Deliverable D2.2

Overall Architecture and Interfaces Definition

Editor(s):	José Cabaça, Pedro Neves and Rui Calé, Altice Labs
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	30th November 2017
Actual delivery date:	26th January 2018
Suggested readers:	Infrastructure Providers, Communication Service Providers, Digital Service Providers Network Operators, Vertical Industries,
Version:	1.0
Total number of pages:	102
Keywords:	Network Slice, Cognition, Multi-Domain, 5G, SDN, NFV, Roles, Stakeholders, Technical Use-Cases, Architecture

Abstract

The main objective of this document is to describe the overall architecture of the SLICENET project. To achieve this aim, the document presents the main stakeholders and actors of the SLICENET system, as well as the main principles – Network Slicing, Plug & Play, One-Stop API, Cognition, Cross-Plane Orchestration and Multi-Domain – that guided the architecture design phase. An overall perspective of the architecture is described, including a logical and a functional view covering the most relevant use-cases related with network slicing, cognition and multi-domain.

Disclaimer

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

All SLICENET consortium parties have agreed to full publication of this document.

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

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

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

The research leading to these results has received funding from the European Union Horizon 2020 Programme under grant agreement number 761913.

Impressum

[Full project title] End-to-End Cognitive Network Slicing and Slice Management Framework inVirtualised Multi-Domain, Multi-Tenant 5G Networks

[Short project title] SLICENET

[Number and title of work-package] WP2 – SLICENET System Definition

[Number and title of task] T2.2 – Overall Architecture and Interfaces Definition

[Document title] Overall Architecture and Interfaces Definition

[Editor: Name, company] José Cabaça, Pedro Neves and Rui Calé, Altice Labs

[Work-package leader: Name, company] Pedro Neves, Altice Labs

Copyright notice

List of authors

Company	Author
ALTICE LABS SA, Portugal	Bruno Parreira, Gonçalo Gaspar, José Cabaça, Mário Rui Costa, Pedro Neves, Rui Calé
CIT INFINITE DESIGNATED ACTIVITY COMPANY, Ireland	Mark Roddy, Paul Walsh
CREATIVE SYSTEMS ENGINEERING (C.S.E) MONOPROSOPI EPE, Greece	John Vavourakis,Konstantinos Koutsopoulos
Dell EMC INFORMATION SYSTEMS INTERNATIONAL	Zdravko Bozakov, Thuy Truong
EFACEC ENERGIA - MAQUINAS E EQUIPAMENTOS ELECTRICOS SA	Ana Cristina Aleixo
Eurecom	Chia-Yu Chang, Navid Nikaein, Xenofon Vasilakos
EURESCOM-EUROPEAN INSTITUTE FOR RESEARCH AND STRATEGIC STUDIES IN TELECOMMUNICATIONS GMBH	Anastasius Gavras, Maria João Barros
IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD	Dean H Lorenz, Katherine Barabash, Valleriya Perelman
NEXTWORKS	Giacomo Bernini, Pietro G. Giardina
ORO	Adrian Matei, Catalin Costea, Cristian Panes, Cristian Patachia, Horia Stefanescu, Madalina Oproiu, Marius Iordache, Vlad Sorici
HELLENIC TELECOMMUNICATIONS ORGANIZATION S.A. - OTE AE	Christina Lessi, George Agapiou
ERICSSON TELECOMUNICAZIONI	Carmine Galotto, Ciriaco Angelo, Giuseppe Celozzi, Raffaele De Santis
UNIVERSITAT POLITECNICA DE CATALUNYA	Albert Pagès, Fernando Agraz, Salvatore Spadaro
UNIVERSITY OF THE WEST OF SCOTLAND	Jose M. Alcaraz Calero, Hector Marco, Qi Wang, Zeeshan Pervez

Table of Contents

List of figures

List of tables

Abbreviations

5G	Fifth Generation (mobile/cellular networks)
5G PPP	5G Infrastructure Public Private Partnership
DSL	Digital subscriber line
FED	Federal Reserve System
HoN	Health of Network
IoT	Internet of Things
KPI	Key Performance Indicator
M2M	Machine to Machine
MRO	Maintenance, repair and operations
NAFTA	North American Free Trade Agreement
OPEX	Operational Expenditure
QoE	Quality of Experience
OPEX	Operational Expenditure
QoS	Quality of Service
R&D	Research and Development
SDN	Software Defined Networks
SLICENET	End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks
SON	Self-Organizing Networks
SWOT	Strengths, Weaknesses, Opportunities, and Threats analysis
VAT	Value-added tax

1 Introduction

The SLICENET project foundations are settled in six main pillars. The first one is the Network Slicing concept, which will enable service providers to open and monetize their main asset – the network – to the vertical industry. From the vertical industry perspective, it will provide the required flexibility and readiness to support the emerging 5G business requirements. The second SLICENET pillar, acting as a complement to the Network Slicing concept, is the ability to deliver and abstract from the Vertical a Network Slice that crosses multiple service providers, also known as administrative domains. This will enable the Vertical to simply request for a Network Slice with certain capabilities to accommodate its business requirements and it will the responsibility of the slice provider to prepare a composed service offer that is able to cross multiple administrative domains. The third pillar is the One-Stop API, which reflects a single-entry point and abstraction for the vertical to reach the system architecture functionalities. Plug & Play is the fourth pillar which provides an innovative combination of customized control functions, APIs and tools to enable verticals to even plug their own control logics and specialize their slices according to their needs, offering significantly enhanced degree of flexibility for tailored services to end users. The fifth pillar of SLICENET is the cognitive network management. Autonomous management and control is evolving from self-healing, self-optimizing and self-configuring automation to artificially intelligent systems that can outlearn their current knowledge and apply newly gained wisdom to achieve network goals through self-decision making. Finally, the six core aspect is the cross-plane orchestration capability, providing a set of coordination functions across several logical layers and constructs (i.e. service, slice, resource, and infrastructure) with the aim orchestrating the provisioning of end-to-end slices.

The abovementioned principles are the basis for the reported work on this document, which is the result of task 2.2, namely, the design of the SLICENET overall system architecture. The designed architecture, although still at a high-level, is the first SLICENET exercise with respect to this matter. Additionally, it also serves as input for the other project activities which will further detail the architecture – task 2.3 with respect to the SLICENET control aspects and task 2.4 on the management side.

In terms of architecture design working strategy, the first step is to research all the ongoing activities related with technical pillars of the project, this includes other 5G projects, open source projects and standardization efforts. The result of this activity led to the creation of the SLICENET related work radar which is an important input for this document, as well as for the lifetime of the project. In parallel, and based on the vertical use-cases identified in D2.1, it is also paramount to identify the SLICENET stakeholders and their business relevance on the architecture environment, enabling us to identify how and when they will interact with the system. Additionally, still based on the vertical use-cases it is also important to extract the most relevant technical use-cases and the related technical requirements.

All this together – related work, stakeholders, technical use-cases and requirements – are the fuel for the system architecture design. Within this document, besides providing the high-level perspective of the architecture, the objective is to go one step further and design a more detailed perspective, already including the core logical components of the system and their main interactions.

