2

AI-driven Service and Slice Orchestration

G. Bernini, P. Piscione, and E. Seder

Nextworks, Italy
E-mail: g.bernini@nextworks.it; p.piscione@nextworks.it;
e.seder@nextworks.it

Abstract

Current MANO solutions and existing tools for network slicing and service orchestration are still implemented as silo-based control and orchestration tools, mostly addressing the coordination of monolithic pipelined services that cannot be easily and transparently adapted to dynamic NG-IoT network and service conditions. Lack of agility and flexibility in the service and slice lifecycle management, as well as in the runtime operation, is indeed still an evident limitation. A tight integration of AI/ML techniques can help in augmenting the slice and service orchestration logics automation and intelligence, as well as their self-adaptation capabilities to satisfy NG-IoT service dynamics.

Keywords: NG-IoT, 5G, network slicing, orchestration, 3GPP, artificial intelligence, machine learning.

2.1 Introduction

In general, a 5G network infrastructure that provides an end-to-end connectivity service in the form of a network slice to the end users requires proper resource allocation and management. The allocation and the management of these resources become critical, especially when the number of user equipment (UE) starts to increase. To this end, the next-generation IoT (NG-IoT) network slice orchestration supported by an artificial intelligence/machine learning (AI/ML) platform plays a crucial role, performing semi-automated decisions on resource allocation and management.

With 5G, the telecommunication industry is more and more looking at comprehensive management and orchestration (MANO) solutions to ease the deployment of heterogeneous vertical services and network slices across several technology domains. The concept of network slicing allows to jointly orchestrate resources (network, computing, and storage) and network functions (NFs) (virtualized or physical), which are managed and delivered together to instantiate and compose network services over a shared infrastructure. Network slices can be dynamically created and customized according to the requirements of the services that will run on top of them, for example, in terms of resource or function isolation and quality of service (QoS) guarantees. This has been considered in the iNGENIOUS project [1], 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.

Current MANO framework solutions and existing tools for network slicing and NFV network service orchestration are still implemented as silo-based control and orchestration tools, mostly addressing the coordination of monolithic pipelined services that cannot be easily and transparently adapted to changing network and service conditions. Lack of agility and flexibility in the service and slice lifecycle management is still an evident limitation, thus requiring *ad-hoc* solutions and customizations for addressing the challenging NG-IoT time sensitive networking and ultra-low-latency requirements. Moreover, a full integration of 5G new radio (NR), NG-IoT, and edge computing technology domains is not yet achieved when it comes to deploying end-to-end network slices. Moreover, the overall capability of such orchestration approaches to fulfill heterogeneous service constraints and requirements still needs to be proved, as it often requires per-service customizations and human-driven adjustments to support end-to-end deployments. In addition, the adoption of AI/ML technologies for cognition-based optimizations, including their interaction across the different technological domains (e.g., network related, edge computing related, cloud computing related, etc.) and their tight integration with the service and slice lifecycle management is still at its early stages.

The current MANO coordination functionalities are highly linked to static internal coordination and orchestration logic. The management operations at different levels follow the workflows, which MANO is responsible for implementing. This results in lack of flexibility because when either minor adjustments are needed or unplanned events occur, MANO remains strict to its static coordination and orchestration logic. In this context, a tight integration with AI/ML techniques could address this kind of problem. AI/ML algorithms generally do not follow the if−then approach, but they are able to "learn" from past experience and, in some cases, take decisions. For the aforementioned reasons, in the iNGENIOUS project, one key innovation for what concerns the orchestration aspects is the intelligent management and orchestration of network resources for the NG-IoT ecosystem. In this context, since the resource demand could be fluctuating during a time period and at the same time the high-level requirements must be satisfied, a semi-automated decision-based approach comes into place.

2.2 Related Work

2.2.1 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 manager (EM) that control the elements composing a 5G network, where each of them can be deployed as a physical network function (PNF) or a virtual network function (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.

