogue

This component allows performing Create, Read, Update and Delete (CRUD) operations on VSBs and VSDs used for managing the vertical service capabilities and its high-level requirements. This catalogue not only is used by the network/system administrator, but also by the other VSMF internal services. Moreover, the VSB and VSD catalogue can interact with the NSMF for retrieving technical network slice related information related to the vertical service.

REST APIs related to CRUD Operation on VSBs and VSDs are documented in section 4.5.

Vertical Service (VS) Lifecycle Management (LCM) Engine

This component, with the support of the Service to Slice Translator, Multi-service arbitrator and the VSB & VSD Catalogue, manages the whole lifecycle of vertical services instances (VSIs), from their instantiation to their termination. Moreover, the VS LCM manages also the runtime modifications of the VSIs. A dedicated Vertical Service LCM interface is exposed to allow LCM operations triggered by Tenants and admin/network administrator.

The related REST APIs are documented in section 4.5.

Service-to-slice translator

This component translates the vertical service into end-to-end network slice capabilities. In particular, it maps the high-level service requirements provided by the VSD into technical specifications available in the NST/GST catalogue of the NSMF. The translation process follows the translation rules provided by the admin/network administrator during the VSB on-boarding process.

Multiservice arbitrator

This component determines the vertical service instantiation feasibility when a new vertical service is requested to be provisioned. In detail, it checks whether the request of the instantiation does violate or not the SLA in terms of maximum resources that can be allocated for a specific Tenant. Moreover, it allows, with specific arbitration policies (that can be pre-configured) to share network slices to serve different vertical services (e.g., from the same Tenant).

Co-funded by the Horizon 2020 Framework Programme of the European Union

NSMF Client

This component is responsible for interacting with the NSMF, sending requests for different operations related to end-to-end network slices. These operations include the instantiation, modification and the termination of the end-to-end network slices. In general, the NSMF Client is in charge of dispatching all the requests towards the end-to-end NSMF coming from the VS LCM Engine and managing the corresponding responses. Furthermore, the interaction mechanism with NSMF is asynchronous: once the request is sent and a reply is received, it means that it has been submitted. Then, when the end-to-end network slice has been successfully provisioned, a notification is sent back to the VS LCM Engine..

4.2.1.3 Network Service Management Function

Figure 4-6 depicts the high-level architecture of the NSMF with its internal components and the related interaction among them. The NSMF is mainly responsible for managing the lifecycle of end-to-end network slices, according to the requirements and capabilities expressed in the Generalized network Slice Template (GST) and Network Service Template (NST).

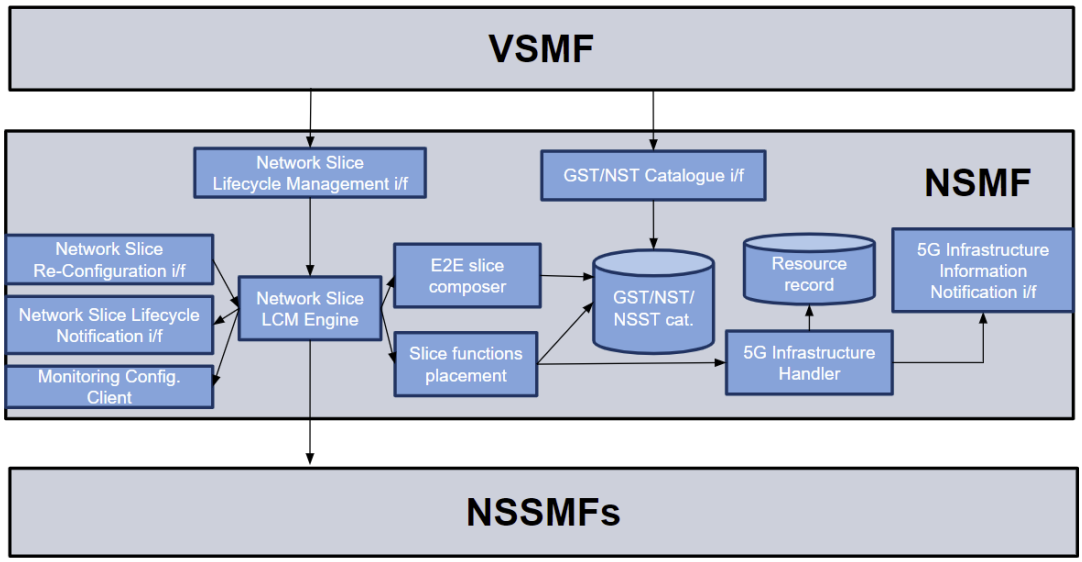

Figure 4-6 - High-level architecture of NSMF

As already described, the GST defined by GSM Association (GSMA) contains a set of attributes for defining a generic network slice regardless of the technology used for the network slice provisioning itself. The GST results in a NEST when GST's attributes are associated with a specific value [11]. Similarly, The NST and NSST, in compliance with the 5G NRM [8] (also detailed in Section 2.2.1), describe through an abstract model the slices' capability, without explicitly stating the internal technical details of the network slice itself. GSTs, NSTs and NSSTs drives the whole lifecycle management of end-to-end network slices, implemented by the different components available within the NSMF.

A brief description of each NSMF internal component (with reference to Figure 4-6) is reported below.

Co-funded by the Horizon 2020 Framework Programme of the European Union

GST and NST catalogue

This component allows to perform CRUD operations on GSTs, NSTs and NSSTs used for defining the end-to-end slice requirements. In particular, a admin/network administrator can manage the GSTs\NSTs, making them available for possible end-to-end network slice instantiation requests. Moreover, the GST/NST catalogue can be queried by the VSMF for the translation process during a vertical service instantiation. For this reason, the GST/NST catalogue exposes a set of REST API is detailed in section 4.5.

End-to-end network slice LCM engine

This component manages the lifecycle of the end-to-end network slices. With the support of the E2E Slice Composer and Slice Functions Placement Service, it manages the set of requests to be sent to the corresponding NSSMFs. The end-to-end network slice LCM engine is triggered by the NSMF client of VSMF through the Network slice lifecycle management interface. This interface (whose rest API are available in section 4.5), allows performing the instantiation, re-configuration and termination of end-to-end network slices. This interface can be also used by an external entity such as an AI/ML platform to automatically trigger the LCM operation to implement closed-loops in the network slice operations.

The interaction with different NSSMFs is asynchronous: each network domain may require a certain amount of time for provisioning the technology-specific resources for the end-to-end network slice. For this reason, an asynchronous approach is used, i.e., the NSMF sends the request and it will receive notifications about the Network Slice Subnet instantiation request.

NSMF services for AI/ML

On the left side of the NSMF architecture depicted in Figure 4-6, the Network Slice reconfiguration interface, Network slice lifecycle notification and Monitoring configure client components are included. These components provide the information related to the lifecycle of the end-to-end network slices to external components (e.g., the AI/ML platform) and allow to perform LCM operations on the end-to-end network slice itself. For instance, the AI/ML engine can be notified about the creation and evolution of network slice instances, possibly retrieving the related capabilities, and enforce runtime operations (e.g., network slice modifications or optimizations) according to the final aim of the specific AI/ML algorithms.

End-to-end slice composer

The main aim of this component is to support the LCM engine into composing end-to-end slice templates, by identifying the related subnets information, from the GST/NST requirements issued by the VSMF. Using the specific GST/NST the network slice instantiation request is referring to and the different network slice subnets available, it composes the end-to-end network slice to be created. In this way, the NSMF can interact with the corresponding NSSMFs, specific for each segment and network domain of the 5G network. More details about the interaction between the NSMF and NSSMFs are detailed in section 4.4.

Co-funded by the Horizon 2020 Framework Programme of the European Union

5G Infrastructure handler

The main purpose of this component is to maintain information on the computational and network capabilities of the available 5G infrastructure, providing a topology view of how the multi-domain 5G network is composed. This information can be exposed to the AI/ML platform through the 5G Infrastructure information notification interface.

4.2.1.4 Network Slice Subnet Management Function

The NSSMF layer is a collection of different NSSMFs. Depending on the specific deployment scenario and specific 5G network infrastructure where the orchestration framework operates, the number and the type of NSSMFs can change. In the case of iNGENIOUS, the high-level architecture of the NSSMF layer is depicted in Figure 4-7.

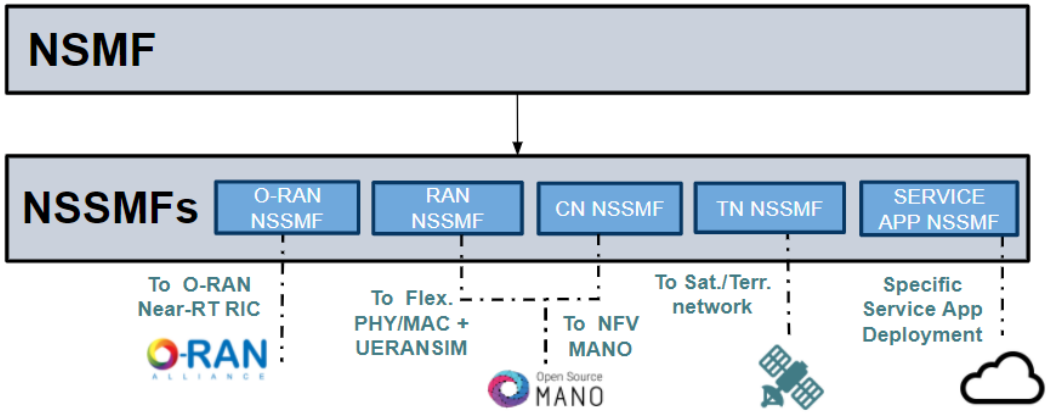

Figure 4-7 - High-level architecture of NSSMF layer

Each specific network domain implements its own mechanisms, data models, REST APIs and workflows for allocating computing and network resources. For this reason, a tailored NSSMF implementation is needed to deal with the domain-specific controllers or local orchestrators, such as NFVOs, RAN controllers, SDN controllers, etc. Furthermore, the technical details of the domain are hidden by an abstraction layer each NSSMF provide: this approach allows the NSMF to deal transparently and uniformly with all the NSSMFs, providing flexibility to the NSMF perspective. All NSSMFs follows a generic and unified functional decomposition, which is show in Figure 4-8.

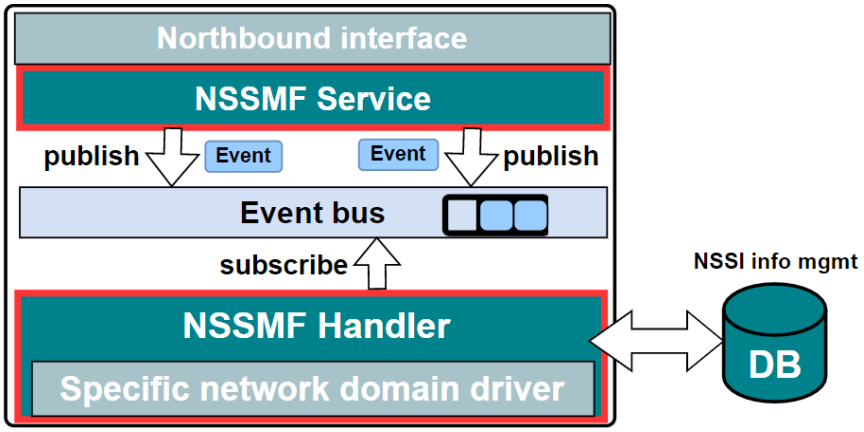

Figure 4-8 - High-level architecture of generic NSSMF

Co-funded by the Horizon 2020 Framework Programme of the European Union

In particular, each NSSMF provide at least the following functionalities:

- **The Northbound Interface (NBI)** is exposed for exploiting the NSSMF functionalities and for receiving subnet slice related requests. From an implementation perspective, REST APIs are used for interfacing with the NSSMF (i.e., from the NSMF).
- **The NSSMF Service** plays the role of validating and dispatching the requests into an event bus, publishing them as events. Moreover, the NSSMF service can perform a subscription to the event bus for receiving notifications related to the requests dispatched.
- **The Event bus** allows communications among components using a topic-based and publish-subscribe mechanisms. In the case of the requests of the NSSMF towards the specific network domain, the NSSMF service is the publisher of the events, i.e. the requests, while the NSSMF Handler represents the subscriber. Similarly, for the notifications, the NSSMF Handler is the publisher while the NSSMF Service is the subscriber.
- **The NSSMF Handler** subscribing to the events receives multiple requests and realizes the internal logic of the NSSMF. For instance, it translates the request appropriately and invokes the specific network driver domain to send the processed request. On the other hand, it could receive notifications from the specific network domain and dispatch them into the event bus. Each specific NSSMF extends this NSSMF Handler with custom translation logic depending on the specific managed domain and related services and resources.

As anticipated, from a software implementation perspective, each specialized NSSMF has its tailored realization: internal logic of NSSI provisioning, payload information model and workflow interactions with the corresponding network domain controllers/orchestrators strictly depend on the technology, vendor, interfaces supported. In the particular case of iNGENIOUS, the NSSMFs included in the overall design are the following:

- **O-RAN NSSMF**: providing the translation of slice profiles into O-RAN A1 policies and A1 policy management operations in the O-RAN Near RT RIC. Early implementation and integration of O-RAN NSSMF with a Near-RT RIC provided by UPV have been described in deliverable D4.3 [25]. In section 5 a software prototype description and early integration activities are described.
- **RAN NSSMF**: providing automatic LCM and custom configuration of NFV Network Services for RAN NFs through ETSI OSM [26], an ETSI-led opensource implementation of the NFV MANO architecture, which is used in iNGENIOUS as NFV Orchestrator. The Network service contains a UERANSIM [26] instance, a tool for emulating gNB and UEs and providing 5G connectivity. Specifically, the connectivity is provided to the flexible PHY/MAC base station (provided by TUD), which UEs are represented by the SDR Platforms. Moreover, the RAN NSSMF configures the Flexible PHY/MAC allocating the correct amount of resources for each UE.
- **Core Network NSSMF**: providing automated LCM and configuration of 5G Core NFV Network Services through ETSI OSM [26]. The network service

Co-funded by the Horizon 2020 Framework Programme of the European Union

contains a 5G Core instance, provided by Cumucore, consisting of the Control Plane and User Plane Network Functions of a 5G Core. Details about the 5G Core can be found in section 3.2. The CN NSSMF manages the subnet slices available in the 5G network interfacing with the available REST APIs.