With respect to the document structure, after a brief introduction, section 2 provides an overview of the relevant related work for SLICENET, covering 5G projects, open source projects and standardization bodies. Section 3 focuses on the identification of the business roles related with providing a slice to a SLICENET customer (typically a vertical industry). The core SLICENET principles, such as 1) Network Slicing, 2) Cognition (Proactive Management), 3) One-Stop API, 4) Plug & Play and 5) Cross-Plane Orchestration are described in section 4. Thereafter, section 5 provides the main requirements for the architecture design, as well as the most relevant technical use-cases. Section 6 provides a first, overall perspective of the SLICENET system architecture, including a high-level view, also known as “Level 0”, and a more detailed view, known as “Level 1”. Furthermore, as a complement to the system architecture definition, a group of workflows was exercised and illustrated to explain how the identified architecture components interaction with each other.

2 Related Work

2.1 Related 5G Projects

2.1.1 Overview of Related 5G Projects

The most related 5G projects are from 5G PPP Phase I and Phase II projects, which are the focus of this subsection. It is noted that SLICENET is a Phase II project, and thus can benefit from exploring the outcomes of Phase I projects, most of which will complete by the middle of 2018. For related Phase II projects, SLICENET will seek collaboration to jointly define and develop key enablers for slicing and slice management etc. in architecture design and beyond. Table 1 summarizes the most relevant Phase I projects and highlights the potential areas to be exploited in SLICENET.

Table 1: Overview of related 5G PPP Phase I projects and potential exploitation in SLICENET

Related Project	Main Topic of the Related Project	Potential Exploitable Design and/or Implementation Results (through Reuse, Extension and/or Adaptation)
5GEx [1]	Focus on enabling cross-domain orchestration of services over multiple administrations or over multi-domain single administrations. This will allow end-to-end network and service elements to mix in multi-vendor, heterogeneous technology and resource environments.	Reusing/extending cross-domain orchestrator and E2E services for multi-domain slicing negotiation and slice management. The 5GEx solution is based on T-Nova's market place and UNIFY project Orchestrator which have been extended for multi-domain environment and orchestration of VNFs. From multi domain network point of view the Virtual Path Slice engine which evolved from EuQoS and CityFlow projects is used.
COHERENT [2]	Focus on control, coordination and flexible spectrum management in heterogeneous radio access networks, providing a unified programmable control framework to coordinate the underlying heterogeneous mobile networks as a whole.	Network programmability, to coordinate the data plane towards the achievement of slicing security and performance isolation.
SELFNET [3]	Proposes an SDN-/NFV-based network management framework for advanced Self-Organizing Network (SON) capabilities in 5G infrastructures. The SON capabilities are demonstrated by mean of a set of use cases to show self-protection, self-healing and self-optimization capabilities.	Automated orchestration functions for optimizing the VM and VNF placement and automatically reacting in case of failures at the infrastructure level.
CogNet [1]	Focus on Autonomic Network Management based on Machine Learning enabling an (almost) self-administering and self-managing network.	Achieving cognition/QoE management by using machine learning algorithms applied over network elements.
5G-NORMA [5]	Focus on a novel mobile network architecture to support multi-service and multi-tenant to handle fluctuations in traffic demand dynamically changing	Inter- and intra-slice MANO framework, SDN and NFV defined concepts, (RAN) edge cloud, service management and

	service portfolios	orchestration
SONATA	Focus on the development of a customized SDK to boost the efficiency of developers of network functions and composed services, by integrating catalogue access, editing, debugging, and monitoring analysis tools with service packaging for shipment to an operator.	Reusing tools already available in SONATA to speed up the results of SLICENET; vision of integrating network slicing in the SONATA platform.
5G-Ensure	Focus on defining the 5G Security Architecture needed to expand the mobile ecosystem giving operators a platform for entirely new business opportunities.	Reusing the knowledge and tools achieved in 5G-Ensure and applying it to securing slices and slice management operations.
SuperFluidity	Focus on achieving the ability to instantiate services on-the-fly, run them anywhere in the network (core, aggregation, edge) and shift them transparently to different locations.	Reusing service operation capabilities achieved and applying them to the concept of slice and slice management.

Table 2 lists the most related Phase II projects and indicates potential areas for collaboration.

Table 2: Overview of related 5G PPP Phase II projects and potential collaboration with SLICENET

Related Project	Main Topic of the Related Project	Potential Technical Collaboration Areas
5G Transformer	Enable Vertical Industries to meet their service requirements within customized 5G Mobile Transport and Computing Platform (MTP) slices.	Vertical informed design; slicing.
MATILDA	Intelligent and unified orchestration mechanisms for the automated placement of applications and the creation and maintenance of the required network slices.	Multi-site virtualized infrastructure manager, VNF Forwarding Graphs (VNF-FGs), Multi-site NFV Orchestrator (NFVO). As ORO is part of both SLICENET and MATILDA consortium, it will be created a collaborative work for the Smart City use case implementation, through coordination and collaboration between the projects.
5G Monarch	Use network slicing to support a variety of use cases in vertical industries such as automotive, healthcare, and media.	Inter-slice control and cross-domain management, to enable the coordination across slices and domains.
5G ESSENCE	Edge Cloud Computing (MEC) & Small Cell-as-a-Service.	MEC
5G City	Distributed cloud and radio platform for municipalities and infrastructure owners acting as 5G neutral hosts.	Slice, and MEC.
5G Media	Innovating media-related applications by investigating how these applications and the underlying	Slicing, QoE, and UHD based video use case.

	5G network should be coupled and interwork to the benefit of both.	
5G Picture	Creation and deployment of programmable network functions and intelligent orchestration schemes.	Service chaining, slicing and multi-tenancy.
5GTango	Integrated vendor-independent platform for automatic testing and validation of packaged NFV forwarding graph of composed services.	Slicing, service development kit, store platform, and service platform.
NGPaaS	Telco-grade PaaS to support different configurations and a large set of deployment options (FPGA/ARM/x86 etc.), private/public cloud in a scalable and unifying manner.	5G Platform-As-A-Service (PaaS).

2.2 Related Open Source Projects

2.2.1 Overview of Related Open Source Projects

In addition to the above 5G and telecom projects, there are a large number of relevant open source projects which match SLICENET requirements, objectives and technological challenges. Table 3 summarizes a collection of most related ones identified at this stage of the project. In the table, for each project, the main focus is outlined, primarily taken from the descriptor of the individual project's official website. The potential exploitation regarding SLICENET planes is highlighted as a starting point for further investigation of selected projects along the R&D progress of SLICENET.

Table 3: Overview of management related open source projects

Name	Potential Exploitation	Organization	Brief Description
ONAP [6]	Management and Control (NFV MANO, ODL)	Linux Foundation	Goal is to create a harmonized and comprehensive framework for real-time, policy-driven software automation of virtual network functions that will enable software, network, IT and cloud providers and developers to rapidly create new services.
OpenBaton [7]	Management (NFV MANO)	Fraunhofer / TU Berlin	OpenBaton is a ETSI NFV compliant Network Function Virtualization Orchestrator (NFVO).
Open Source MANO (OSM) [8]	Management (NFV MANO)	ETSI	An Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV

Open-O [9]	Management (NFV MANO)	Linux Foundation	OPEN-Orchestrator Project (OPEN-O) framework and orchestrator to enable agile software-defined networking (SDN) and network function virtualization (NFV) operations
OpenMano [14]	Management (NFV MANO)	Telefónica NFV Labs	OpenMano provides a practical implementation of the reference architecture for Management & Orchestration under standardization at ETSI's NFV ISG
OpenStack [15]	Management (VIM)	OpenStack Foundation	Open source software for managing the virtual infrastructure of private and public clouds
OPNFV [16]	Management (NFV MANO)	Linux Foundation	Accelerating NFV's evolution through an integrated, open platform.
Tacker [17]	Management (NFV MANO)	OpenStack	Tacker is an official OpenStack project building a Generic VNF Manager (VNFM) and a NFV Orchestrator (NFVO) to deploy and operate Network Services and Virtual Network Functions (VNFs) on an NFV infrastructure platform like OpenStack. It is based on ETSI MANO Architectural Framework and provides a functional stack to Orchestrate Network Services end-to-end using VNFs
TeNOR (T-NOVA) [18]	Management (NFV MANO)	FP7 T-NOVA project	A novel enabling framework, for deploying vNFs and offering them to customers, as value-added services implementing the concept of NFaaS.
Docker [19]	Management (App management)	Docker	Software containerization platform
Juju [20]	Management (automation etc.)	Canonical Ltd.	Deploy, configure, scale and operate software on public and private clouds
Kubernetes [21]	Management(orchestration)	Linux Foundation	Application container orchestration engine providing an abstraction layer for managing full stack operations of hosts and containers.