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 of 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

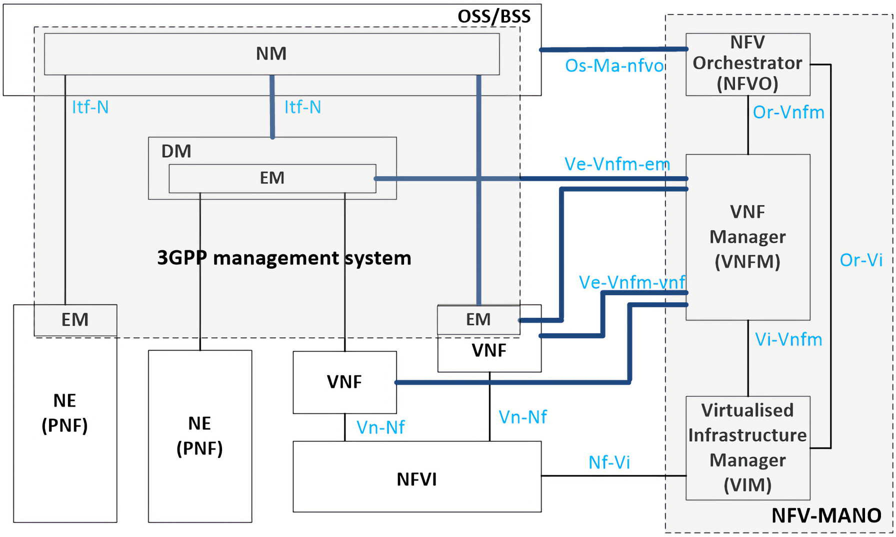

Figure 2.1 Mobile network management architecture – interaction between 3GPP management system and NFV-MANO framework [2].

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.

2.2.2 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, pedestrians, etc.

The concept of network slicing is introduced in the 3GPP TS 23.501 [4] 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. An NF can be implemented within an 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 toward 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 [5] 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. Different types of CS can be supported through dedicated NSIs. An NSI can support one or more instances of CS. Moreover, an NSI is formally modeled 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 figure shows a common pattern of network slice modeling, 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, an 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.

2.2.3 Information models for 5G network slices

The information model of an NSI is defined in the 3GPP TS 28.541 [6], as part of the 5G network slice resource model (NRM). The model, represented in Figure 2.2, 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.

A network slice is associated with 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

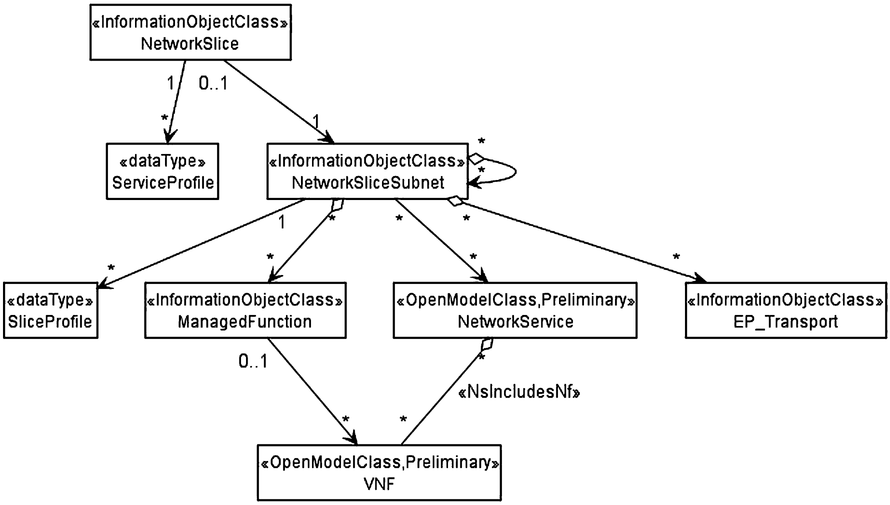

Figure 2.2 Structure of network slices and network slice subnets in network services and virtual network functions [6].

area, maximum latency, per-slice and per-UE throughput in uplink and downlink, maximum number of allowed connections, jitter, UEs' maximum speed, etc.

On the other hand, an NSD represents the topology of a network service, identifying its internal network functions (through references to the VNF 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.

The 3GPP information model reported in Figure 2.2 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 toward 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) [7] defined by the GSM association (GSMA).

2.2.4 Management of 5G network slices

The 3GPP TR 28.801 specification [8] 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).

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 that 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 related to the NSSMFs, which are finally responsible for the management and orchestration of each NSSI.

As analyzed in the 3GPP TS 28.533 specification [9], 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 NSSMFs are customized according to the specific requirements and technologies adopted in their own target domain. As detailed in Section 2.4, 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 2.3.