- **Transport Network NSSMF**: providing the allocation of network resources (i.e., in the form of connectivity services) in the transport network, when available to interconnect 5G RAN/edge and core network domain. Details of backhaul and transport network and its management and orchestration can be found in section 3.3.
- **Service Application NSSMF**: providing automated LCM and configuration of NFV Network Services modelling service virtual applications through ETSI OSM [26].

4.2.2 AI/ML AND MONITORING PLATFORM

As anticipated above, beyond the pure orchestration features, the iNGENIOUS end-to-end orchestration framework will provide closed-loop functionalities through the integration of a dedicated AI/ML and monitoring platform. First, the implementation of a closed-loop concept to fully automate the runtime optimization and adaptation of network slices requires knowledge on status and performance of (at least) the various involved NFs, network and computing resources. For this, specific monitoring capabilities have to be considered as key to collect and store relevant data on how the provisioned network slice instances (and the related resources) behave. Moreover, with the aim of going beyond the traditional reactive approach in fault and performance management, iNGENIOUS targets the implementation of predictive, proactive and automated network slice runtime operation. For this, the end-to-end network slice orchestration framework makes use of AI/ML techniques to assist the decision-making processes mostly at the network slice management (and thus NSMF) level.

Therefore, the end-to-end network slice orchestration framework relies on an AI/ML and Monitoring Platform which is designed with the main purpose of supporting automated lifecycle management procedures for the optimization of network slices related resources (both network and computing). In practice, it aims at collecting metrics and information from heterogeneous resources, providing a variety of data inputs to AI/ML based analytics and decision algorithms that can feed and assist the NSMF. The proposed platform is kept agnostic with respect to the specific algorithms consuming the monitoring data, and provides two ways for accessing the data. First, it offers query-based access to retrieve historical or periodical data, for example for the training of ML models. Second, it implements a subscribe/notify mechanism that allows to access streams of real-time data and can be used for real-time inference.

Figure 4-9 shows the high-level functional architecture of the AI/ML and monitoring platform. It is implemented through the integration of different data management open-source tools, augmented with additional ad-hoc components (such as the Configuration Manager and the Adaptation Layer) to ease the integration with the network slice orchestration components. As

Co-funded by the Horizon 2020 Framework Programme of the European Union

shown in the figure, the AI/ML and monitoring platform is built by the interaction of two building blocks: the monitoring platform and the AI/ML engine.

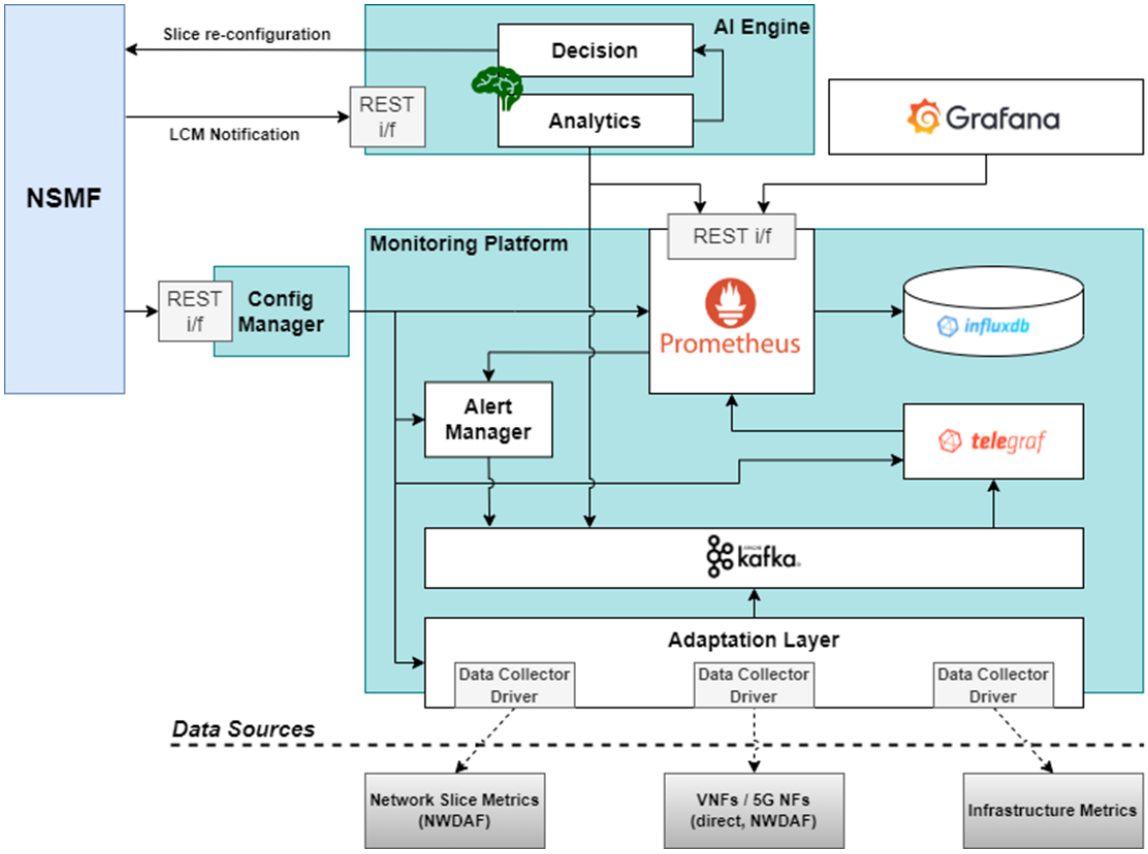

Figure 4-9 - AI/ML and monitoring platform functional architecture

The monitoring platform provides both data storing and streaming functionalities, with proper interfaces exposed towards the AI/ML engine to consumes the monitoring data. The data can be collected from different and heterogeneous data sources through the Adaptation Layer, that provides the necessary interfaces and logic to map the data from the sources to proper messages topics on the internal data bus. In particular, the Adaptation Layer is designed to be plug-in oriented, where each plug-in (or data-collection driver) collects data from a specific data source. This approach provides a high level of flexibility since the composition of the active plugins may vary with respect the different network slices to be monitored or during the different phases of a network slice lifetime.

The data streaming service is provided by a central Apache Kafka Bus [28]that is accessed by an instance a Telegraf [29] plugin, which acts as a data collection manager and listens to new messages on the bus to send them to an instance of Prometheus [30]. Prometheus provides data aggregation functionalities, and uses InfluxDB [31] as database to store the collected raw data in form of timeseries. Therefore, InfluxDB represent the Data Lake of the whole AI/ML and monitoring platform. In addition, a configurable Alert Manager (which is a built-in component of Prometheus) sends alarms to the bus when specific data exceeds certain thresholds, so that the alarm notification can be captured by Telegraf and stored in the Data Lake. The alarms and the data, both historical and near-real time are therefore

Co-funded by the Horizon 2020 Framework Programme of the European Union

immediately available to the AI/ML engine that can access both the Data Lake and the Kafka Bus through dedicated interfaces. The whole monitoring platform is configured by the NSMF through the Config Manager, that for each network slice instance can tailor the behaviour of the monitoring platform to properly collect, manage and store the required data. Indeed, the Config Manager provides the logic for configuring Prometheus to properly aggregate the data collected through the Kafka bus. Similarly, the Alert Manager is configured to produce different types of alerts when a given metric is exceeding a specific threshold. Moreover, the Config Manager is also responsible for the configuration of the different Data Collector Drivers to tailor the data collection from the various available sources according to the given network slice requirements, including Telegraf and Kafka configuration to create ad-hoc topics on the bus and consume the related data to be forwarded to Prometheus.

The AI Engine is divided into two functional blocks, Analytics and Decision. The live data inputs are obtained by the Analytics block through the monitoring platform, with analytics performance and results reported in Grafana. The Decision block passes the determined slice adaptations to the network slice orchestration components.

The Analytics block can be subdivided into 4-stages designed for robust functionality on real world data:

- Stage 1. Data pre-processing - real-time data contains many irregularities (ex. null values) unrelated to the useful information derived from the target analysis. This noise can directly affect the ability of models to reliably infer behaviours in the incoming data. The data pre-processor cleans and normalises the incoming dataset to avoid misbehaviour of the model on real world data.
- Stage 2. Feature detection - correlated time series data is analysed with respect to long term behaviours which create unique features that can be used to predict future trends in network behaviour. Selection of these features is achieved through the use of trained models capable of discriminating target behaviours.
- Stage 3. Inference engine - inference of future trends is performed using the identified features of the incoming dataset that are used as inputs in AI/ML algorithms to determine the most probable future state of the system. These predictions are then sent to the scaling logic to determine the most appropriate system adaptation.
- Stage 4. logic - the predictions of the state of the system are combined with operational parameters to decide if, how, and when an adaptation will optimise the resources of the system. The logic interacts with the NSMF to accept any changes to the slice reconfiguration.

For what concerns the interaction with the network slice orchestration components, the AI/ML and Monitoring Platform offers a set RESTful APIs on top of the Config Manager and the AI/ML Engine. The purpose of the Config Manager API is to enable the automated configuration of specific monitoring jobs from the NSMF. Indeed, during the provisioning of the end-to-end network slice instances, through this API the NSMF can trigger the monitoring of specific service and network related metrics, to be then stored in the data

Co-funded by the Horizon 2020 Framework Programme of the European Union

lake, visualized in customized dashboards, and consumed by the AI/ML Engine. On the other hand, the AI/ML Engine offers an API that is exploited by the NSMF to notify the analytics and decision functionalities about the evolution of network slices lifecycle (e.g., instantiation, scaling, termination) as well as on the result of the related lifecycle operations (i.e., success or failure) to help in the contextualization of data retrieved from the monitoring platform.

4.3AI/ML based network slice optimization

AI/ML techniques are being adopted into 5G networks to support full automation in closed loops related to the management and runtime operation of 5G services and network slices. In practice, the target is to improve the optimization of network performances, while enhancing the users perceived experience. At the same time, AI/ML techniques can help in solving network management complexities brought by 5G, where several technologies and domains coexist for the provisioning of end-to-end services and slices. Currently, this requires ad-hoc integrations and knowledge of heterogeneous per-domain control and management solutions. Exploiting data that can be easily collected from the 5G infrastructure, network functions and applications, AI/ML techniques can therefore help in fully automating 5G network services and slices runtime operations with a truly closed-loop approach.

In particular, the concept of network self-X (self-healing, self-optimization, etc.) based on the continuous monitoring of service attributes and performance parameters (data-driven), is a well-known approach in the context of 5G management platforms. The iNGENIOUS end-to-end network slice orchestration framework implements such automation mechanism by involving all the components building the platform: the orchestration stack, the monitoring platform and the AI/ML engine. Indeed, when an end-to-end network slice is deployed, the orchestration platform (i.e., through the NSMF), as final step, configures the Monitoring Platform in order to continuously collect data which are relevant to determine the current status of the slice itself and the related services. The collected data are related to the different network subnet slices and their resources (e.g., 5G Core NFs, virtual applications, etc.). The monitoring platform collects and stores the data and make them available for the AI/ML engine that continuously takes decisions based on the monitored status, that can be a simple “do nothing” or slice optimization requests to be enforced towards the slice re-configuration interface offered by the NSMF. At this point the NSMF translates such requests to real actions on the target (monitored) end-to-end slice.

The AI/ML innovation scenario considered for the end-to-end network slice optimization targets the trigger of a pre-emptive auto-scaling of local-edge and central User Plane Functions (UPFs), in support of low latency communication services, as shown in Figure 4-10. A single UPF instance can handle multiple protocol data unit (PDU) sessions, however the resources of a UPF instance are finite. As traffic load increases, to avoid degradations in service caused by finite resources, more UPF instances can be deployed and started, and likewise, an idle UPF instance can be terminated when the traffic is low. This process can be achieved in a closed-loop continuous fashion that

Co-funded by the Horizon 2020 Framework Programme of the European Union

monitors, measures, and assesses real-time network data, then automatically acts to optimise according to the SLA. It is important to note that human operators configure the automated actions and can manually modify them at any point within the loop.