Table 4: Overview of control plane related open source projects

Name	Potential Exploitation	Organization	Brief Description
ONOS [22]	Control plane (SDN controller)	ONF/ON.Lab	Carrier-grade SDN network operating system designed for high availability, performance, scale-out
OpenDaylight (ODL) [23]	Control plane (SDN controller)	Linux Foundation	Production-ready open SDN platform containing features, protocols and plug-ins that can be integrated in a number of ways to deliver a broad set of SDN use cases
Ryu [24]	Control plane (SDN controller)	NTT Labs	The Ryu Controller provides software components, with well-defined APIs, that make it easy for developers to create new network control applications.
OpenContrail Controller [25]	Control (service chaining, SDN controller etc.) and management/cognition planes (analytics etc.)	Juniper Networks	OpenContrail is an Apache 2.0-licensed project that is built using standards-based protocols and provides all the necessary components for network virtualization–Controller, virtual router, analytics engine, and published northbound APIs. It has an extensive REST API to configure and gather operational and analytics data from the system. Built for scale, OpenContrail can act as a fundamental network platform for cloud infrastructure.

Table 5: Overview of Cognition-related open source projects

Name	Potential Exploitation	Organization	Brief Description
Nagios Core [26]	Cognition (monitoring)	Nagios Enterprises	Monitoring tool that checks scheduling, execution, and processing, and deals with event handling and alerting.
TensorFlow [27]	Cognition (machine learning)	Google	Open source library for machine learning applications such as neural networks.
Zeppelin [28]	Cognition (machine learning and	Apache	Web-based Interactive data science and scientific computing across multiple programming languages.

	analytics)		
Spark [29]	Cognition (machine learning)	Apache	A general, fast, scalable engine for large-scale data processing. Libraries to ML, streaming and graph analytics.
sFlow-RT [30]	Cognition (monitoring of data plane, analytics)	InMon	sFlow-RT delivers real-time visibility to Software Defined Networking (SDN), DevOps and Orchestration stacks.
SkyDive [31]	Cognition (monitoring, analytics)		Open source real-time network topology and protocols analyzer.
Jupyter [32]	Cognition (machine learning and analytics)	Jupyter	Web-based Interactive data science and scientific computing across multiple programming languages, born out of IPython project.
Ceilometer [33]	Cognition (monitoring)	OpenStack Foundation	Data collection tool for OpenStack components to provide customer billing, resource tracking, and alarming capabilities etc.
Monasca [34]	Management plane (monitoring)	OpenStack Foundation	Monitoring-as-a-service solution for OpenStack.

Table 6: Overview of Data plane related open source projects

Name	Potential Exploitation	Organization	Brief Description
ClickOS [35]	Data plane (programmability)	NEC	A minimalistic, tailor-made, virtualized operating system to run Click-based middleboxes.
DPDK (Data Plane Development Kit) [36]	Data plane (programmability)	6WIND/INTEL	Set of libraries and drivers for fast packet processing
IO Visor Project [37]	Data plane (programmability)	Linux Foundation	It opens up new ways to develop and share IO and networking functions allowing to contribute to an open programmable data plane for modern IO and networking

			applications.
ONIE [38]	Data (and control) plane (switching)	Cumulus Networks	Open Network Install Environment (ONIE) defines an open “install environment” for bare metal network switches.
OpenContrail vRouter [39]	Data (and control) plane (routing)	Juniper Networks	OpenContrail vRouter is a forwarding plane (of a distributed router) that runs in the hypervisor of a virtualized server. It extends the network from the physical routers and switches in a data center into a virtual overlay network hosted in the virtualized server
OpenFastPath [40]	Data plane (programmability)	OFP Foundation	OpenFastPath is an open source implementation of a high performance TCP/IP stack designed to run in Linux user-space.
P4FPGA [41]	Data plane (programmability)	Cornell	P4FPGA: FPGA made easy
SONiC [42]	Data plane (programmability)	Microsoft	Open source networking switch software that powers the Microsoft Global Cloud.
OpenvSwitch (OVS) [43]	Data (and control) plane (switching)	Linux Foundation	Open vSwitch is a production quality, multilayer virtual switch designed to enable massive network automation through programmatic extension.
Mosaic-5G [44]	Data and control planes (infrastructure platform)	Eurecom	Software-based 4G/5G service delivery platform.
OpenAirInterface [45]	Data and control planes (infrastructure platform)	OpenAirInterface Software Alliance	OpenAirInterfaceTM (OAI) wireless technology platform is a flexible platform towards an open LTE ecosystem that offers an open-source software-based implementation of the LTE system spanning the full protocol stack of 3GPP standard both in E-UTRAN and EPC

2.3 Related Standardization Activities

2.3.1 Overview of Related Standards

The following table identifies Standard Definition Organizations (SDO), Industry Fora (IF) or Other (O) organizations that are relevant to SLICENET. For each of the presented organizations/activities SLICENET should have different approaches depending on the impact of that organizations/activities in SLICENET work.

SLICENET is targeting to consider these organizations as reference for either adoption of solutions (and enhancements where applicable) or proposal of new solutions from the project. In particular, for each organization the collaboration approach is reported as:

- Follow, when SLICENET plans to only be aware of activities and specifications,
- Align, when SLICENET plans to use their solutions as reference baseline,
- Collaborate, when SLICENET plans to actively contribute to the organization activities.

Table 7: Overview of related standard organizations

Organization	Brief description	Type	Approach
3GPP [46]	3GPP follows the network slice concept defined by NGMN. In 3GPP, seven key issues for network slice management have been identified: • Issue 1. Align slice instance related terms. • Issue 2. Creation of the network slice to support services provided by the operator - How the services are requested via the 3GPP management system and how they are facilitated using an NSI. • Issue 3. NSI FCAPS management - How to use FCAPS management for network slicing management. • Issue 4. SON evolution for network slice management - How to apply SON concept for network slicing management. • Issue 5. Orchestration of network slice - What are the components of a network slice and how it is orchestrated? • Issue 6. Shared slice instance management - Management of a network slice that supports multiple services. • Issue 7. Orchestration of Slice across multiple administrative domains.” In the scope of SLICENET, Issues 2, 3, 4, 5, 6 and 7 would be explicitly addressed, corresponding to slice creation (slicing), slice FCAPS management, cognition-based slice management, slice orchestration, and multi-domain slicing. Moreover, 3GPP has outlined a number of technical use cases for network slice management. Those use cases have been analysed in SLICENET, and a selected subset of key use cases that are most relevant to demonstrate SLICENET’s capabilities is elaborated in Section 5 based on a coordination of multiple planes (management, control, data and cognition) in SLICENET.		
3GPP SA1 - Services [47]	Service and feature requirements applicable to mobile and fixed communications technologies.	SDO	Align