3GPP standards do not mandate any specific implementation of the NSMF and NSSMF components. However, the 3GPP TR 28.533 specification [9] proposes a deployment option, widely used in production infrastructures, where the management of the network slices and slice subnets lifecycle

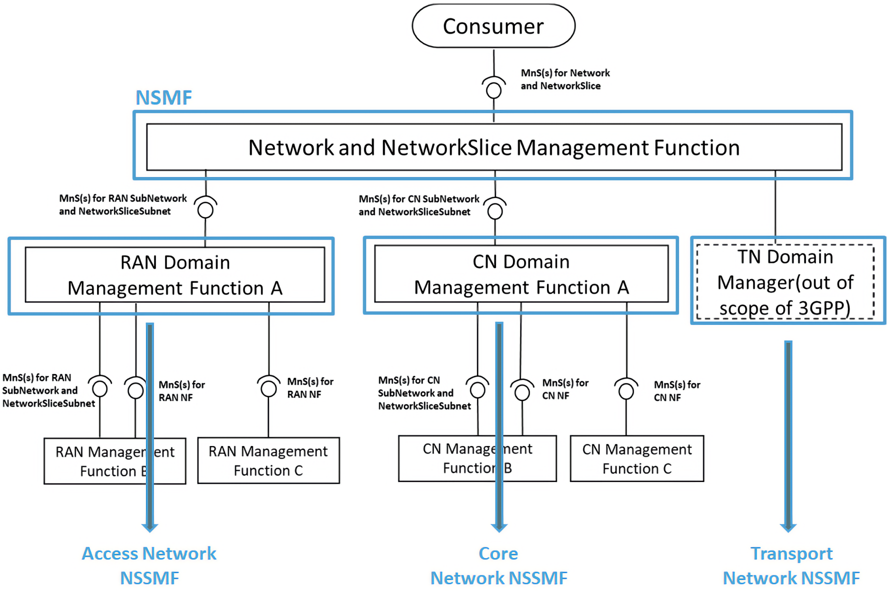
Consumer
MnS(s) for Network and NetworkSlice
NSMF
Network and NetworkSlice Management Function
MnS(s) for RAN SubNetwork and NetworkSliceSubnet
MnS(s) for CN SubNetwork and NetworkSliceSubnet
RAN Domain Management Function A
CN Domain Management Function A
TN Domain Manager(out of scope of 3GPP)
MnS(s) for RAN SubNetwork and NetworkSliceSubnet
MnS(s) for RAN NF
MnS(s) for RAN NF
MnS(s) for CN SubNetwork and NetworkSliceSubnet
MnS(s) for CN NF
MnS(s) for CN NF
RAN Management Function B
RAN Management Function C
CN Management Function B
CN Management Function C
Access Network NSSMF
Core Network NSSMF
Transport Network NSSMF

Figure 2.3 Hierarchical interaction between NSMF and per-domain NSSMFs [9].

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 with the NSSIs. The orchestration solution proposed by iNGENIOUS is aligned with 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.

2.3 Architectural Principles

The end-to-end network slice orchestration framework solution proposed by the iNGENIOUS project has been conceived to satisfy the following architectural principles:

- *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 the 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.
- *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.

2.4 Functional Architecture

The main principles and motivations described in the previous section led to the specification of the end-to-end network slice orchestration framework. In Figure 2.4 is available a mapping between the functional architecture described in the 3GPP TR 28.801 specification (Figure 2.4(a)) 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 2.4(b)).

Figure 2.4(b) 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, etc.).

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).

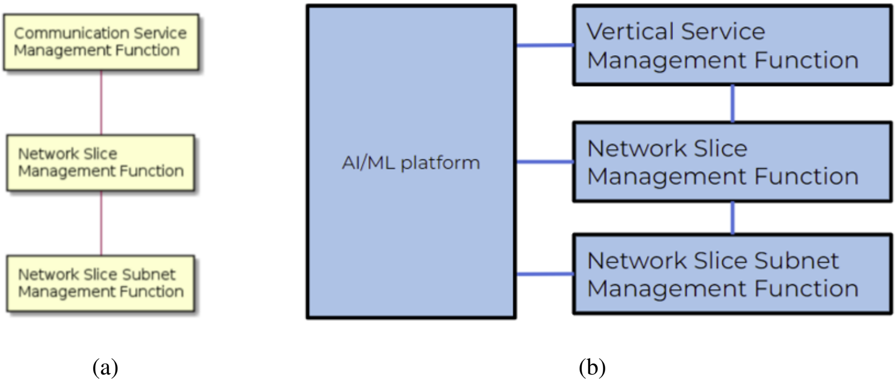
Communication Service Management Function
Network Slice Management Function
Network Slice Subnet Management Function
AI/ML platform
Vertical Service Management Function
Network Slice Management Function
Network Slice Subnet Management Function
(a)
(b)

Figure 2.4 End-to-end network slice orchestration high-level architecture supported by AI/ML platform.