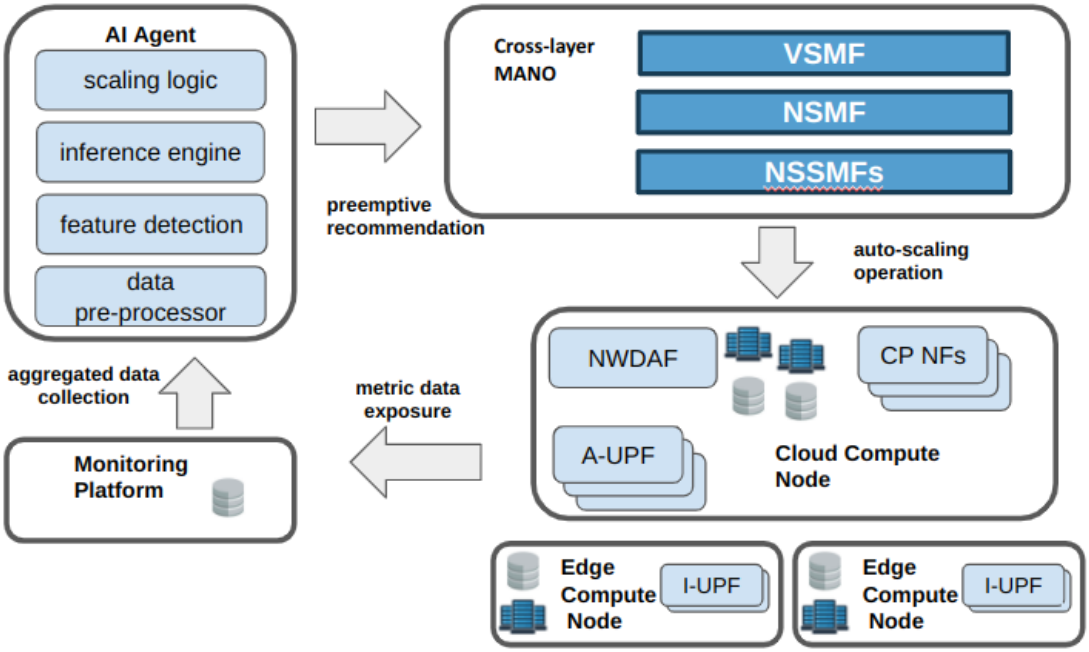

Figure 4-10 - Closed-loop pre-emptive auto-scaling of UPF

The information used in pre-emptive auto-scaling, collected from the 5G infrastructure, and applications, can be related to specific UEs, (mobility, communication pattern, etc.), NFs, network slices, or the network as a whole. UPF load information available from the NWDAF, including CPU, memory, and disk usage, can be supplemented with user plane data like bandwidth, latency, packet loss, etc., as well as UE-related information (mobility, position, etc.) to get accurate predictions of future network conditions. The current list of attributes considered by the AI Engine is given in Table 4-2, additional metrics are expected to be added as they become available. Within an Edge Compute Node, a local NWDAF collects data from the UPF and exposes it to the Monitoring Platform. The Platform collects the data from the NWDAF as well as other sources which are ingested after a pre-processing by the AI agent which performs a decision about the pre-emptive auto-scaling operation on UPF itself.

Table 4-2 - Data model of inputs to the AI Engine

Attribute (item associated with)	Source	Data type	Frequency	Description
# of PDU sessions (UPF)	CMC UPF logs	int	Event-based (Event=connection UE)	number of simultaneous sessions handled by UPF
NF load (UPF) at least one	NWDAF	int	polling or event based	CPU usage

Co-funded by the Horizon 2020 Framework Programme of the European Union

value will be given		int	polling or event based	memory usage
		int	polling or event based	Storage usage
		int	polling or event based	Avg load level
packet loss (UPF, session)	iperf3 output stats (UDP)	int	Using iperf3, at the end of each experiment. Experiment duration is a customizable parameter	percentage of packets lost with respect to packets sent
Number of retransmissions (UPF, session)	iperf3 output (TCP)	int	Using iperf3, at the end of each experiment. Experiment duration is a customizable parameter	Number of retransmissions of TCP session

From a data collection perspective, the setup for the UPF data collected by the monitoring platform at the time writing this deliverable, is depicted in Figure 4-11.

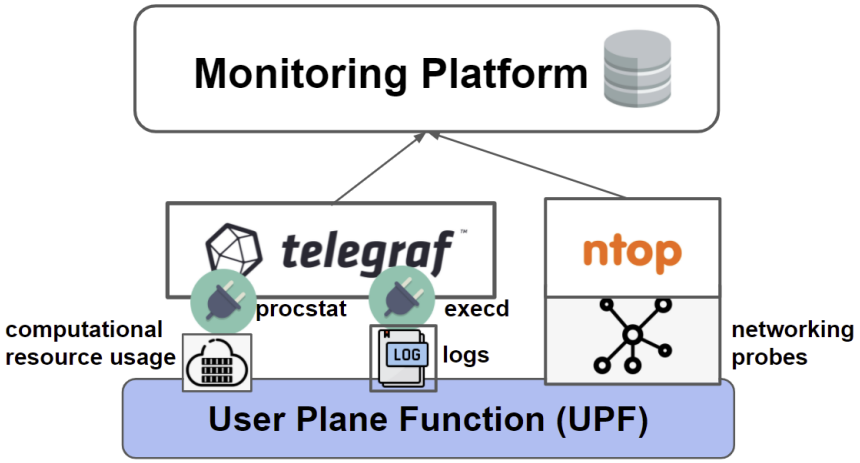

Figure 4-11 - Data source available so far for the monitoring platform

Currently, from the UPF it is possible to collect three types of data:

- The UPF resource usage, in particular related to the CPU, RAM, disks and so on. This information is collected using a Telegraf plugin called *procstat*. With this plugin it s possible to obtain, at given time, metrics like the amount of time that the UPF uses the CPU, the amount of memory that the UPF uses for data, the number of bytes that the UPF has read from disks and has written into disks and so on. In general, it provides a detailed perspective about the resource usage of the UPF in a given time.
- From the UPF logs, information related to the PDU sessions. In this case, a Telegraf plugin called *execd* is properly configured and then used. Currently, the processing of the logs from the UPF is ongoing and

Co-funded by the Horizon 2020 Framework Programme of the European Union

information such as the total number of PDU sessions and PDU sessions per gNB can be extracted from the logs. Also in this case, the information is time-based and is sent to the database each time a new PDU session is established.

- The UPF networking usage information, using *ntop* [32], an open-source traffic monitoring tool, is possible to extract metrics about the network interface the UPF is using. These metrics include a plethora of information: number of bytes and packets transmitted and received to and from a specific interface or from a specific autonomous system, TCP packets dropped, number of active flows and hosts, out of order packets, protocols used for the monitored traffic and so on. Once collected, all these data are sent to the monitoring platform.

The possibility to collect more data related to UPF performances will be further investigated. In this way, the Monitoring Platform can be enriched with all the data needed by the AI/ML platform to extract as much as possible knowledge from the UPF status.

4.4 Network Slice Orchestration Workflows

This subsection describes the high-level workflows of the network slice orchestration. Specifically, the workflow related to the provisioning, termination and re-configuration of an end-to-end network slices are described. The communication mechanisms occurring among the different components is asynchronous for two reasons: first, it is not known at priori the time taken by a specific network domain to either allocate or re-configure or free a certain number of resources, and second, the different components could serve simultaneously multiple requests.

4.4.1 PROVISIONING OF END-TO-END NETWORK SLICE

Figure 4-12 depicts a high-level sequence diagram describing the end-to-end network slice provisioning. The end-to-end network slice provisioning is part of a vertical service provisioning and it is assumed that a vertical service instantiation request is correctly translated into an end-to-end network slice instantiation request. For completeness, it is assumed also that the network domains involved in this particular case are the RAN, Transport Network (TN) Core Network and the vertical (virtual) applications deployment.

Co-funded by the Horizon 2020 Framework Programme of the European Union

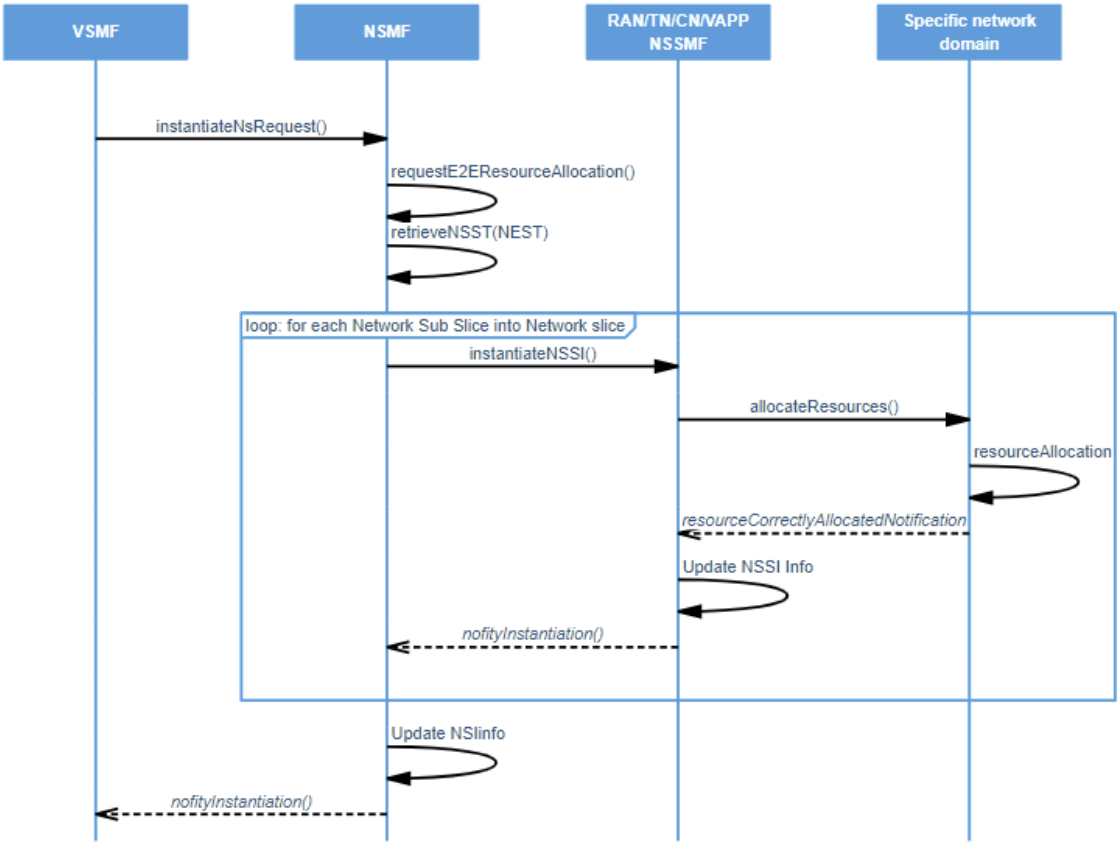

Figure 4-12 - High-level sequence diagram of end-to-end network slice provisioning

The below steps summarize the end-to-end network slice provisioning:

- The NSMF receives the instantiation request, validates it and decomposes the related NST into multiple NSST.
- In this way, the NS LCM Engine knows the NSSMF to make the requests to and the specific parameters to be sent.
- NSMF sends multiple specific requests for network slice subnet instantiation to the specific NSSMFs.
- At this point, each NSSMF translates the instantiation request into domain-specific requests, sending it to the related network domain controller/orchestrator.
- Depending on the domain and the type of resource provisioning, this operation is not synchronous. For this reason, it can be possible that in some cases the request is submitted and eventually a notification is expected. Consequently, each NSSMF saves internally the information about the Network Slice subnet provisioning and notifies the NSMF about the instantiation.
- The NSMF updates the information related to the end-to-end network slice instantiation.
- Finally, the VSMF is notified about the end-to-end network slice provisioning.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4.4.2 TERMINATION OF END-TO-END NETWORK SLICE

Figure 4-13 depicts the high-level workflow of the termination of an end-to-end network slice as part of a vertical service. It is assumed that the tenant wants to terminate the vertical service and the VSMF triggers the workflow termination of the end-to-end network slice.

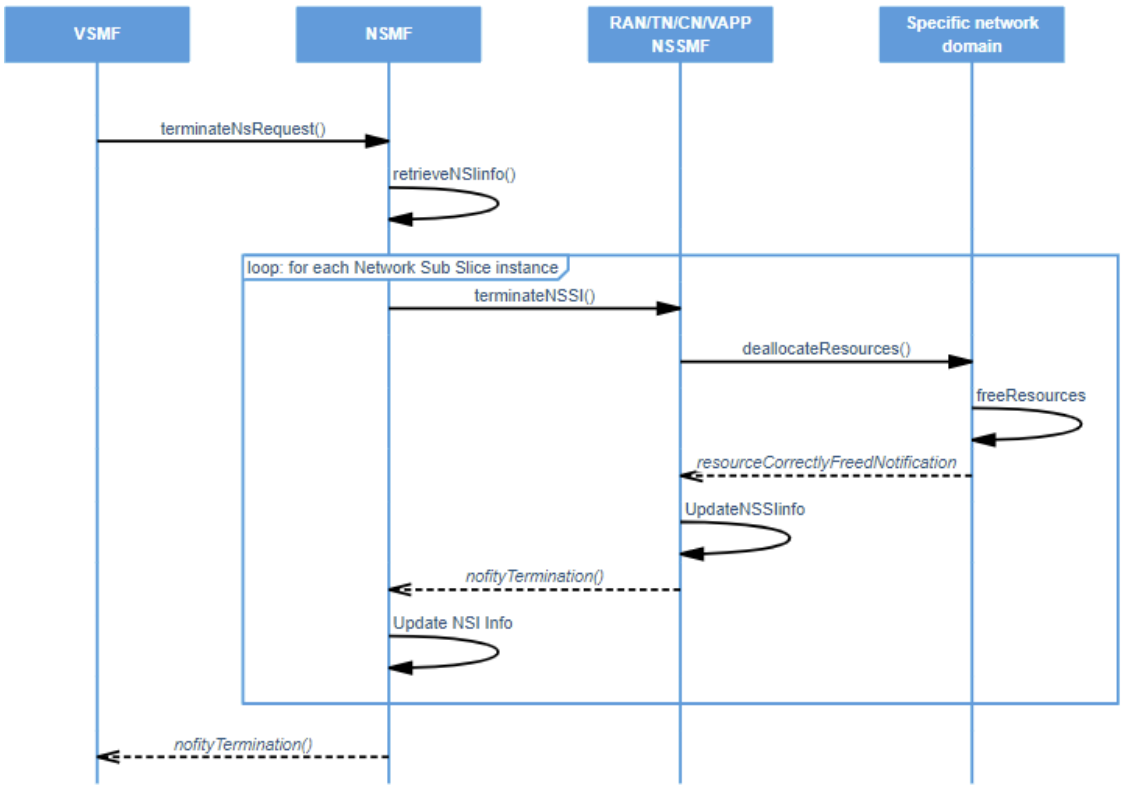

Figure 4-13 - High-level workflow of end-to-end network slice termination

The below steps summarize the end-to-end network slice termination:

- The VSMF requests to terminate the end-to-end network slice towards the NSMF.
- The NSMF receives and checks the request, sending the corresponding termination requests to the NSSMFs where the network slice subnet instances have been previously provisioned.
- Then, the NSSMFs request the related and specific network domain controllers/orchestrators to free or terminate the resources previously allocated.
- Depending on the network domain workflow implementation the resources can be freed synchronously or asynchronously. In the latter case, once these resources have been correctly deallocated, then notifications are sent, the NSSMFs update the NSSI information.
- At this point, the NSMF is notified by the NSSMFs and updates the NSI info.
- Finally, the VSMF is notified about the correct termination of end-to-end network slice termination.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4.4.3 RE-CONFIGURATION OF END-TO-END NETWORK SLICE

Figure 4-14 shows a high-level sequence diagram of an end-to-end network slice re-configuration. In general, a re-configuration consists of scaling up or scaling down the end-to-end network slice resources. In other terms, either new computational and networking resources can be allocated, or eisting ones can be deallocated. For simplicity, it is assumed that the re-configuration, in the case of scaling up operation, does not violate any SLA belonging to the related Tenant and all the involved specific network domains have enough resource to satisfy this request. Moreover, the re-configuration of the end-to-end network slice could be either trigger by the VSMF or by the AI/ML engine as part of the scenario defined in 4.3.