3GPP SA2 - Architecture [48]	SA WG2 Architecture is in charge of developing the Stage 2 of the 3GPP network. Based on the services requirements elaborated by SA WG1, SA WG2 identifies the main functions and entities of the network, how these entities are linked to each other and the information they exchange.	SDO	Align
3GPP SA3 - Security [49]	SA WG3 is responsible for security and privacy in 3GPP systems, determining the security and privacy requirements, and specifying the security architectures and protocols. The WG also ensures the availability of cryptographic algorithms which need to be part of the specifications. The sub-WG SA3-LI provides the requirements and specifications for lawful interception in 3GPP systems.	SDO	Align
3GPP SA5 - Telecom Management [50]	SA WG5 will specify the requirements, architecture and solutions for provisioning and management of the network (RAN, CN, IMS) and its services. The WG will define charging solutions in alignment with the related charging requirements developed by the relevant WGs, and will specify the architecture and protocols for charging of the network and its services. The WG will ensure its work is also applicable to the management and charging of converged networks, and potentially applicable to fixed networks. The WG will coordinate with other 3GPP WGs and all relevant SDOs to achieve the specification work pertinent to the provisioning, charging and management of the network and its services.	SDO	Collaborate
ETSI NFV [51]	The ETSI NFV Management and Orchestration (MANO) standard is of primary interest. NFV MANO comprises a set of functional blocks, interfaces and data models which together define a standard way to manage the lifecycle of VNFs and Network Services. This ETSI NFV MANO architecture has featured the management plane of a number of existing 5G projects that leverage the NFV concept, and it is expected that this trend will continue to ensure standard compliance. It is noted that SLICENET would further leverage this architecture in addressing the NFV related management and orchestration issues as SLICENET will require VIM, VNFM and NFVO for the MANO of slices, which are largely based on NFV. ETSI NFV considers network slicing as a kind of use case for NFV MANO principles. Indeed, within the context of NFV EVE Working Group (WG) activities they have started to analyse the gaps between MANO and 3GPP/NGMN standards, in terms of approaches and models. In particular, ETSI NFV does not foresee major enhancements to the MANO functional split and building blocks, while it has defined a set of requirements to enable network slicing in the EVE012. Anyhow the ETSI architectural concepts do not covers the cross-provider management and orchestration of the resources, also without any specificity requirements for 5G security concepts. The ETSI MANO architecture provides in fact the operational stack that enables the orchestration capabilities for end-to-end network services (OpenStack NFV Orchestration) including form a high level perspective the NFV Catalogue (VNF and Network Services descriptors), VNFM (VNF Lifecycle, monitoring and health status check, policy base VNF healing/scaling,	SDO	Follow

	configuration), NFVO (end-to-end service deployment, resource allocation and checks, orchestration through multiple points of presence/domains, service function chaining) and APIs that can be used inside the service provider domain through the OSS/BSS system or the Orchestrator(NFVO), deploying services through the network and possible to the customer site. In particular, the NFVO could help the life-cycle management of the slice-based NSs in SLICENET assuming that a SLICENET service could be mapped to one or more ETSI NFV NS, e.g. leveraging on NS nesting concepts.		
ETSI MEC [52]	Multi-access Edge Computing (MEC) offers application developers and content providers cloud-computing capabilities and an IT service environment at the edge of the network. This environment is characterized by ultra-low latency and high bandwidth as well as real-time access to radio network information that can be leveraged by applications. MEC provides a new ecosystem and value chain. Operators can open their RAN edge to authorized third-parties, allowing them to flexibly and rapidly deploy innovative applications and services towards mobile subscribers, enterprises and vertical segments.	SDO	Follow
ETSI ENI [53]	The ETSI ISG on ENI focuses on improving the network operator experience, adding closed-loop artificial intelligence mechanisms based on context-aware, metadata-driven policies to recognize and incorporate new and changed knowledge, and hence, make actionable decisions. SLICENET aims at aligning the Intelligence functionalities in its logical architecture with the ETSI ENI principles, use cases and requirements. In this direction, SLICENET and ETSI ENI has already started to liaise with the aim of jointly progressing in the definition of architectural principles and features.	SDO	Follow
IETF/IRTF [54]	Several Internet Drafts have been created in IETF/IRTF regarding network slice management. Network slice management functional architecture, proposed by IETF, follows NGMN's recursive slice model and explicitly place a Network Slice Management Function (NSMF) and a Network Slice Subnet Management Functions (NSSMF) to manage end-to-end slices and sub-slices respectively. The NSSMF manages both VNFs via an interface with the NFV-MANO and PNFs directly. Lower level recursive sub-slices are also managed by the NSSMF directly. The architecture also indicates that the NSMF and the NSSMF could be part of the OSS platform and the Network Management/Element Management (NM/EM) systems, respectively. SLICENET reference logical architecture, as presented in Section 6, shares a similar view of recursive service/slice management, where the NSMF is represented by the Service Management (based on the "End-to-End Slice as a Service" vision) and the NSSMF by the Slice Management, which manages recursive sub-slices, respectively. In addition, the different levels of slice control exposure proposed in this Internet Draft [IETF] is also useful in help define the P&P Control in SLICENET.		

IETF SFC [55]	Network operators frequently utilize service functions such as packet filtering at firewalls, load-balancing and transactional proxies (for example spam filters) in the delivery of services to end users. Delivery of these types of services is undergoing significant change with the introduction of virtualization, network overlays, and orchestration. For a given service, the abstracted view of the required service functions and the order in which they are to be applied is called a Service Function Chain (SFC).	SDO	Follow
IRTF NFVRG [56]	The technologies enabling the virtualization of network functions (NFs) are currently in an early stage, and they need researchers to develop new architectures, systems, and software, and to explore trade-offs and possibilities for leveraging virtualized infrastructure to provide support for network functions. The Network Function Virtualization Research Group (NFVRG) will bring together researchers and grow the community around the world in both academia and industry to explore this new research area. Beyond the direct activity through the IRTF collaboration tools it will organize research group meetings and workshops at premier conferences (such as IEEE ICC, IEEE GLOBECOM) and inviting special issues in well-known publications. The NFVRG will focus on research problems associated with NFV-related topics and on bringing a research community together that can jointly address them, concentrating on problems that relate not just to networking but also to computing and storage aspects in such environments. It is hoped that the outcome of the research will benefit research efforts in other groups within the IRTF (and especially the SDNRG) and standardization activities of IETF WGs (like the ones going in SFC). Specific results can also spawn activities via IRTF & IETF BoF meetings and/or provide useful input to other related efforts in the ETSI NFV ISG or other standards bodies.	SDO	Follow
IRTF NMRG [57]	The Network Management Research Group (NMRG) provides a forum for researchers to explore new technologies for the management of the Internet. In particular, the NMRG will work on solutions for problems that are not yet considered well understood enough for engineering work within the IETF. The initial focus of the NMRG will be on higher-layer management services that interface with the current Internet management framework. This includes communication services between management systems, which may belong to different management domains, as well as customer-oriented management services. The NMRG is expected to identify and document requirements, to survey possible approaches, to provide specifications for proposed solutions, and to prove concepts with prototype implementations that can be tested in large-scale real-world environments.	SDO	Follow
MEF [58]	The MEF's Third Network will enable services between not only physical service endpoints used today, but also virtual service endpoints running on a blade server in the cloud to connect to Virtual Machines (VMs) or VNFs. The MEF will achieve this vision by building upon its successful CE 2.0 foundation by defining Lifecycle Service Orchestration (LSO) capabilities and supporting Application Program Interfaces (APIs).	IF	Follow