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 following 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.

2.4.1 Orchestration components

This section describes the internal components of the end-to-end network slice orchestration framework. Figure 2.5 depicts the functional architecture of the end-to-end network orchestration framework, derived from the high-level view of Figure 2.4. In particular, the 3GPP CSMF functionalities are realized by the vertical service management function (VSMF), the 3GPP

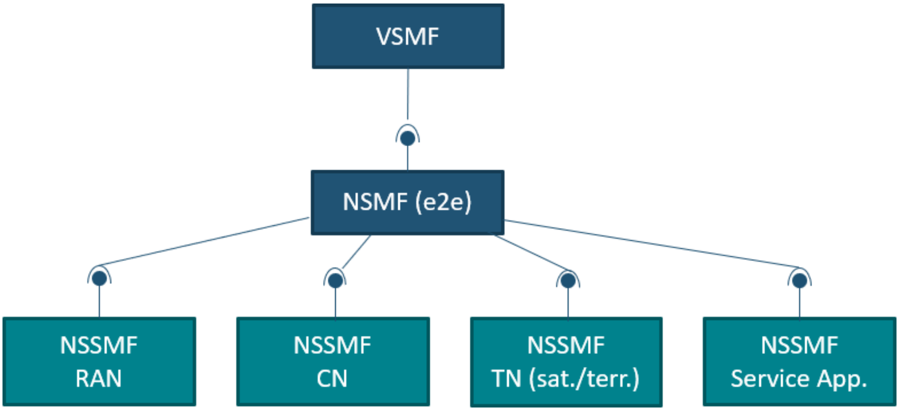

Figure 2.5 High-level software architecture end-to-end network orchestration framework.

NSMF functionalities are realized by the end-to-end NSMF, and finally the 3GPP NSSMF layer is mapped into multiple specific technology-tailored NSSMFs.

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 modeled), the following sub-sections detail the functional decomposition and internal design of the VSMF, NSMF, and NSSMF components.

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 [10]) 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, which 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 [6], which provides network requirements and related resources' configuration. In particular, the NST, whose simplified class diagram is shown in Figure 2.6, contains a list of service profiles, each of them specifying the network slice type and the related

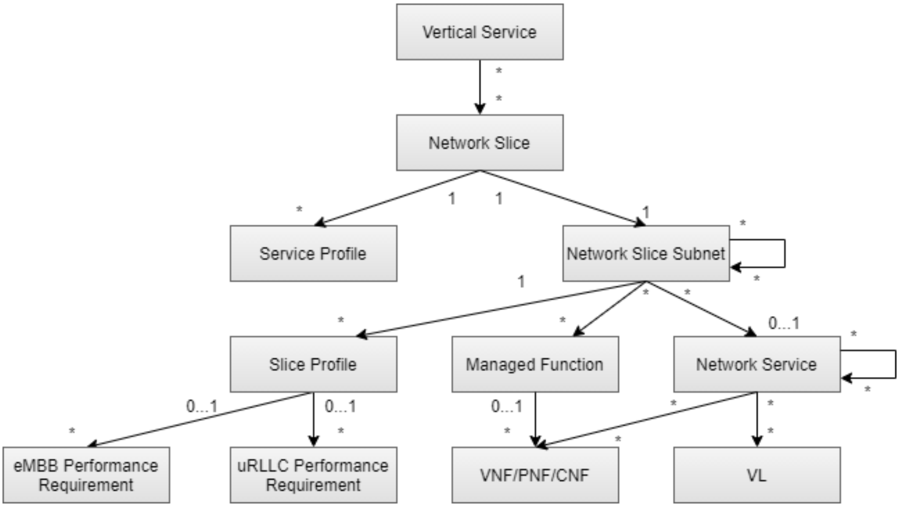

Figure 2.6 Network slice simplified class diagram.

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 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.

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.

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, etc.).

Network service management function:

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).

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 [7]. Similarly, the NST and NSST, in compliance with the 5G NRM [6] (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 drive the whole lifecycle management of end-to-end network slices, implemented by the different components available within the NSMF.

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 2.7.