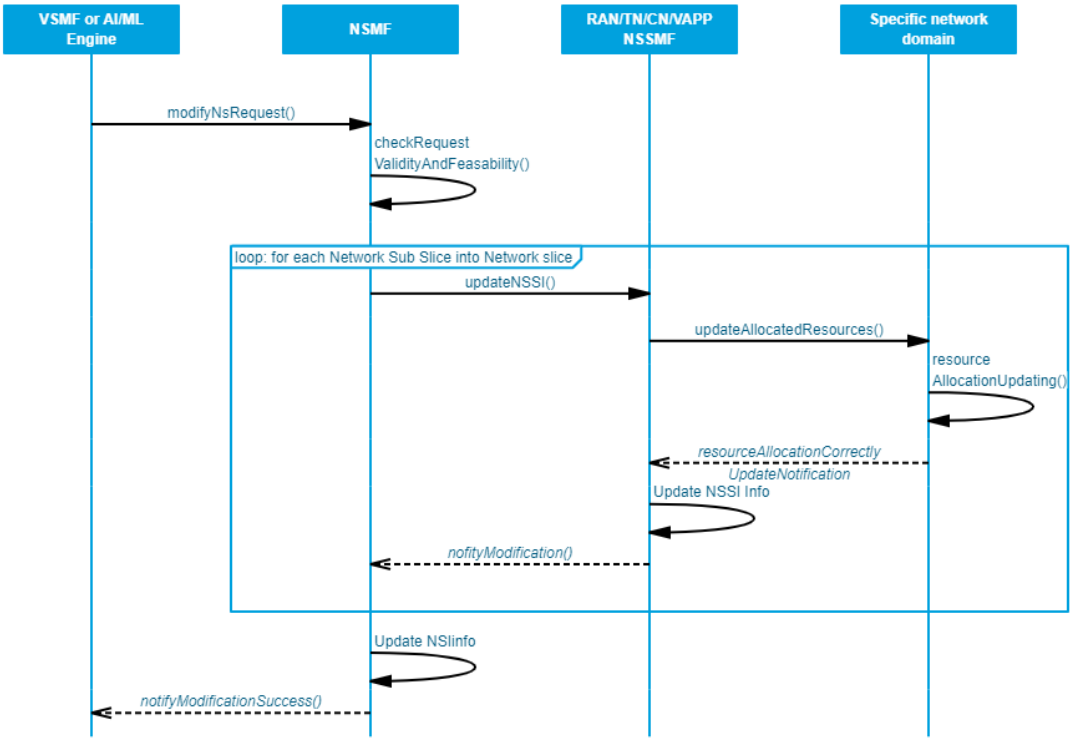

Figure 4-14 - High-level workflow of end-to-end network slice re-configuration

The below steps summarize the end-to-end network slice re-configuration:

- The AI/ML Engine or VSMF requests to re-configure the end-to-end network slice towards the NSMF, specifying what are the new high-level slice capabilities to enforce. In the former case, the request is automatically triggered by the AI/ML internal logic, in the latter case the request is manual through the VSMF user interface.
- The NSMF receives and validates the request, processing the new slice capabilities and identifying the involved domains a related subnet slices. At this point, it sends the corresponding re-configuration requests to the correspondent NSSMFs. Depending on the high-level request, not all NSSMFs could be involved.

Co-funded by the Horizon 2020 Framework Programme of the European Union

- Then, the NSSMFs, after having properly translated the NSMF request, interact with the specific network domain controllers/orchestrators to re-configure properly the resources.
- If a notification mechanism is present at network domain level, after a while, the specific network domain controllers/orchestrators update their allocated resources and send the corresponding notifications. Then, the NSSMFs update the NSSI information.
- At this point, the NSMF is notified by the NSSMFs about the NSSI re-configuration and updates the status of the NSI.
- Finally, the AI/ML engine or VSMF is notified about the correct re-configuration of the end-to-end network slice.

4.5 Interfaces and APIs

In this section the high-level REST APIs of the different components and services part of the end-to-end network orchestration framework are described. In particular, for each layer, a specific subsection containing the REST APIs are available in table format. In particular, for each REST API, the specific functionality offered, the requestor, the HTTP method, the URL, a brief description and the request parameter are provided.

4.5.1 VSMF REST API

In this section the REST APIs exposed by the VSMF are described. Table 4-3 details the REST APIs for Tenant and SLA management. These requests can be performed by an administrator that, after a successful login, can manage the tenants, groups and the related SLAs within the VSMF.Table 4-5 details the REST API of Vertical Service LCM. Some of the described REST APIs are in a working in progress state and their URL could change during future development.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Table 4-3 - REST APIs for tenant and SLA management

Functionality	HTTP method	URL	Description	Request parameters
Group creation	POST	/vs/admin/group/{{group}}	Request to create a new group	Group name in the URL.
Tenant instance creation	POST	/vs/admin/group/{{group}}/tenant	Request to create a new tenant inside a group specified	A group identifier that identifies the tenant's group in the URL. In the body, a JSON object that represents the Tenant information
Tenant(s) information retrieval	GET	/vs/admin/group/{{group}}/tenant	Request to retrieve tenants' information by group name	In the URL, a group name to retrieve all tenants inside a group.
Tenant information removal	DELETE	/vs/admin/group/{{group}}/tenant / {{tenant}}	Request to delete a specific tenant	In the URL, the group name and the username that identifies the tenant to be deleted.
SLA creation	POST	/vs/admin/group/{{group}}/tenant / {{tenant}}/sla	Request to create a new tenant's SLA	In the URL, the group name and the username that identifies the tenant; As payload, a JSON object representing the SLA to be created for a specific tenant.
SLA information retrieval	GET	/vs/admin/group/{{group}}/tenant / {{tenant}}/sla	Request to retrieve a specific SLA	In the URL, the group and the username to retrieve all SLAs of a specific tenant;

Table 4-4 - REST APIs for VSB and VSD management

Functionality (Requestor)	HTTP Method	URL	Description	Request parameters
VSB creation (A)	POST	/portal/catalogue/vsblueprint	Request to create a new VSB	As payload, a JSON object representing the on board VSB request.
VSB(s) information retrieval	GET	/portal/catalogue/vsblueprint/ {{vsb_id}}	Request to retrieve one or all VSBs	In the URL, a VSB identifier that identifies the VSB to be retrieved; None to retrieve all VSBs.

Co-funded by the Horizon 2020 Framework Programme of the European Union

(A or T)				
VSB removal (A)	DELETE	/portal/catalogue/vsblueprint/{{vsb_id}}	Request to delete a VSB	In the URL, a VSB identifier that identifies the VSB to be deleted.
VSD creation (T)	POST	/portal/catalogue/vsdescriptor	Request to create a new VSD	As payload, a JSON object representing the on board VSD request.
VSD(s) information retrieval (A or T)	GET	/portal/catalogue/vsdescriptor / {{vsd_id}}	Request to retrieve a specific VSD or all VSDs	In the URL, a VSD identifier that identifies the specific VSD to be retrieved; None to retrieve all VSDs.
VSD removal (T)	DELETE	/portal/catalogue/vsdescriptor / {{vsd_id}}	Request to delete a specific VSD	In the URL, a VSD identifier that identifies the VSD to be deleted.

Table 4-5 - REST APIs for Vertical Service LCM

Functionality (Requestor)	HTTP method	URL	Description	Request parameters
VSI Instantiation (T)	POST	/vs/basic/vslcm/vs/{{vsi_id}}/instantiate	Request to instantiate a new VSI	As payload, a JSON object representing the VSI Instantiation
VSIs information retrieval (A or T)	GET	/vs/basic/vslcm/vs/{{vsi_id}}	Request to retrieve a specific VSI or all VSIs	In the URL, a VSI Identifier that identifies a specific VSI None to retrieve all VSIs.
VSI modification (T)	PUT	/vs/basic/vslcm/vs/{{vsi_id}}	Request to modify a specific VSI	In the URL, a VSI Identifier that identifies the VSI to be modified and a JSON object representing the VSI modification request.
VSI purging (T)	POST	/vs/basic/vslcm/vs/{{vsi_id}}/purge	Request to purge a specific VSI	In the URL, a VSI Identifier that identifies the VSI record to be deleted.
VSI termination (T)	GET	/vs/basic/vslcm/vs/{{vsi_id}}/terminate	Request to terminate a specific VSI	In the URL, a VSI Identifier that identifies the VSI to be terminated.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4.5.2 NSMF REST API

This section contains all the REST APIs of the Network Slice Management Function.

Table 4-6 describes the NST/GST management and the end-to-end network slice management, respectively. In this case, the requestor is either the VSMF or the administrator. In both cases, a valid authentication towards the NSMF must be performed.

Table 4-7 describes the REST APIs for the LCM management of an end-to-end network slice. These requests can be made either by the VSMF or an external entity like the AI/ML platform engine.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Table 4-6 - REST APIs for NST/GST management

Functionality	HTTP method	URL	Description	Request parameters
NST/GST on boarding	POST	/ns/catalogue/nstemplate	Request to onboard a new NST and/or a GST	As payload, a JSON object representing the NST/GST
N(S)STs/GST(s) retrieval	GET	/ns/basic/nslcm/ns/{nstId}	Request to retrieve one or more NST/GST. Single NST/GST are retrieved specifying its identifier.	In the URL, a NST identifier to retrieve a single NST or GST.
NST/GST Removal	DELETE	/ns/basic/nslcm/nss/{nstId}	Request to retrieve one or more NSST specifying its identifier.	In the URL, a NSST identifier to retrieve a single NST or GST.

Table 4-7 - REST APIs for end-to-end network slice management

Functionality	HTTP Method	URL	Description	Parameters
NSI entry creation	POST	/ns/basic/nslcm/ns	Request to create a NSI identifier	As payload, a JSON containing the NST associated with, the NSI name and a description.
NSIs information retrieval	GET	/ns/basic/nslcm/ns(s)/{nstId}	Request to retrieve information about one or more NSIs	In the URL, a NSI identifier that identifies the specific NSI that have to be retrieved; None to retrieve all NSIs.
NSI Instantiation	PUT	/ns/basic/nslcm/ns/{nsiId}/action/ instantiate	Request to instantiate a new NSI	In the URL, a NSI identifier and as payload a JSON object representing the instantiation request.
NSI modification	PUT	/ns/basic/nslcm/ns/{nsiId}/action/ modify	Request to modify an NSI	In the URL, a NSI identifier and as payload and a JSON object representing the modification request.
NSI termination	POST	/ns/basic/nslcm/ns/{nsiId}/action/ configure	Request to configure an NSI	In the URL, a NSI identifier and as payload and a JSON object representing the configuration request.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4.5.3 NSSMF REST API

This section contains all the REST APIs of the Network Slice Subnet Management Function. Table 4-8 describes the REST APIs for the network slice subnet management. The request body of each REST API could be different depending on the specific implementation of the NSSMF.

Table 4-8 - REST APIs for end-to-end network slice subnet management

Functionality	HTTP method	URL	Description	Request parameter
Create NSSI	POST	/nss/createnss	Request to create a new NSSI identifier	No request parameter
Instantiate Network Slice Subnet	PUT	/nss/{nsiId}/action/instantiate	Request to instantiate a network slice subnet	In the URL, the identifier of the network slice subnet and a customized payload depending on the NSSMF logic.
Modify an instantiated Network Slice Subnet		/nss/{nsiId}/action/modify	Request to modify a network slice subnet	
Terminate an instantiated Network Slice Subnet		/nss/{nsiId}/action/terminate	Request to terminate a network slice subnet	

Co-funded by the Horizon 2020 Framework Programme of the European Union

5 Preliminary Prototype and Early Integration

This chapter provides a brief overview of the preliminary prototype of the end-to-end network orchestration framework. It also describes the initial integration with the various iNGENIOUS Network Layer technologies and related network domain controllers/orchestrators. In detail, this refers to the software prototype and related integration activities with the 5G Core, Flexible PHY/MAC and O-RAN.

5.1 Software prototype description

This section describes the software prototype of the end-to-end network orchestration framework. In particular, for each component described in Section 4.2.1 is available a technical description.