	SLICENET can take advantage of the MEF LSO principles, interfaces and models for service orchestration and delivery across multiple operators.		
IEEE [59]	NFV and SDN enable rapid networks and services innovation for all participants in the ecosystem. Decoupling network functions from the underlying physical infrastructure using virtualization technologies enables Service innovation. SDN allows programmability of the NFV infrastructure to support the deployment of new network functions in a variety of environments including campus & enterprise networks, data centers, telecommunications networks providers, ISPs, cloud providers and over-the-top (OTT) applications & services providers. NFV enables network capacity and functionality to be decoupled, allowing network functions to be dynamically deployed whenever they are required and wherever they can be hosted.	SDO	Follow
ITU IMT-2020 [60]	ITU has proposed an IMT-2020 (5G) network architecture based on network slice in a draft recommendation [IMT-2020]. The life-cycle of an IMT-2020 network slice instance is managed by coordinating a stack of planes (applications, service, control, and data) and resources via a cross-plane MANO (including slice templates) functional box. The SLICENET reference logical architecture is in line with this multi- and cross-plane network slice management vision.	SDO	Follow
NGMN [61]	NGMN defines a recursive network slice concept and construct model. Firstly, a Service Instance is a run-time construct of an end-user service or a business service that is realised within/by a Network Slice. Secondly, a Network Slice Instance (NSI) is a set of run-time NFs, and resources to run these NFs (an NSI can include zero, one or more sub-network instances), forming a complete instantiated logical network to meet certain network characteristics (e.g., ultra-low-latency, ultra-reliability) required by a Service Instance(s). Thirdly, Network Slice Sub-network Instance (NSSI) is a run-time construct and comprises a set of NFs and the required resources, and Sub-network Instance examples are IMS, HSS etc. Finally, a resource refers to an asset for computation, storage or transport network including radio access, physical or logical/virtual. The concept of slice in SLICENET is in line with the NGMN's view, in particular for what concern the decomposition of slices into multiple interconnected logical constructs for end-to-end service offerings. SLICENET focuses on slice-based services for vertical businesses, and an end-to-end service is realised by joining slices or sub-slices (i.e., sub-network slices) from a single administrative domain or multiple domains based on the "End-to-End Slice as a Service" perspective.	IF	Follow
TM Forum [62]	TM Forum is the global industry association that drives collaboration and collective problem-solving to maximize the business success of communication and digital service providers and their ecosystem of suppliers. Our vision is to help communications service providers (CSPs) and their suppliers to digitally transform and thrive in the digital era. We do this by providing an open, collaborative environment and practical support which enables CSPs and suppliers to rapidly transform their business operations,	IF	Align

	IT systems and ecosystems to capitalize on the opportunities presented in a rapidly evolving digital world.		
ONF [63]	The Open Networking Foundation (ONF) is a non-profit operator led consortium driving transformation of network infrastructure and carrier business models. The ONF serves as the umbrella for a number of projects building solutions by leveraging network disaggregation, white box economics, open source software and software defined standards to revolutionize the carrier industry.	O	Follow

3 Model for Business Roles and Relationships

3.1 Business Modelling Concepts

The justification for work item FS_BMNS in 3GPP SA1 states that the network slice requirements in 3GPP TS 22.261 may be separated from considering the business role models that will apply with slicing. The concept of network slices introduces the possibility to support new business role models in 5G.

This section proposes business modelling concepts that enable the definition of business models and the identification of business roles and eases this task by providing definition of terms and how these terms shall be used to describe the business roles and business models. It contributes to helping the determination of:

- who the stakeholders are,
- which roles each stakeholder plays, and
- how trust relationships among stakeholders are.

3.1.1 Business roles separation

There are mainly three reasons for business roles separation:

- Economic: business roles which are considered users and producers of services could be assigned to different business roles
- Technical: areas of different development speed of technology could be placed in different business roles
- Regulatory: due to regulatory constraints, certain separations of business roles are induced

The duality of the *user/provider* concept pair is maintained for the description of the relationship between the business roles. A *contract* governs the *user/provider* relationship.

3.1.2 Summary of concepts definition

The following concepts are used in the remainder of the section and are based on the definition found in ITU-T Z.600 Annex A – Business Modelling concepts

- *Stakeholder:* A party that holds a business interest or concern in the telecommunications business. A stakeholder who owns one or more business administrative domains.
- *Business administrative domain:* A business administrative domain is defined by the requirements of one or more business roles and is governed by a single business objective.
- *Business role:* The expected function performed by a stakeholder in a telecommunications business environment.
- *Business relationship:* An association between two business roles.
- *Contract:* The context defining constraints for one or more interfaces to operate under.
- *Interface (Reference Point):* The manifestation of a business relationship in the telecommunications system. An interface consists of several related specifications governed by a contract.

3.1.3 Identification of interfaces (reference points)

Each function is atomic and could be offered by a single business. The concept can be used in a virtualised or non-virtualised context. In the context of 5G, such an atomic function will usually refer to a VNF (Virtual Network Function).

The separation of business roles and a clear definition of network function responsibilities enable the functional modularisation of the architecture. Even more, it could imply that the functional modularisation of the architecture should be defined by business models.

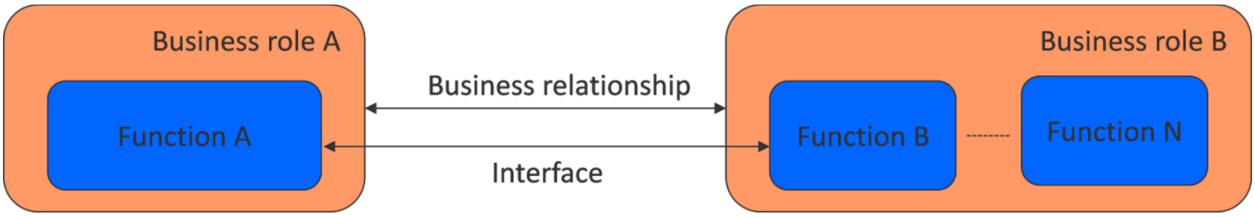

Figure 1: Identification of interfaces

Figure 1 illustrates two business roles, A and B, whereas business role A provides Function A (e.g. VNF A) and business role B provides other functions, such as Function B … Function N. An interface can be identified between Function A and Function B (also called reference point) that allows the exchange of information between the two functions. The information flow is depending on the user/provider assignment of the functions.

The interface is the manifestation of the business relationship between business role A and business role B. In Figure 1 the business roles are independent of which stakeholders assume the business role (see next sub-section).