Each specific network domain implements its own mechanisms, data models, REST APIs, and workflows for allocating computing and network

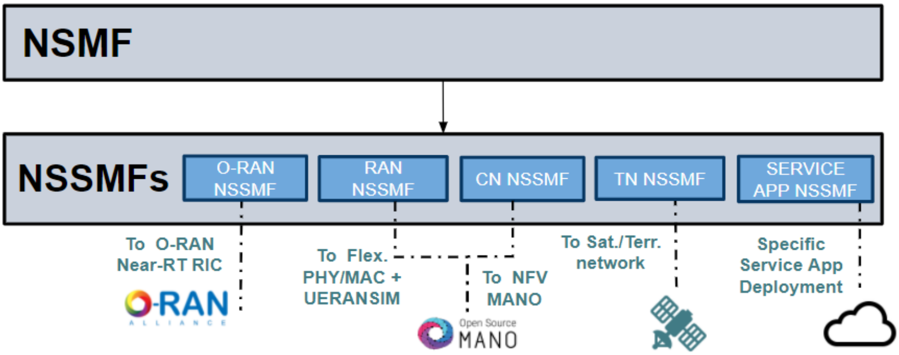

Figure 2.7 High-level architecture of NSSMF layer.

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 follow a generic and unified functional decomposition, which aims at providing a set of common functionalities, which include: a northbound interface (NBI) for exploiting the NSSMF functionalities and for receiving subnet slice related requests (e.g., REST APIs), a core NSSMF service for validating and dispatching the requests into an event bus, publishing them as events, an event bus to allow communications among components using a topic-based and publish–subscribe mechanisms, an NSSMF handler to receive and process multiple requests and realizes the internal logic of the NSSMF.

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, and interfaces supported. Some examples of NSSMFs developed in the context of the iNGENIOUS project are: O-RAN NSSMF, providing the translation of slice profiles into O-RAN A1 policies and A1 policy management operations in the O-RAN near real-time RAN intelligent controller (RIC); 5G Core network NSSMF, providing automated LCM, and configuration of 5G Core NFV network services through ETSI OSM [11]. The network service contains a 5G Core instance consisting of the control plane and user plane network functions of a 5G Core; service application NSSM, providing automated LCM and configuration of NFV network services modeling service virtual applications through ETSI OSM [11].

2.4.2 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 that 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 2.8 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 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.

The monitoring platform provides both data storing and streaming functionalities, with proper interfaces exposed toward the AI/ML engine to consume the monitoring data. The data can be collected from different and heterogeneous data sources through the adaptation layer, which 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 plug-ins may vary with respect to the different network slices to be monitored or during the different phases of a network slice lifetime. A configurable Alert Manager (which is a built-in component of Prometheus) sends alarms to the bus when specific data

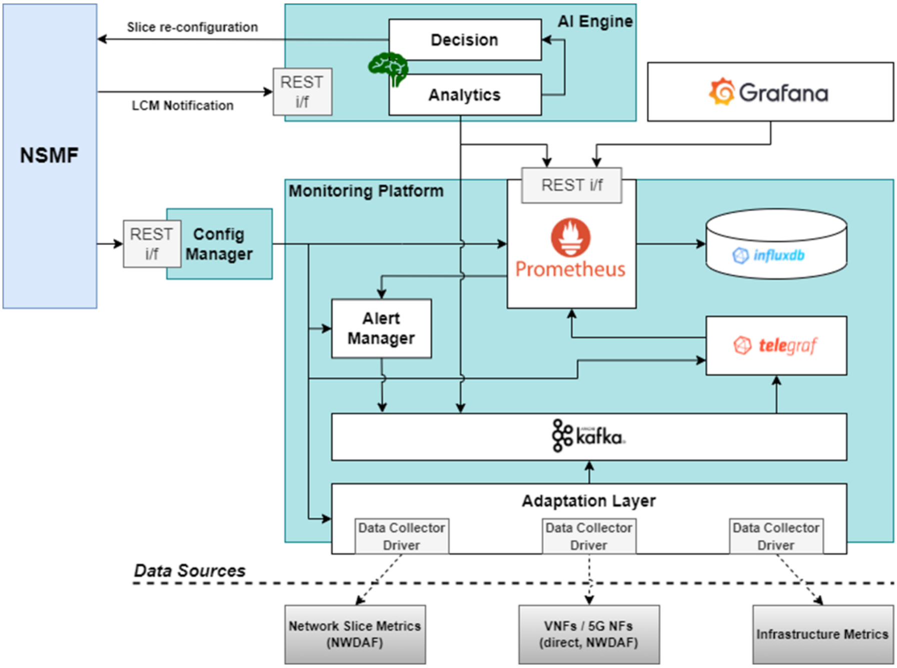

Figure 2.8 AI/ML and monitoring platform functional architecture.