The source code of the software prototype components developed for the end-to-end network orchestration framework, which is an extension of the Nextworks opensource Slicer [23], makes use of the Spring Boot framework [33]. This framework allows to easily manage multiple services, each of them specific for a functionality, maintaining a high degree of modularity. To maintain the data persistence, postgreSQL [34] database has been used. Starting from the available information model implemented as Java classes, the tables within the database are automatically created.

Some of the internal components are in a preliminary release stage, while other components are in the development phase and will be delivered with the final version of the end-to-end network slice orchestration framework, planned to be described in deliverable D4.5. All the software components described in this section are currently available in an internal Gitlab repository. However, it is under evaluation to release either part or all of the software as open-source in the main Nextworks Slicer repository.

5.1.1 VSMF SOFTWARE

The implementation of the VSMF is realized using the Object-Oriented Programming Language Java. It consists of different Java libraries and other tools for realizing the interactions among them.

Figure 5-1 depicts how each component described in Section 4.2.1.2 is built in terms of internal Java modules. This diagram maps the high-level software architecture depicted in Figure 4-5 with the actual implementation. For clarity, each VSMF component has a different shade of blue, while the internal libraries are depicted as white blocks.

Co-funded by the Horizon 2020 Framework Programme of the European Union

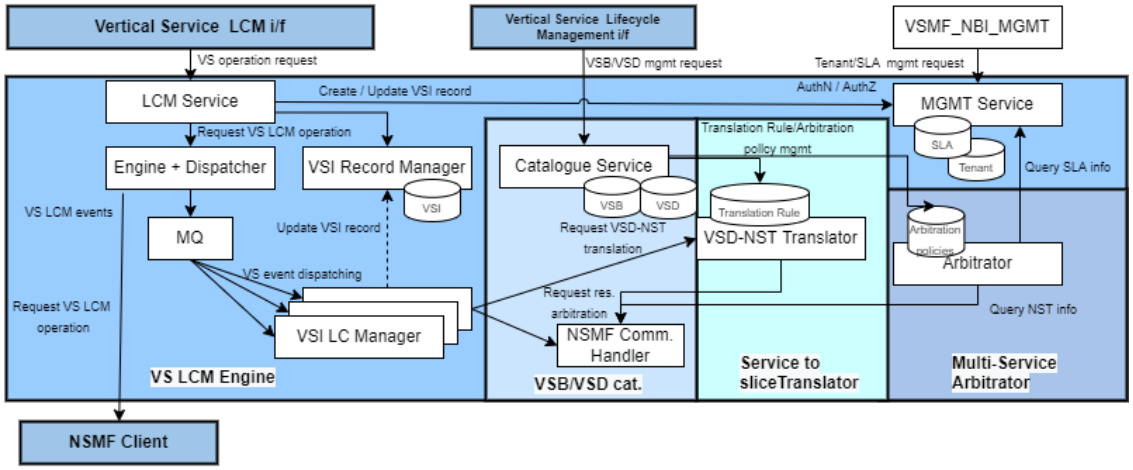

Figure 5-1 - VSMF implementation diagram

Table 5-1 describes the software technical description of the VSMF components. According to its internal complexity, the component is either an independent java library or a software module embedded within the source code of the VSMF itself.

Table 5-1 - Technical description of VSMF components

Software component	Technical description
Vertical Service LCM i/f	REST APIs for the LCM of the Vertical Slices. This component is mapped one-to-one with its design.
VS LCM Engine	It provides the whole LCM capabilities for the vertical services using the implementation of a state finite machine logic for representing the status of vertical services. In consists of different Java modules: • The Identity management (MGMT Service and VSMF_NBI_MGMT blocks), consisting of Java classes for representing the information model about the Tenant and SLA, the implementation for creating, reading, updating and deleting the tenant and SLA information into the database. It contains also the REST API implementation (VSMF_NBI_MGMT block) for exposing the information related to group, tenants and SLAs. • VSI record manager for managing the information of the vertical service instances. • The LCM Service where the vertical service instances are validated against the related Tenant and SLAs. • Engine + dispatcher where the core logic of the VSMF is implemented. The requests are dispatched through an MQ Rabbit Queue (MQ block) to the VSI LCM instance(s).

Co-funded by the Horizon 2020 Framework Programme of the European Union

	• VSI LCM Manager instance to manage the lifecycle of the single Vertical Slice. It interacts with the Service to slice translator and the VSB/VSD catalogue.
VSB & VSD catalogue	It consists of the Java classes for representing the VSB and VSD information model, the implementation of the logic to create, read, updated and read the VSB and VSD to and from the database. It contains also the REST API layer to expose the VSB and VSD information.
Service-to-slice translator	It is a Java module, consisting of the implementation of the translation of the vertical service instantiation request into end-to-end network slice instantiation request. In particular, following the attributes of the VSD, it translates them in one or more available NSTs.

5.1.2 NSMF SOFTWARE

Figure 5-2 depicts the implementation diagram of the NSMF and the main internal interactions. This diagram contains the preliminary implementation of the NSMF components briefly described in Section 4.2.1.3 and whose high-level architectural is depicted in Figure 4-6. Similar to the VSMF implementation, the Java programming Language has been used for developing all the modules and libraries within the NSMF. Some components are not depicted because they are still in development phase.

For clarity, the different components are in shades of blue, while the internal Java libraries and modules are the ones in white color.

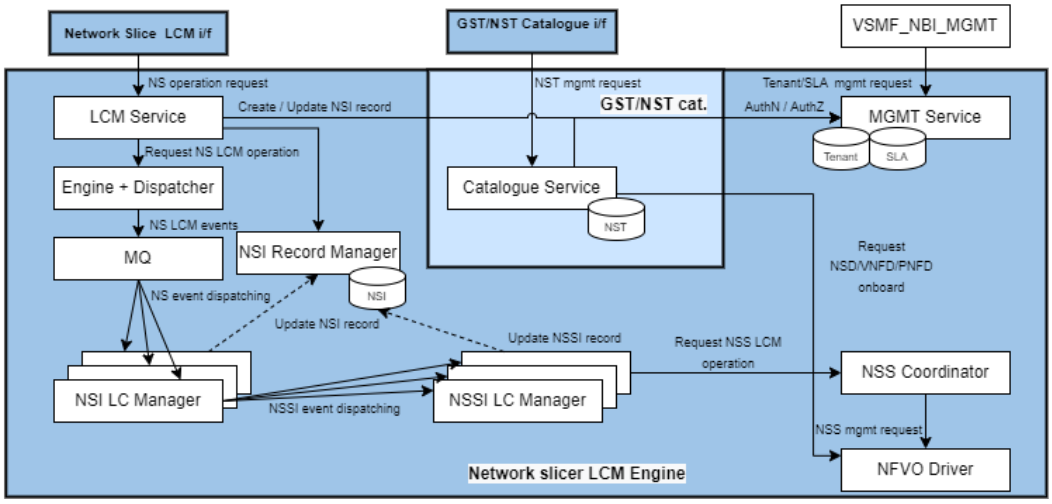

Figure 5-2 - Implementation diagram of NSMF

Table 5-2 contains the software technical details of the NSMF components. Some components are standalone Java library that can be reused, while other are software modules developed within the NSMF Spring boot Application.

Table 5-2 - Technical description of NSMF Components

Co-funded by the Horizon 2020 Framework Programme of the European Union

Software component	Technical description
Network Slice LCM i/f	Its implementation consists of the REST APIs for providing network slicing LCM functionalities. It is mapped one-to-one with the related designed component.
Network slice LCM Engine	It provides the whole LCM of the end-to-end network slices using the implementation of a state finite machine logic for representing the status of end-to-end slices. Moreover, to guarantee the management of several requests and notifications asynchronously, it exploits the RabbitMQ bus message exchange. This service exposes its functionalities using REST interfaces. In particular, the Network slice LCM Engine has the following modules: • LCM service, where the request is validated and checked against the identity management. The network slice information is managed through the NSI Record Manager. • The Engine + dispatcher, where the core logic of NSMF is implemented and the requests are dispatched towards the NSI LCM instances. • NSI LCM, that manages the LCM of a single end-to-end network slices and the related network slice subnet instances, through the specific drivers (NSS Coordinator and NFVO driver blocks).
GST/NST catalog	It consists of the Java classes representing the GST, NST and NSST data model and the implementation to create, read, update and delete GST(s), NST(s) and NSST(s) from and to database. Moreover, a translation mechanism that maps the attributes of GST into the NST attributes and vice versa is available too.
Slice function placement	It breaks down the NST, following its attributes, into multiple NSSTs for the end-to-end network slice instantiation. In particular, it specifies where the VNFs, vertical service virtual applications representing one or more NSSTs should be deployed. The development of this component is on-going, for this reason is not depicted in the diagram.

5.1.3 NSSMF SOFTWARE

Table 5-3 describes the software and technical details of the NSSMF components. This technical description refers to a generic NSSMF, which implementation can be specialized for realizing the specific NSSMF functionalities.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Table 5-3 - Technical description of (O-RAN) NSSMF components

Software component	Technical description
Northbound interface	It has been implemented using the REST API provided by Spring Boot framework. Except in special cases, these REST APIs are the same for all NSSMFs regardless their implementation. However, the payload sent to this interface is specific for the NSSMF implementation.
NSSMF Service	Its implementation consists of checking the payload content of the request. If valid, it is dispatched to the event bus pushing it as event. In the case of the O-RAN NSSMF implementation it checks if all the mandatory field are available and valid.
Bus Event	It has been used the Guava Event bus [35] to manage and dispatch simultaneously multiple and concurrent requests, allowing not blocking requests from the NBI.
NSSMF Handler	It represents the engine of the NSSMF where the whole logic is implemented. It subscribes to all the events of the Guava Event Bus and translates the requests for the specific network domain. In the O-RAN case, the slice profiles within the NSSI requests are translated into A1 policy requests. It performs operations towards the postgreSQL DB and uses the specific network domain driver to send the translated requests.

More details about the implementation of the O-RAN NSSMF can be found in Section 5.4.

5.2 Integration with 5G Core

In general, a 5G network contains the Radio network segment for connecting the UEs and the 5G Core for the control and user planes. From a deployment perspective, for providing the 5G connectivity to the UEs, all the 5G network components and functions must be properly setup and configured, exploiting the provided hardware and software infrastructure. When network slices are considered and provisioned, end-to end network slice resources must be properly managed and allocated at the different segment of the networks. This task is usually performed by an end-to-end network slice orchestrator that interacts with the different segments of the network (RAN, 5G Core and so on).

To this end, the integration between the end-to-end network slice orchestration framework presented in this document and the 5G Core has been started. At the current status, it is possible to deploy and configure a monolithic instance of 5G Core as part of an NFV Network Service through the ETSI OSM Web GUI, as already described in deliverable D4.3 [25] and shown in Figure 5-3.

Co-funded by the Horizon 2020 Framework Programme of the European Union

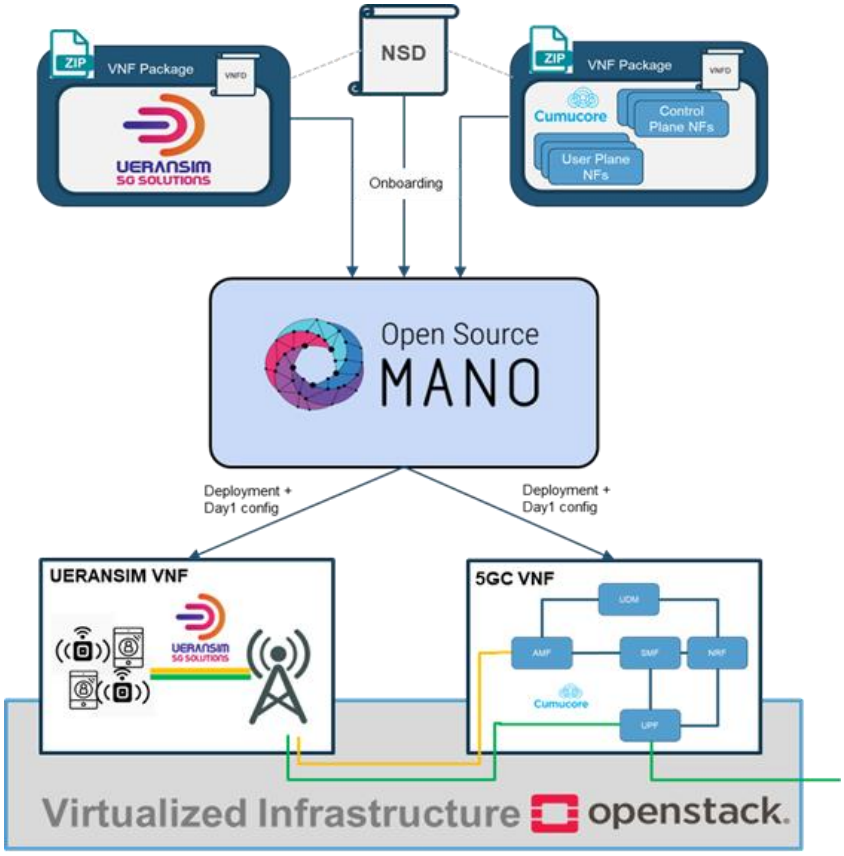

Figure 5-3 - High-level workflow of manual deployment of a 5G network

A NSD containing the VNF Descriptor packages of UERANSIM and 5G Core is onboarded on ETSI OSM and then deployed on an OpenStack [36] virtualized infrastructure. After having deployed the 5G Core and UEARANSIM instances, the day-0 and day-1 configurations are properly executed, using cloud-init [37] and the juju [38] open-source tool, respectively. Day-0 configurations of UERANSIM and 5G Core briefly consists of setting up the SSH credentials on both machines and using the assigned IP address for configuring the 5G Core Network Function and gNB, respectively.

On the other hand, the day-1 configurations of 5G Core instance consist of the following three steps:

1. Get the assigned local IP address of the 5G Core in order to make the emulated gNB interconnect to it, in particular the AMF Network Function. This step is used by a juju relation.