3.1.4 Mapping of business roles to business administrative domains

Figure 2 illustrates the relationship between business administrative domains and business roles. In particular the following relationships apply:

- Each business role is performed by a business administrative domain
- Each administrative domain can assume one or more business roles
- A contract exists between business administrative domains
- The contract governs the business relationship and the interfaces between the functions performed by the business roles that are assumed by the business administrative domains

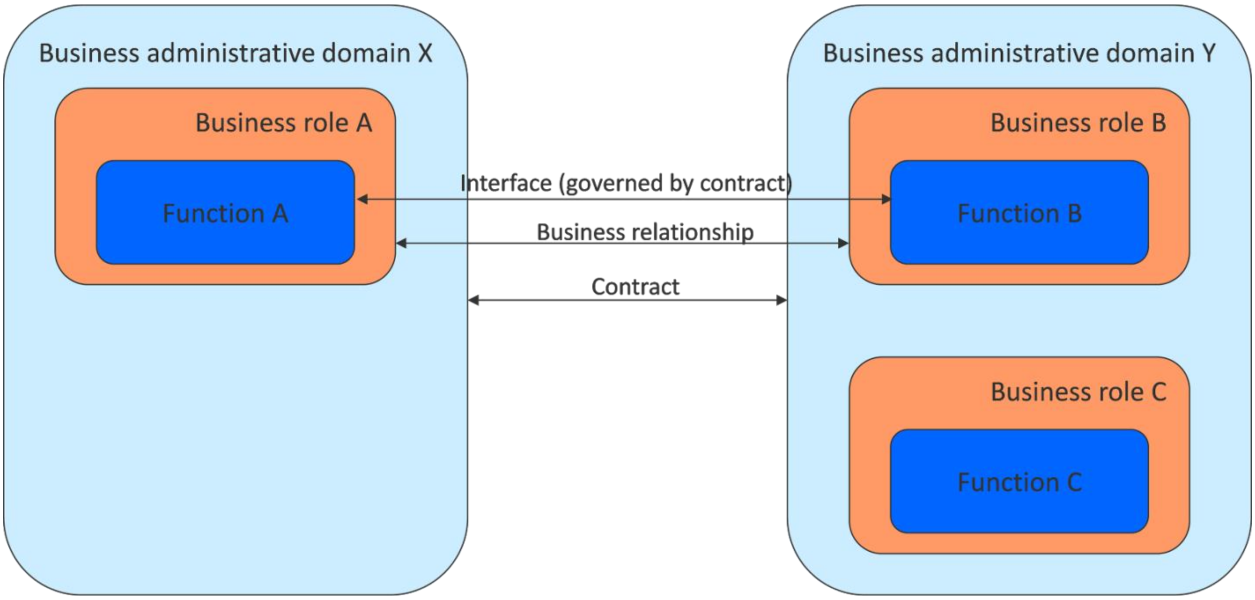

Figure 2: Mapping of business roles to business administrative domains

3.1.5 Usage of Services

One of the business administrative domains will offer service usage to the customer which will be realized as a reference point to the application. Figure 3 expands on the previous concepts and includes the usage administrative domain, which through an interface to Function B, offered by business administrative domain Y, assuming the business role B, consumes the services of Function B.

Note that the interface of Function B (assuming a provider role for Function B) may be the same as that provided:

- to Function A, which lies in the foreign business administrative domain X, or
- to Function C, which is owned by the same business administrative domain Y as Function B, but in the context of a different business role C, as well as
- to the Application in the Usage domain

In its manifestation as an intra-business administrative domain reference point (between Function B and C), it is not governed by any contract, yet it is a useful conformance point regardless of whether it is used inter-domain or intra-domain.

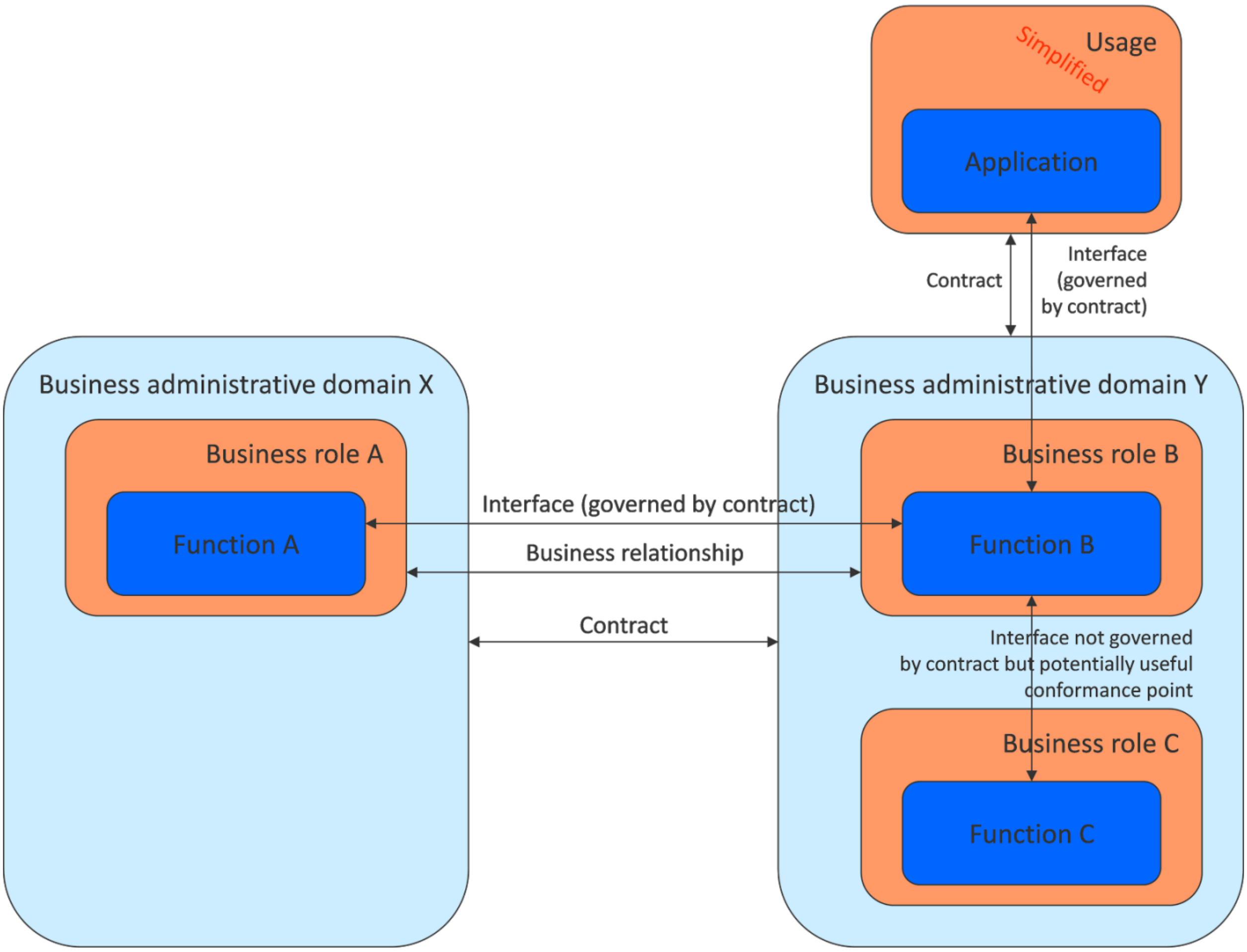
Usage
Simplified
Application
Contract
Interface (governed by contract)
Business administrative domain X
Business role A
Function A
Interface (governed by contract)
Business relationship
Contract
Business administrative domain Y
Business role B
Function B
Interface not governed by contract but potentially useful conformance point
Business role C
Function C

Figure 3: Usage of services

3.2 A Business Roles Model for SLICENET

SLICENET is particularly focused towards the exploration of realistic Use Cases from business verticals, and the project consortium includes a variety of stakeholders that can assume a variety of roles in the business relations involved in providing a service to a final SLICENET customer.

As such, the *a priori* identification and definition of the roles that can be assumed by the various stakeholders is important, as this helps to define the context and boundaries of the solution being addressed by SLICENET.