exceeds certain thresholds, so that the alarm notification can be captured and stored in the Data Lake. The alarms and the data, both historical and near-real time, are therefore immediately available to the AI/ML engine that can access both the Data Lake and the message bus through dedicated interfaces. The whole monitoring platform is configured by the NSMF through the Config Manager, which for each network slice instance can tailor the behavior 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 message 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.

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 four stages designed for robust functionality on real-world data:

- Stage 1. Data pre-processing – real-time data contains many irregularities (e.g., null values) unrelated to the useful information derived from the target analysis. This noise can directly affect the ability of models to reliably infer behaviors in the incoming data. The data pre-processor cleans and normalizes the incoming dataset to avoid misbehavior of the model on real-world data.
- Stage 2. Feature detection – correlated time series data is analyzed with respect to long term behaviors that create unique features that can be used to predict future trends in network behavior. Selection of these features is achieved through the use of trained models capable of discriminating target behaviors.
- 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 optimize 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 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, etc.) 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.

2.5 Example of AI/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 that 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, which can be a simple "do nothing" or slice optimization requests to be enforced toward 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 in iNGENIOUS 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 2.9. A single UPF instance can handle multiple protocol data unit (PDU) sessions; however,

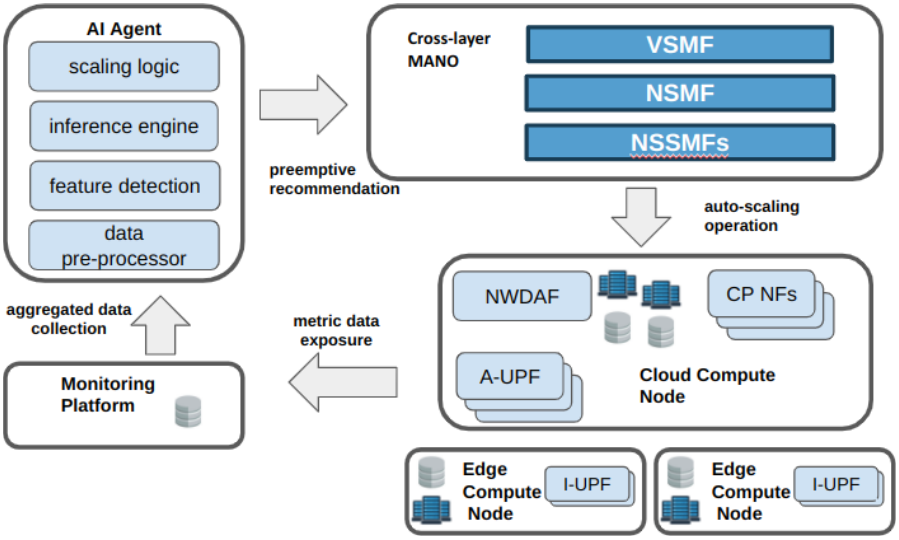

Figure 2.9 Closed-loop pre-emptive auto-scaling of UPF.

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 monitors, measures, and assesses real-time network data, and then automatically acts to optimize 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.

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. 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 that are ingested after a pre-processing by the AI agent that performs a decision about the pre-emptive auto-scaling operation on UPF itself.

Acknowledgements

This work has been partially supported by the European Union's Horizon 2020 research and innovation program through the project iNGENIOUS under grant agreement No. 957216.

References

[1] H2020 iNGENIOUS, https://ingenious-iot.eu/web/

[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 23.501, "System architecture for the 5G System (5GS); Stage 2 (Release 17)", v17.1.1, June 2021

[5] 3GPP TS 28.530, "Management and Orchestration; Concepts, use cases and requirements (Release 17)", v17.1.0, March 2021

[6] 3GPP TS 28.541, "Management and Orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 17)", v17.3.0, June 2021

[7] GSMA, "Generic Network Slice Template", v5.0, June 2021

[8] 3GPP TR 28.801, "Study on management and orchestration of network slicing for next generation network (Release 15)", v15.1.0, January 2018

[9] 3GPP TS 28.533, "Management and Orchestration; Architecture framework (Release 16)", v16.7.0, March 2021

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

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