2. Start in the correct order the 5G Core Network Functions.

Similarly, the day-1 configuration of UERANSIM instance consists of properly configuring the emulated gNB, setting the IP address of the 5G Core AMF. This IP is got from the step 1 of day-1 configuration of 5G Core using the juju relations available into the Network Service Descriptor. At this point, the emulated gNB is ready to be connected to the 5G Core instance as well as the UEs registered in the 5G Core itself.

Co-funded by the Horizon 2020 Framework Programme of the European Union

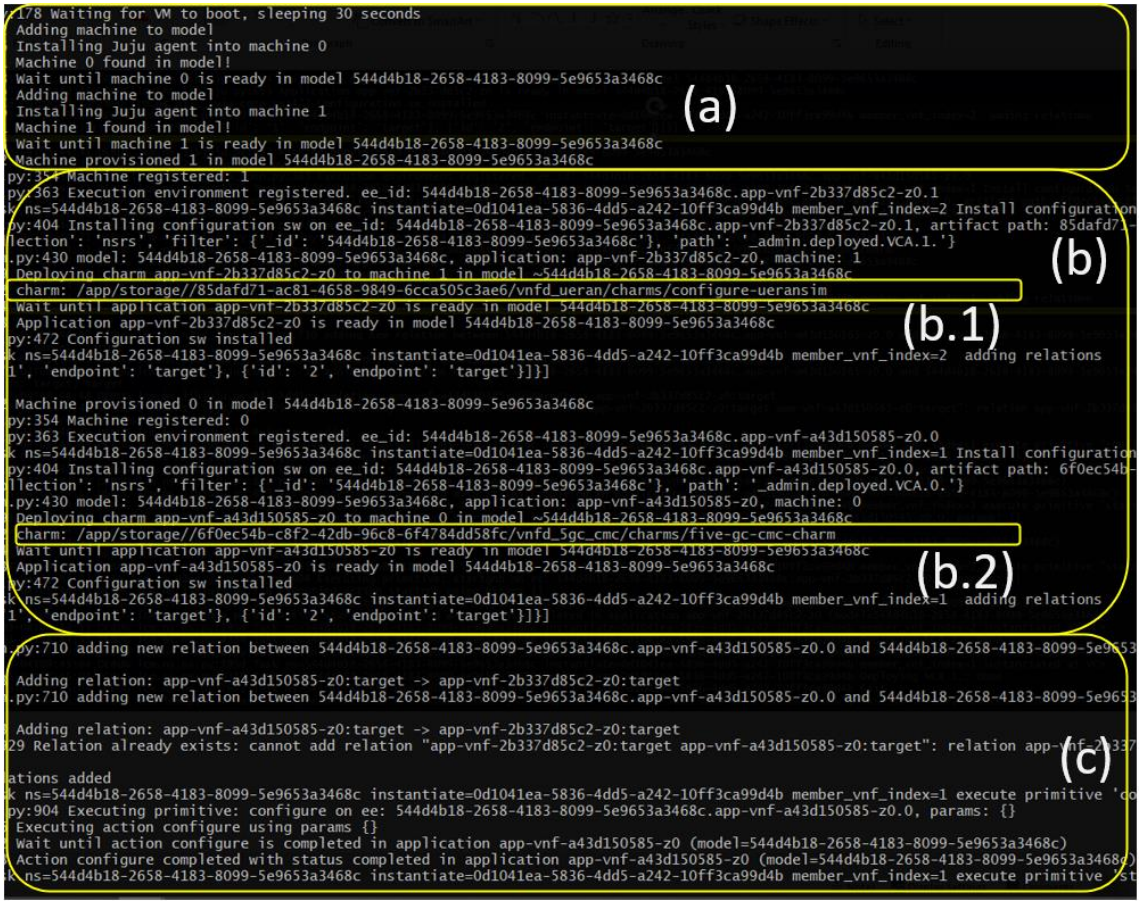

Figure 5-4 - Logs of day-1 configuration stages

Additionally, Figure 5-4 depicts the different stages day-1 configurations are composed of. Specifically, “machine 0” is associated with UERANSIM instance, while “” is associated with 5G Core instance. Both day-1 configurations are associated with the so-called juju models. The stages of day-1 configuration are the following:

- The first step (Figure 5-4 (a)) consists of waiting the machines to be up and running on OpenStack. Using a polling mechanism, the juju model checks whether the machines are ready to be configured or not.
- Once machines are ready, the day-1 configuration can be executed in both machines as depicted in Figure 5-4 (b). Specifically, the related Python scripts are executed in the machine 0 and machine 1 as shown in Figure 5-4 (b.1) and Figure 5-4 (b.2), respectively.
- Finally, the juju-relation is executed (Figure 5-4 (c)) for making the 5G Core reachable to the UERASIM instance.

As an example, Figure 5-5 shows a snippet of the Python script used in the day-1 configuration of the 5G Core instance. In particular, the *on_configure_action* function is used for debugging purposes creating a file, while *_on_start5gc_action* function is used for starting the NFs in the following order: NRF, UPF, SMF and AMF. Finally, the web GUI is started too.

Co-funded by the Horizon 2020 Framework Programme of the European Union

```
def _on_configure_action(self, event):
    ip = "127.0.0.1"
    if("ip" in event.params):
        ip = event.params['ip']
    cmd = "touch /home/ubuntu/charm_on_configure_TODO.out"
    self.execute_command(cmd)

def _on_start5gc_action(self, event):
    sample_par = "a value"
    if("sample_par" in event.params):
        sample_par = event.params['sample_par']
    cmd = "systemctl start nrf.service"
    self.execute_command(cmd)
    time.sleep(2)

    cmd = "systemctl start upf.service"
    self.execute_command(cmd)
    time.sleep(2)

    cmd = "systemctl start smf.service"
    self.execute_command(cmd)
    time.sleep(2)

    cmd = "systemctl start amf.service"
    self.execute_command(cmd)
    time.sleep(2)

    cmd = "systemctl start epcgui.service"
    self.execute_command(cmd)
    time.sleep(2)

    cmd = "touch /home/ubuntu/start5GC.out"
    self.execute_command(cmd)
```

Figure 5-5 - Snippet of Python script of day-1 configuration of 5G Core instance

Currently, activities to automatically deploy the new Cumucore 5G Core version available in containerized version are ongoing. In particular, Figure 5-6 depicts the Control Plane Network Functions available as Docker containers and the Data Plane Network Functions available as separated but related network services. From a deployment perspective, Control Plane NFs and User Plane NFs can be deployed as either single or multiple VNF instances belonging to the same Network Service. Moreover, the Core NSSMF can manage the slices available in the User Plane NFs using the corresponding REST APIs offered by the UPF.

Co-funded by the Horizon 2020
Framework Programme of the European Union

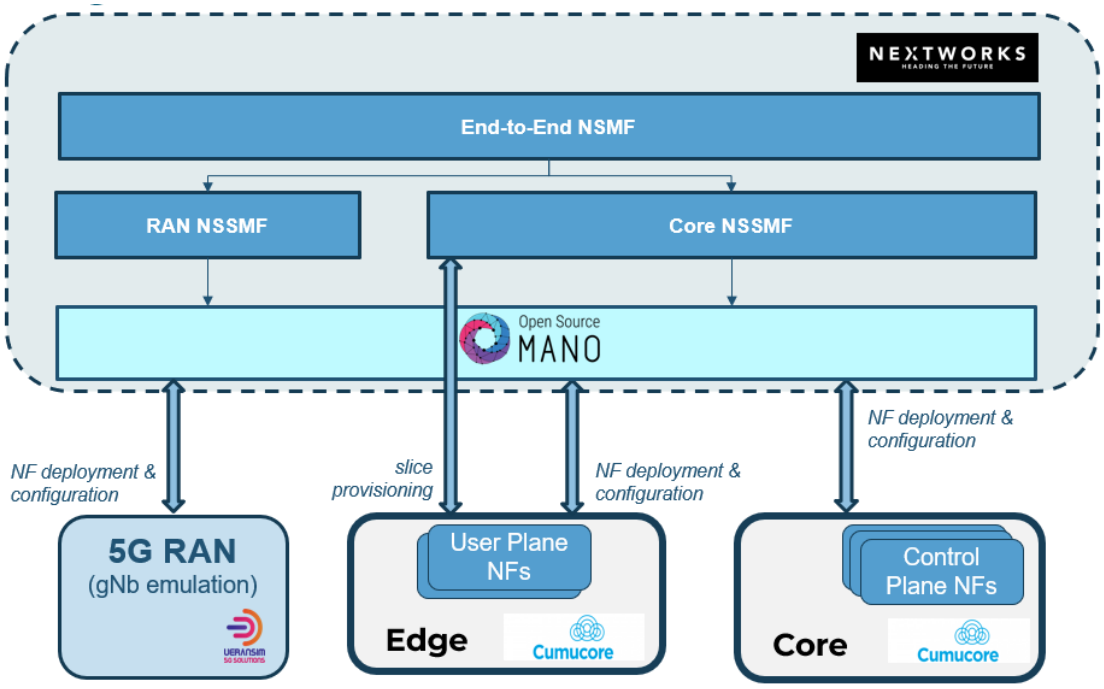

Figure 5-6 - Edge-core deployment of 5G Core through the end-to-end Network Slicer

5.3 Integration with Flexible PHY/MAC

As part of the end-to-end network slice orchestration work, integration activities with the Flexible PHY/MAC have been started. These activities are still in an initial phase, mostly focused on the identification of the integration approach as depicted in Figure 5-7.

The final aim of this integration is to make the Flexible PHY/MAC solution 5G compatible and thus enable 5G connectivity to multiple SDR platforms representing the UEs.

From an orchestration point of view, the end-to-end network slice orchestrator deploys a 5G Core instance, an UERANSIM instance and configures properly the Flexible PHY/MAC enabled Base Station (BS). In details:

- The 5G Core and the related NFs are deployed and configured by the Core NSSMF through ETSI OSM. Moreover, slice configurations can be performed by the Core NSSMF itself.
- The RAN NSSMF not only deploys the UERANSIM instance through ETSI OSM, but it also it connects to the Flexible PHY/MAC BS, where communication resources can be allocated depending on specific application requirements. Indeed, Flexible PHY/MAC can expose dedicated APIs to allocate radio resources according to application requirements.

Co-funded by the Horizon 2020 Framework Programme of the European Union

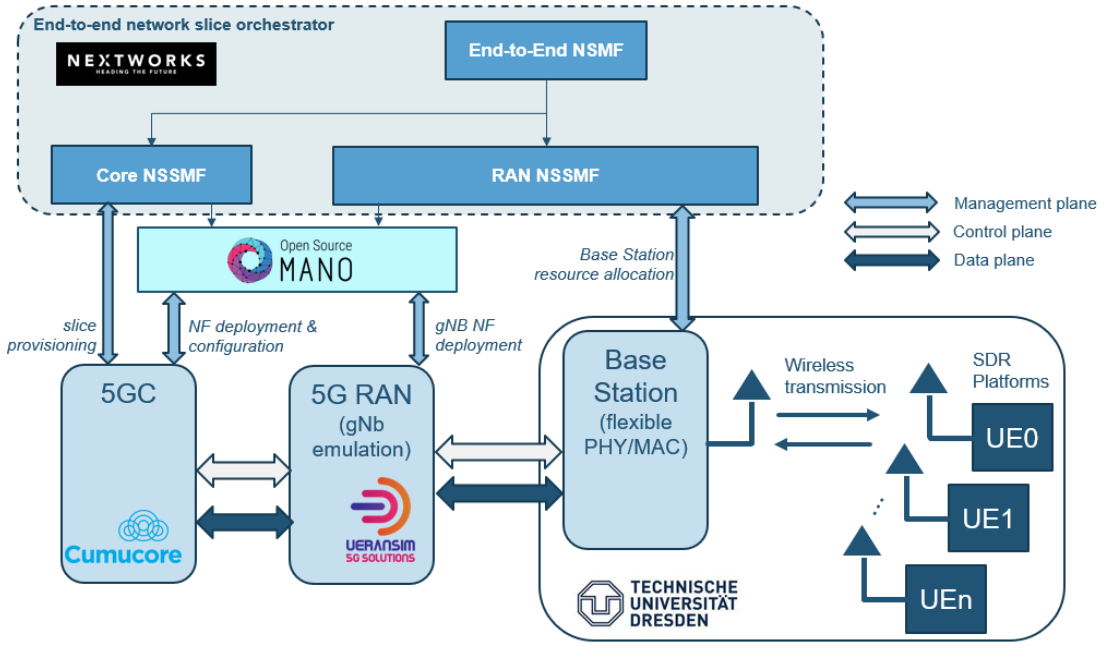

Figure 5-7 - Integration diagram between end-to-end network slice orchestrator and flexible PHY/MAC

Some initial experiments have been conducted in order to check the viability of the integration between the UERANSIM, the 5G Core and the BS. The setup is depicted in Figure 5-8. The data communication flows from the UEs to the sinks consist of the following steps:

1. Starting from the right of Figure 5-8, the UEs send the data in the uplink direction using the wireless transmission.
2. Then, the *Base Station* forwards the user plane data to the *UE Emulator*.
3. In the *UE Emulator*, a Python script forwards the data to the corresponding tunnel interface created by the UERANSIM.
4. At this point, the UERANSIM sends the data to the corresponding data sink, i.e., *Sink UE0* or *Sink UE1*, via the 5G Core. At this initial step, the Open5GS [39] 5GCore has been used to connect the users.