3GPP has already addressed this matter in [TR28.801] and issued a model identifying the main roles that are expected to be played in a slicing-based ecosystem, where Network Softwarization (SDN, NFV, Cloud) will enable new roles that will most likely result in new business entities and new business relationships:

- Virtualization Infrastructure and Data Center Services may be provided by new specialized business actors
- VNFs are beginning to be marketed and expected to be an important market in the future [VNF market Reference]

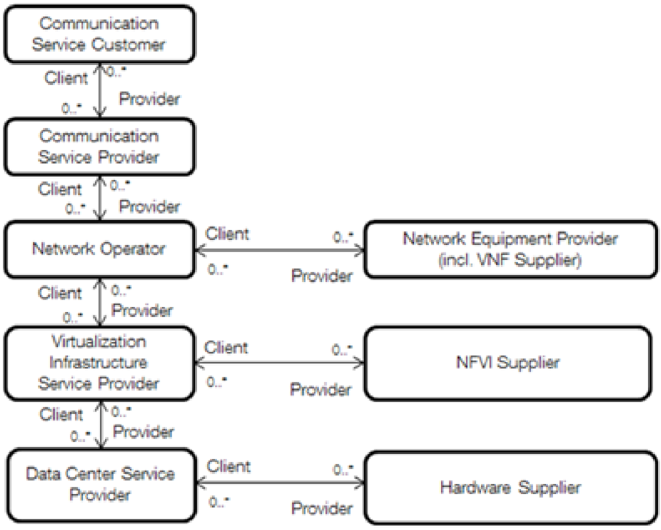

Figure 4: High-level model of roles 3GPP TR28.801

In this model, the Communication Service Customer (CSC) represents the vertical consumer of the services that are offered by a Communications Service Provider (CSP). This is the basic role for our verticals: verticals are today the customers for the communications services that are offered to them by CSPs, who are usually also Network Operators, but not necessarily:

- OTT CSPs offer their services either over a Network Operator or over the Internet Connection Service of another CSP;
- Virtual Network Operators are CSPs that use a Network Operator’s network to provide their own services,

SLICENET is seeking a business roles model that clearly supports the business opportunities created by Network Slicing, namely those that arise from the possibility of a Network Operator offering to a customer:

- a network slice instance as a service, i.e., providing an environment that “is like” an isolated, purpose-built network, that fits to the customer’s requirements.
- the possibility to manage and control the purpose-built network that is offered.
- the possibility to host customer’s virtualized applications and services on the slice infrastructure.

In this scenario, Communication Service Providers may fall short in what can be provided using the resources that a Network Operator may offer.

To address this, SLICENET extends the roles model proposed by [3GPP TR 28.801], by defining a new pair of roles: the **Digital Services Provider (DSP)** and the **Digital Services Consumer (DSC)**, as depicted in Figure 24. The DSP represents the role of a service provider who can take advantage of the aforementioned possibilities, to offer new, wider scoping tailor-made services, and the DSC represents the end client that takes advantage of these new network services. The CSP's role is still part of the model, since both the DSP and the DSC can be clients to more the "traditional" communication services.

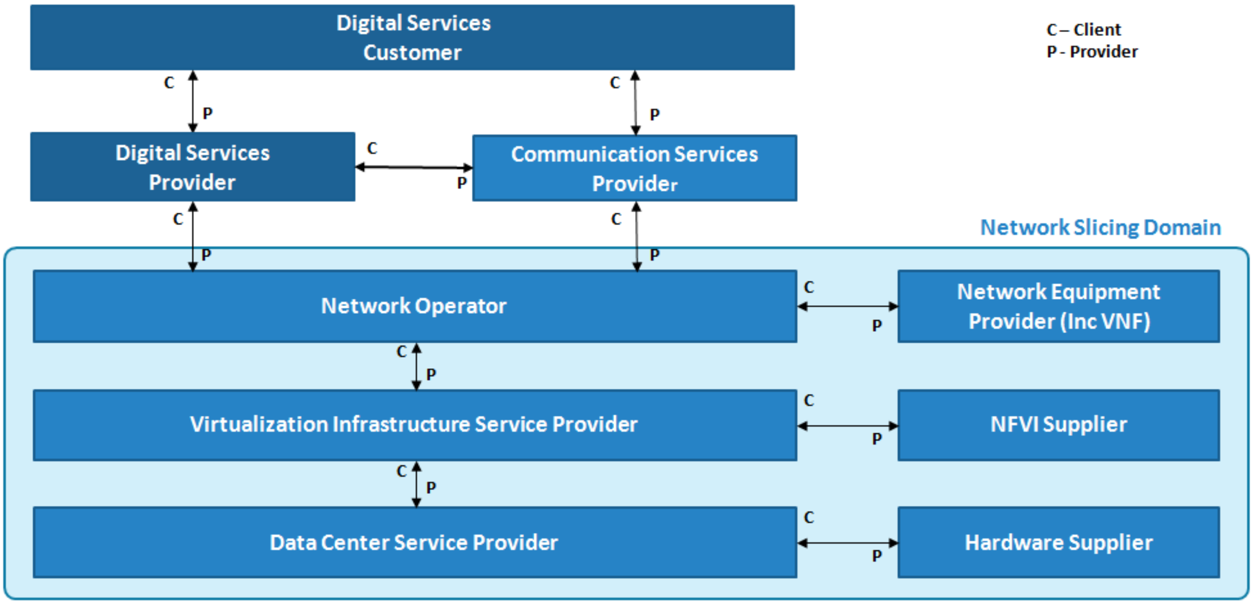
Digital Services Customer
C – Client
P - Provider
Digital Services Provider
Communication Services Provider
Network Slicing Domain
Network Operator
Network Equipment Provider (Inc VNF)
Virtualization Infrastructure Service Provider
NFVI Supplier
Data Center Service Provider
Hardware Supplier
C
P

Figure 5: SLICENET high-level model of roles

It is important to note that this is not a hierarchical model. The relationships between the various roles have only a Client/Provider nature. It is also worth noting that these relationships are not only confined to the ones illustrated here. There may be others, e.g., a DSP may be a client to a Data Center Service Provider who will host his services. Represented here are those relationships that are considered to be mainstream.

Digital Services: Extension to Communications Services, defined in [TR 28.801]. Digital Services encompass Communication services and other services that may be built using the resources offered by Network Operators and other sources. These services may be provided under several categories (B2B, B2C, B2B2x). SLICENET's focus is on B2B2x, namely services that a Network Operator may provide to a Digital Services Provider, who provides their services to a Digital Services Consumer (e.g. a business vertical).

Digital services have a communications component, but they may encompass other aspects into an offer that covers the needs of DSCs, e.g., a Smart Grid Protection Service may include fixed and mobile communications services for protection nodes interconnection, as well as specific protection mechanisms and protocols, adequate to an energy transport operator.

Digital Services Customer: The DSC is the consumer of the Digital Services provided by the Digital Services Provider. Business verticals typically assume this role, and the SLICENET Business Use Cases (i.e. an energy transport Operator, a Smart City grid, and a Medical Emergency Service) fulfil this role under a B2B or B2B2B relationship.

Digital Services Provider: The DSP builds and explores services that are adequate for the needs of Digital Services Customers.

The Digital Services Provider may be a client to a Communications Services Provider, in the sense that he subscribes to Communications Services, but also a client directly to the Network Operator, using

Network Slices as a network support for services that may span across a wider scope than just communications, and offer highly customized services to Digital Services Consumers (typically business verticals).