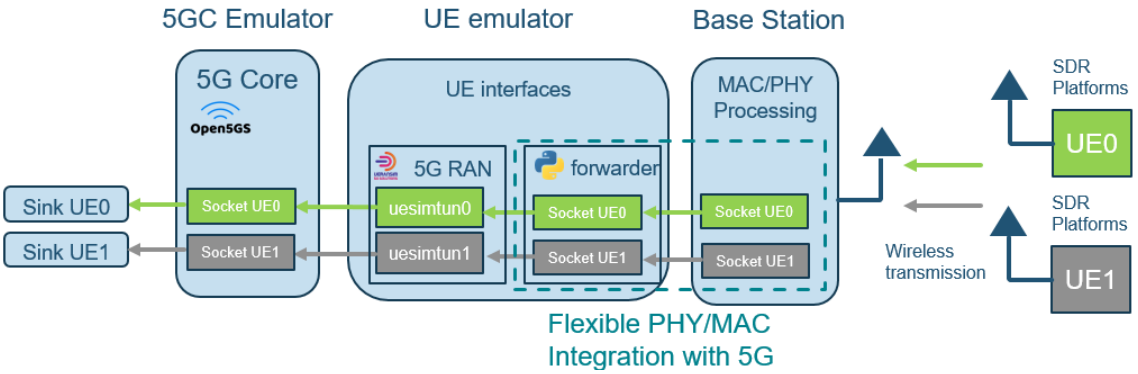

Figure 5-8 - Flexible PHY/MAC integration with UERANSIM

At the current state, the 5G Core and UERANSIM instances can be configured and deployed using ETSI SOM and through a customized script, while the SDR

Co-funded by the Horizon 2020 Framework Programme of the European Union

platforms can connect to the simulated gNB using the physical base station. In this way, the UEs can send data using the deployed 5G network. On the orchestration side, control plane and user plane are provided and developments for automated deployment of the 5G Core and UERANSIM instances and configuration of the Flexible PHY/MAC enabled Base Station are ongoing.

5.4 Integration with O-RAN

An initial integration work between the end-to-end network slice orchestration framework and O-RAN has been carried out and described in the D4.2 Deliverable [14]. In this context, an early prototype of the O-RAN NSSMF has been implemented, tested and integrated with a Near RT RIC. Additionally, slice profiles available in the network slice subnet instances are translated into A1 policies within the Near RT RIC itself. Specifically, guaranteed flow bit rate, priority level and packet delay budget available into the A1 policy QoS objective can be configured.

The development and integration activities have been continued after D4.2, leading to an enhancement of the early prototype of the O-RAN NSSMF whose high-level architecture is depicted in Figure 5-9. These enhancements consist of supporting multiple network sub slice instantiation requests asynchronously towards the Near RT RIC of O-RAN. In detail, a specialized handler receives, processes and dispatches the specific sub slice instantiation requests without blocking the whole NSSMF O-RAN. Moreover, as part of the improvement of the NSSMF O-RAN, the information about the network slice subnet instance, the A1 policy and their mapping are persistently stored within a database.

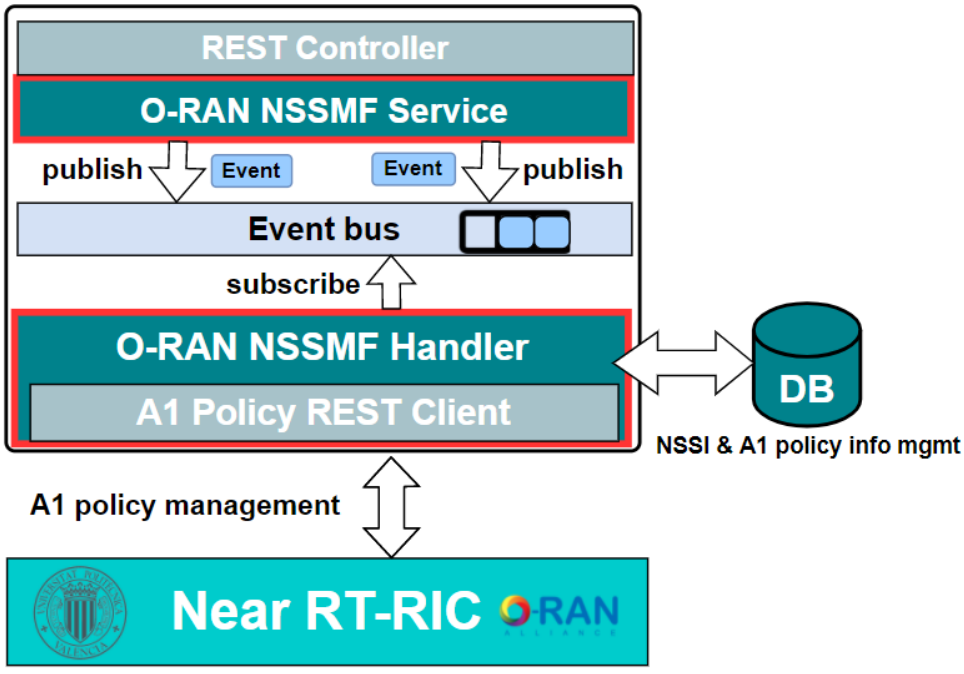

Figure 5-9 - High-level architecture of O-RAN NSSMF

Co-funded by the Horizon 2020 Framework Programme of the European Union

Figure 5-10 depicts the workflow of a network slice subnet instance creation using the enhanced NSSMF implementation described above. The workflow is summarized by the following steps:

- At the start of the O-RAN NSSMF, a subscription to all requests and all responses is made by the O-RAN NSSMF service and O-RAN NSSMF Handler, respectively. In this way, the former can receive and manage all the requests, the latter all the responses.
- When a request of instantiating a RAN network slice subnet instance is received, it is managed by the NSSMF Service, publishing it on the Event bus.
- The NSSMF Handler takes the request as notification and for each slice profile available, it translates into an A1 policy request.
- The RAN NSSMF Handler through the O-RAN A1 Client sends the requests to create the A1 policy.
- Once the A1 policies are created, the responses are notified back to the original requestor.

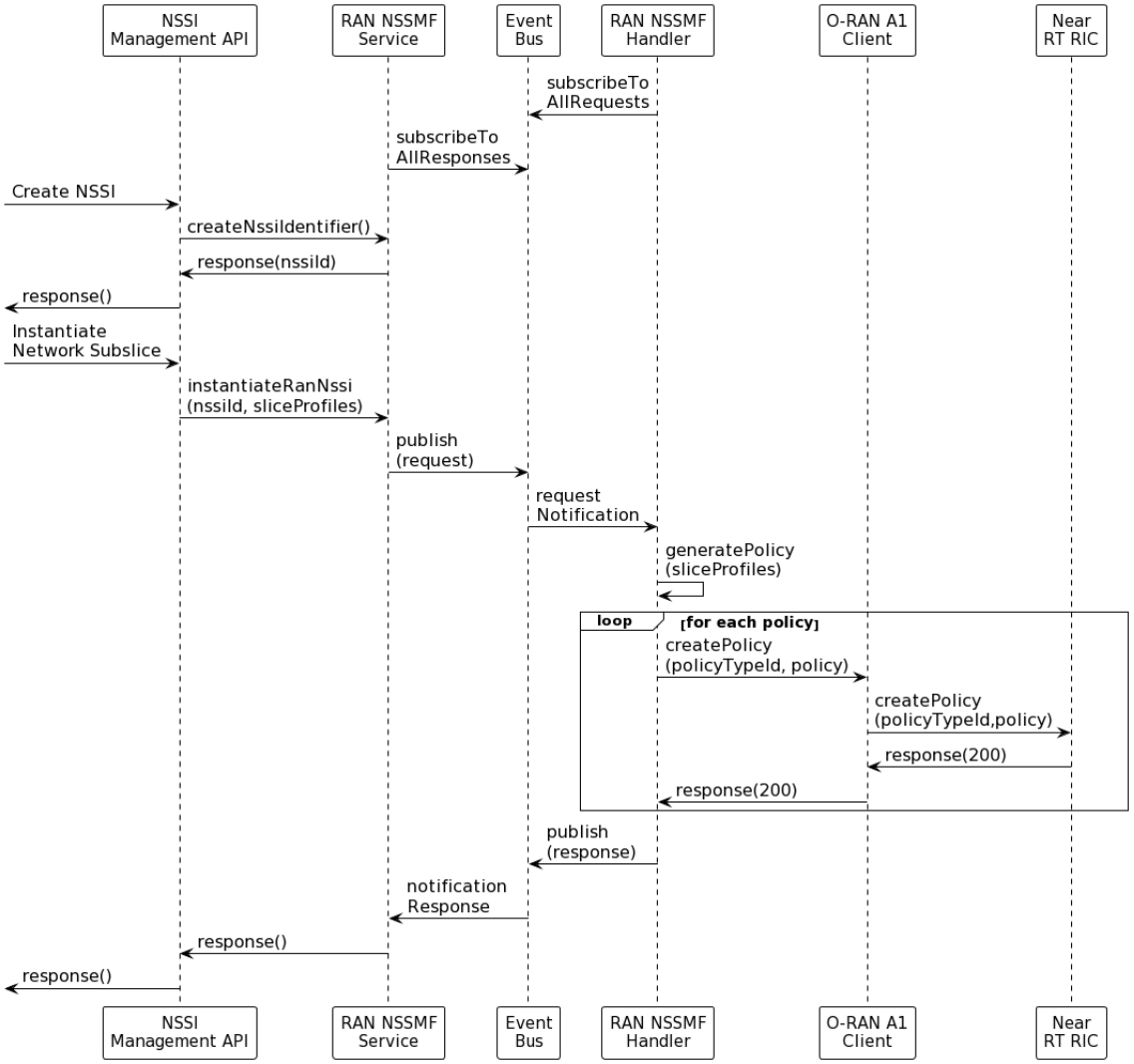

Figure 5-10 - Sequence diagram of network slice subnet instance

Co-funded by the Horizon 2020 Framework Programme of the European Union

From an implementation perspective, Figure 5-11 shows the tree structure of the source code of the O-RAN implementation.

Figure 5-11 - Tree structure of source code of O-RAN NSSMF

In particular, it mainly consists of three libraries:

- **RecordIM**, which contains all the information model (Java classes) that represents the data model for O-RAN: the Java classes of A1 policy type JSON schema and for the specific NSSI.
- **RecordService**: which contains the logic for managing the information model into postgresDB defined into *RecordIM*.
- **O-RAN Handler**: with the support of the *RecordService*, implements the internal logic of the O-RAN NSSMF, managing and translating the slice profiles into A1 policies and sending them to the Near RT RIC.

These libraries are included within a portable Spring Boot Application for exposing through REST APIs the functionality of the O-RAN NSSMF.

Co-funded by the Horizon 2020
Framework Programme of the European Union

6 Relation to UCs and Main Innovations

iNGENIOUS implements the end-to-end network slice orchestration framework described in this document and deploys it in real environments for demonstrating specific project use cases (UCs) in different industrial IoT and supply chain scenarios. This section is dedicated to the identification of the main innovations brought by the end-to-end network slice orchestration, and how they are mapped and used in the relevant iNGENIOUS UCs.

6.1 End-to-end network slice orchestration innovations

The end-to-end network slice orchestration framework described in the previous sections brings a set of technical innovations in the way industrial IoT and supply chain services and network slices are managed. While an initial list of innovations was already presented in deliverable D4.1 [15], together with their mapping to the various iNGENIOUS UCs, few updates have occurred according to the evolution of the technical activities in WP4 and WP6 mostly.

In particular, the end-to-end network slice orchestration functionalities are integrated and validated mostly in the context of the Automated Robots with Heterogeneous Networks UC, with plans of demonstrations in the ASTI factory, as well as in the UPV and TUD testbeds. However, for the Situational Understanding and Predictive Models in Smart Logistics Scenarios and Supply Chain Ecosystem Integration UCs it is still planned to validate some of the key functionalities at least in a partially emulated environment. On the other hand, no specific activities are planned for the other UCs in relation with the end-to-end network slice orchestration framework.

The Table 6-1 below shows the updated list of innovations brought by the end-to-ed network slice orchestration (with respect to D4.1), and for each of them a mapping with the relevant iNGENIOUS UCs in terms of expected innovation maturity. For this mapping, two options are shown:

- innovation concept, for those innovative functionalities relevant to the UCs but that will not be demonstrated in a real environment (e.g., they could be implemented but showcased in emulated scenarios only),
- innovation demonstrated, for those functionalities that will be implemented and showcased as part of the UC in a real environment.

Table 6-1 - Network slice orchestration functionalities mapped to UCs

Functionality	Factory UC	Port Entrance UC	DVL UC
End-to-end network slice provisioning	innovation demo	innovation concept	innovation concept
Orchestration and slicing of Flexible RAN	innovation demo		

Co-funded by the Horizon 2020 Framework Programme of the European Union

Orchestration and slicing of 5G RAN	innovation demo		
Orchestration and data collection from 5G Core	innovation demo		
Orchestration of applications and NFs at edge/MEC	innovation demo		
AI-assisted (network-data based) slice optimization	innovation demo		
AI-assisted (IoT/DVL-data based) slice optimization		innovation concept	innovation concept

6.2 Relation to UCs

This section provides a brief description for each end-to-end network slice orchestration functionality, highlighting the innovative aspects and their maturity for each iNGENIOUS UC, with a mapping to the relevant orchestration components involved.

6.2.1 END-TO-END NETWORK SLICE PROVISIONING

The end-to-end network slice provisioning innovation refers to the capability of automated creation of network slices and the deployment of the required NFs and virtual applications to support industrial IoT and supply chain services. Specifically, this is a key basic innovation, as it refers to the coordinated orchestration of different technological domains, including RAN, edge and core.

This innovation will be fully demonstrated as part of the Automated Robots with Heterogeneous Networks UC. Different incremental demonstrations are planned to be carried out to validate the VSMF and NSMF operation in orchestrating and managing the various subnet slices required for the 5G Core, Flexible PHY-MAC and 5G-RAN. The demonstration activities will be carried out in the different UC sites and testbeds (i.e., ASTI factory, UPV testbed and TUD testbed).

In addition, the innovation concept will be also validated in emulated environments (setup in the Nextworks lab) in the context of the Situational Understanding and Predictive Models in Smart Logistics Scenarios and Supply Chain Ecosystem Integration UCs.

6.2.2 ORCHESTRATION AND SLICING OF FLEXIBLE RAN

The orchestration and slicing of flexible RAN refer to the capability of the end-to-end network slice orchestration framework to manage network resources when the RAN includes flexible PHY-MAC for 5G-NR functionalities. Specifically, this innovation is mapped to the integration and validation of the specific NSSMF component developed for the Flexible PHY-MAC and aiming

Co-funded by the Horizon 2020 Framework Programme of the European Union

at managing and provisioning network subnet slices that combine the PHY-MAC processing with a legacy 5G-RAN and 5G Core, as shown in Section 5.3.

This innovation will be demonstrated as part of the Automated Robots with Heterogeneous Networks UC, and the related activities will be carried out in the TUD testbed.

6.2.3 ORCHESTRATION AND SLICING OF 5G-RAN

The orchestration and slicing of 5G-RAN refer to the capability of the end-to-end network slice orchestration framework to configure and manage network slice subnets resources in a standard 5G-RAN.

This innovation also includes the integration with O-RAN Near-RT RICs for provisioning QoS policies. Here, network slice subnets are mapped with O-RAN A1 policies through a dedicated O-RAN NSSMF, as exposed in Section 5.4.

The first part of this innovation is planned to be demonstrated as part of the Automated Robots with Heterogeneous Networks UC, and the related activities will be carried out in the ASTI factory and UPV testbed. For this a dedicated 5G-RAN NSSMF will be developed to interact with the specific gNBs that will be used for the UC. For the second part of the innovation, the validation and demonstration of the O-RAN NSSMF is planned to be carried out in emulated environments (either in the UPV or Nextworks testbeds).

6.2.4 ORCHESTRATION AND DATA COLLECTION FROM 5G CORE

The orchestration and data collection from 5G Core refer to the capability to the capability of the end-to-end network slice orchestration framework to automatically deploy, configure and operate 5G Core NFs as part of the network slices. It also refers to the interaction with the NWDAF for slice, NFs and UEs related data collection

Specifically, this innovation is mapped to the integration and validation of the specific NSSMF component developed for the Cumucore 5G Core and aims at managing, provisioning and operating the related network subnet slices. Moreover, as it involves the integration with the 5G Core NWDAF, the innovation is also mapped to the AI/ML and monitoring platform described in section 4.2.2, for the part related to the data collection and storage.

This innovation will be demonstrated as part of the Factory UC, and the related activities will be carried out in the different UC sites and testbeds (i.e., ASTI factory, UPV testbed and TUD testbed).

6.2.5 ORCHESTRATION OF APPLICATIONS AT EDGE/MEC

The orchestration of applications and NFs at the edge refers to the capability of the end-to-end network slice orchestration framework to support deployment, configuration, and operation of virtualized applications (e.g., specific vertical or use case applications) and NFs (e.g., 5G Core UPF) at the edge locations in support of specific latency or real-time requirements of network slices.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Specifically, this innovation is provided by the coordination (implemented by the NSSMF) of the specific NSSMFs components developed for the Cumucore 5G Core and the service applications (as described in Section 4.5.3). In particular, the deployment of application VNFs and 5G Core NFs (e.g., UPF) at the edge is coordinated by the specific NSMF, and then actuated by these two NSSMFs that interact with the specific NFV and resource orchestrators.

This innovation will be demonstrated as part of the Automated Robots with Heterogeneous Networks UC, carrying out the related activities in the different UC sites and testbeds (i.e., ASTI factory, and UPV and TUD testbeds).

6.2.6 AI-ASSISTED (NETWORK-DATA BASED) SLICE OPTIMIZATION

The AI-assisted network slice optimization is based on network monitoring data and refers to the capability of the end-to-end network slice orchestration framework to be assisted in the decision-making processes by external AI/ML algorithms. They process network slice related data collected from the 5G Core NWDAF and other network and compute infrastructure related sources.

Specifically, this innovation is mapped to the closed-loop functionalities implemented by the AI/ML and monitoring platform described in section 4.2.2. In particular, this innovation focuses on the optimization of the runtime operation of network slices, targeting the scenario described in section 4.3 and thus acting as an added value with respect to all the previous innovations.

This innovation will be demonstrated as part of the Automated Robots with Heterogeneous Networks UC, and the related activities are planned to be carried out at least in one of the UC sites and testbeds (i.e., ASTI factory, UPV testbed and TUD testbed).

6.2.7 AI-ASSISTED (IOT/DVL-DATA BASED) SLICE OPTIMIZATION

The AI-assisted network slice optimization (IoT/DVL data) refers to the capability of the end-to-end network slice orchestration framework to be assisted in the decision-making procedures by external AI/ML algorithms that process data related to IoT applications as collected and exposed by the DVL.

From a technical perspective, this innovation is similar to the previous one, in the sense that it can be mapped to the closed-loop functionalities provided by the AI/ML and monitoring platform. Specifically, in this case, the ML enabled analytics and decision features in the AI/ML engine should be able to derive and trigger network slice optimization by processing the IoT and M2M data, and possibly correlating it with network slice related data (as used in the previous innovation).

This innovation is linked to the Situational Understanding and Predictive Models in Smart Logistics Scenarios and Supply Chain Ecosystem Integration UCs. However, only a basic integration of the AI/ML and monitoring platform with the DVL is planned to be performed to validate the heterogeneity in the data collection and management aspects of the platform. For this reason, a dedicated AI/ML analytics and decision pipeline is not planned for development at the time of writing, and therefore the maturity of this innovation is kept at the conceptual level.

Co-funded by the Horizon 2020 Framework Programme of the European Union

7 Conclusions and Next Steps

The Management and Orchestration aspects are crucial in the iNGENIOUS architecture, and in particular aim at providing key automation and programmability features in the Network Layer functionalities. To this purpose, an end-to-end network slice orchestration framework has been designed to coordinate the automated deployment, configuration and operation of network slices in support of industrial IoT and supply chain 5G services.

The proposed end-to-end network slice orchestration framework is aligned and compliant with the relevant standard architectures and solutions provided by 3GPP and ETSI NFV. Specifically, this document has described the iNGENIOUS end-to-end network slice orchestration framework design approach, which is based on a cross-layer architecture that combines vertical services, network slices and resource management functionalities. In particular, the proposed orchestration framework is made of three main functionalities, namely VSMF, NSMF and NSSMF, which follow the 3GPP high level architecture for the management of network slices. The alignment with the 3GPP, GSMA and ETSI standards refer also to the supported information models, particularly with support of GSMA NEST and GST, as well as the 3GPP 5G Network Resource Model for NSTs and NFV descriptors for virtualized 5G network functions. To achieve full automation and align with current trends of native integration of AI/ML functionalities within network control and management functionalities, the proposed iNGENIOUS end-to-end network slice orchestration framework is also integrated with an AI engine that exploit a monitoring platform to assist the orchestration components in the runtime operation and optimization of network slice instances.

In addition to the architecture design, this document has also reported on the implementation of a preliminary software prototype for the iNGENIOUS end-to-end network slice orchestration framework. For this deliverable, a subset of the VSMF, NSMF and NSSMF functionalities has been developed, which anyway allowed to perform some initial integration and validation activities. In particular, the advancements on the integration with the Cumucore 5G Core, the Flexible PHY/MAC and O-RAN Near-RT RIC have been reported, which lay the foundation for the validation of the end-to-end network slice orchestration software stack in the iNGENIOUS use cases.

With specific relation to the project use cases, the iNGENIOUS end-to-end network slice orchestrator framework provides a set of innovations in the way the 5G-IoT network is managed to support industrial IoT and supply chain services. In particular, these innovations enable automated orchestration and slicing of the various relevant network technologies developed in the project, which are validated experimentally in the Automated Robots with Heterogeneous Networks use case and conceptually in the context of the Situational Understanding and Predictive Models in Smart Logistics Scenarios and Supply Chain Ecosystem Integration use cases.

In terms of next steps, while different development and integration activities have already started and reached a mature status, the work will be continued until the orchestration stack will be completed and ready to be reported in the final deliverable D4.5. In particular, these upcoming activities are related to the

Co-funded by the Horizon 2020 Framework Programme of the European Union

consolidation of some internal services of the VSMF, NSMF and NSSMF orchestration components, as well as to the integration with domain specific resource controllers and orchestrators. In summary, the main relevant next steps for the end-to-end network slice orchestration activities are:

- Finalization of the development of the VSMF, NSMF and NSSMF internal components, targeting a full stack prototype ready to be integrated and demonstrated as part of the iNGENIOUS use case activities
- Completion of the integration with the Flexible PHY/MAC, consisting of the implementation of a dedicated RAN NSSMF software prototype for the automatic deployment and joint configuration of 5G RAN emulation (UERANSIM) and Flexible PHY/MAC processing logic
- Completion of integration with the latest version of 5G Core containerized version, and support the automatic deployment and configuration of 5G Core instances through a dedicated Core Network NSSMF.
- Implementation of the Monitoring platform, and its integration with the end-to-end network slice orchestration software stack for automated collection of custom slice monitoring data
- Implementation of the AI/ML engine, with its analytics and decision functionalities, and specifically of the ML algorithm supporting the pre-emptive auto-scaling of local-edge UPFs.

Co-funded by the Horizon 2020 Framework Programme of the European Union

References

[1] iNGENIOUS D2.2, “System and Architecture Integration (Initial)”

[2] 3GPP TS 28.500, “Management concepts, architecture and requirements for mobile networks that include virtualized network functions (Release 16)”, v16.0.0, July 2020

[3] ETSI GS NFV-MAN 001, “Network Function Virtualisation (NFV); Management and Orchestration”, v1.1.1, December 2014

[4] 3GPP TS 28.528, “Life Cycle Management (LCM) for mobile networks that include virtualized network functions; Stage 3 (Release 16)”, v16.0.0, July 2020

[5] ETSI GS NFV-SOL 005, “Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point”, v3.3.1, September 2020

[6] 3GPP TS 23.501, “System architecture for the 5G System (5GS); Stage 2 (Release 17)”, v17.1.1, June 2021

[7] 3GPP TS 28.530, “Management and Orchestration; Concepts, use cases and requirements (Release 17)”, v17.1.0, March 2021

[8] 3GPP TS 28.541, “Management and Orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 17)”, v17.3.0, June 2021

[9] ETSI GS NFV-IFA 014, “Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Network Service Templates Specification”, v4.2.1, May 2021

[10] ETSI GS NFV-IFA 013, “Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Os-Ma-nfvo reference point - Interface and Information Model Specification”, v4.2.1, May 2021

[11] GSMA, “Generic Network Slice Template”, v5.0, June 2021

[12] 3GPP TR 28.801, “Study on management and orchestration of network slicing for next generation network (Release 15)”, v15.1.0, January 2018

[13] 3GPP TS 28.533, “Management and Orchestration; Architecture framework (Release 16)”, v16.7.0, March 2021

[14] iNGENIOUS D4.2, “Smart NR and NG-RAN IoT designs “

[15] iNGENIOUS D4.1, “Multi-Technologies Network for IoT”

[16] ETSI TS 128 531 V16.6.0 – “5G; Management and orchestration; Provisioning”

[17] ETSI TS 123 288 V16.4.0 - “5G; Architecture enhancements for 5G System (5GS) to support network data analytics services”

[18] 3GPP TS 29.520 V17.0.0 - “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; 5G System; Network Data Analytics Services;”

[19] ETSI TS 123 502 V16.5.0 – “5G; Procedures for the 5G System (5GS)”

[20] ONF Transport API,

Co-funded by the Horizon 2020 Framework Programme of the European Union

https://wiki.opennetworking.org/display/OTCC/TAPI+Overview

[21] YANG Data Modelling Language RFC 7950 - https://datatracker.ietf.org/doc/html/rfc7950

[22] P. Sayyad, P. Khodashenas, C. Fernandez, D. Guija, K. Liolis, C. Politis, G. Atkinson, J. Cahill, R. King, M. Kavanagh, B. Jou and O. Vidal, "Benefits and Challenges of Software Defined Satellite-5G Communication," in IEEE/IFIP WONS 2019, Wengen, Switzerland, 2019.

[23] Nextworks Slicer Open-source repository - https://github.com/nextworks-it/slicer

[24] ETSI GS NFV-SOL006, "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; NFV descriptors based on YANG Specification", v3.5.1, July 2021

[25] iNGENIOUS D4.3, "Core network automation design for 5G-IoT"

[26] ETSI Open Source MANO (OSM), https://osm.etsi.org/

[27] Open-source UERANSIM tool repository https://github.com/aligungr/UERANSIM

[28] Open-source Apache Kafka Event data analytics streaming platform - https://kafka.apache.org/

[29] Open-source Telegraf server agent - https://www.influxdata.com/time-series-platform/telegraf/

[30] Open-source Prometheus Monitoring Platform - https://prometheus.io/

[31] InfluxDB Timeseries-based platform - https://www.influxdata.com/

[32] ntop Traffic analysis open-source tool - https://www.ntop.org/

[33] Spring Boot Framework - https://spring.io/projects/spring-boot

[34] Postgres SQL - https://www.postgresql.org/

[35] Guava Event Bus - https://www.baeldung.com/guava-eventbus

[36] OpenStack cloud Software - https://www.openstack.org/

[37] cloud-init documentation - https://cloudinit.readthedocs.io/en/latest/

[38] Juju as Service - https://jaas.ai/

[39] Open5GS website - https://open5gs.org/

Co-funded by the Horizon 2020 Framework Programme of the European Union