Network Operator: *Provides network services. Designs, builds and operates its networks to offer such services [sic 3GPP TR 28.801].*

In the particular context of SLICENET, these Network Services will be provided using Network Slice Instances that are exposed as Services (NSaaS – Network Slice-as-a-Service).

The capabilities that an NO exposes may have a wider scope than just Network Services, as the NO will be able to expose aspects of a slice's control layer or to host client's own VNFs.

NOs will be able to expose Network Slices to other NOs, allowing the construction of end-to-end slices that involve more than one administrative domain.

Network Equipment Provider: *Supplies network equipment. For sake of simplicity, VNF Supplier is considered here as a type of Network Equipment Provider [sic 3GPP TR 28.801].*

VNFs will have to go through a specific validation process (not in the scope of SLICENET), but the outcome is essentially the same as for PNFs: VNFs are onboarded to the NO to be used like PNFs do. Nevertheless, the business dynamics for VNFs can be very different.

Virtualization Infrastructure Service Provider: *Provides virtualized infrastructure services. Designs, builds and operates its virtualization infrastructure(s). Virtualization Infrastructure Service Providers may also offer their virtualized infrastructure services to other types of customers including to Communication Service Providers directly, i.e. without going through the Network Operator [sic 3GPP TR 28.801].*

NFVI Supplier: *Supplies network function virtualization infrastructure to its customers [sic 3GPP TR 28.801].*

Data Center Service Provider: *Provides data center services. Designs, builds and operates its data centers [sic 3GPP TR 28.801].*

Hardware Supplier: *Supplies hardware [sic 3GPP TR 28.801].*

3.3 Stakeholders

Business roles are performed by real world entities that may aggregate more than one role. For instance, a "traditional" TelcoA will typically accumulate the roles of a Network Operator and a Communications Service Provider, providing to its customers (business or consumer) the communications services that it builds using the network it typically owns. In this case, TelcoA is a stakeholder assuming two business roles: Network Operator and Communications Service Provider.

For the SLICENET Use Cases, the various roles are assumed by the organizations involved, in stakeholder configurations that are variable. Various roles may be assumed by the same organization and roles may be omitted.

3.4 Business Roles vs. Slicing Management

Figure x+1 highlights a number of business roles under the name "Network Slicing Domain". This is to make clear that, although SLICENET is mostly about Slice Management, some of the business roles identified in this scope are completely unaware of Network Slices. What a Communication/Digital Services Provider expects from a NO that is providing a Network Slice-as-a-Service, is an isolated network with a certain set of features, that he can treat as his own network, obviously under the conditions that the NO defines as a Service Provider.

The Digital Services Customer is even further away from the Network Slice Concept, since he will be using the services exposed by the CSP/DSP. Hence, Network Slicing is a concept that is (as should be) completely opaque to verticals.

It is within the technical domain of the NO and its (direct and indirect) providers that network slicing is important and carries consequences:

Network Operator: It is the NO that explores the Network Slicing principle to provide isolated, custom built Networks to its customers. To be able to do that, the NO will have to require certain capabilities from its providers:

- **Network Equipment Provider:** All Network Equipment, either physical and/or virtual network functions, will have to guarantee the mechanisms for traffic isolation and multi-tenancy;
- **Virtualization Infrastructure Service Provider, NFVI Supplier:** Compute Nodes, Storage and Network will have to provide the isolation mechanisms for building slices
- **Data Center Service Provider, Hardware Supplier:** Data centers and Hardware must meet the requirements needed to implement slicing.

Obviously, to be able to build isolated network slices the network technology must support it. For instance, for mobile generations prior to 5G, the radio network protocols are not the most adequate for building slices at the RAN level.

3.5 Roles and Stakeholders for SLICENET Use-Cases

3.5.1 Smart-Grid Use-Case

The Smart Grid Self-Healing Use Case focuses on communication-based distributed Fault Detection, Isolation and Restoration (FDIR) algorithms implemented on the Medium Voltage (MV) power grid. The self-healing schemes covered by the use case scenarios rely on ultra-reliable, very low latency, peer-to-peer communication between power system Intelligent Electronic Devices (IEDs) installed in remote locations.

The mapping between the SLICENET top-level roles and the Smart Grid Self-Healing use case actors/stakeholders is presented in Table 8.

The actors/stakeholders addressed by the Smart Grid use case with the greater impact in the SLICENET system are the power system operator/Distribution System Operator (DSO), all the communicating power system devices directly or indirectly involved in smart grid self-healing (e.g., protection and control IEDs, Supervisory Control And Data Acquisition (SCADA) system, engineering stations in the control center), the network slice provider and the mobile communications operator.

In the context of the Smart Grid use case, the vertical (typically a power system operator/DSO such as a utility that exploits/owns part of the power system infrastructure) is the Digital Service Customer (DSC) that subscribes the service. The power system devices that are using the 5G network slice(s) may also be considered DSCs, since these are the actors that are actually consuming the network slice resources.

The Digital Service Provider (DSP) delivers digital communication services to the DSC. The DSP can be seen as the interface between the Communication Service Provider (CSP) and the vertical industries and must be able to understand the vertical business requirements in order to be able to provide network slices tailored to DSC specific applications. In the Smart Grid Self-Healing use case this role may be carried out by a mobile communications service provider, by a third party service provider, or by the vertical itself.

The CSP offers communication services within the network operator infrastructure. In this context, the CSP may correspond to a mobile communications service provider or to a network operator (frequently, the mobile communications service provider is the network operator). The CSP may provide a slice catalogue with a set of pre-defined network slices adequate for a range of distinct applications. Further customization to these network slices may be provided to the DSC by the DSP.

If the slice or slices required for the vertical application must traverse networks run by multiple operators, the negotiations with the different network operators should be performed by the DSP or the CSP – not by the DSC. This may prove necessary for the Smart Grid Self-Healing use case in order to guarantee coverage and/or the adequate levels of availability and reliability.

Table 8: Smart Grid use case actor/stakeholder to SLICENET role mapping

SLICENET Stakeholder	SLICENET Role	Smart Grid Use Case Actor/ Stakeholder
Digital Service Customer (DSC)	Uses and consumes the specific 5G SLICENET network slicing service for Smart Grid Self-Healing applications.	• Power system operator/ DSO that subscribes the service; • Power system devices • Power system protection and control IEDs • SCADA system; • Station servers; • Engineering stations.
Digital Service Provider (DSP)	Provides digital communication services, with support from the CSP. The DSP takes into consideration the specific communication requirements demanded by the CSC for Smart Grid Self-Healing applications and provides network slices tailored to the customer's needs.	• DSO • Mobile communications service provider • Third party communication service provider
Communication Service Provider (CSP)	Offers communication services within the network operator infrastructure. May provide a slice catalogue with a set of pre-defined network slices adequate for several applications. Further customization to these network slices may be provided to the CSC by the DSP.	• Mobile communications service provider; • Mobile network operator.
Network Operator (NOP)	Provides core network slicing services. Designs, builds and operates the 5G SLICENET slicing service on top of its core networks.	• Mobile network operator.

3.5.2 Smart-City Use-Case

The Smart-City use-case is included in the big family of Internet of Things network, as a collection of devices and objects, appliances, software, sensors and actuators, communication services and different network connectivities with specific QoS and QoE requirements (delay, jitter, bandwidth, reliability, etc..).

The Smart-City implementation scenarios consists in various solutions as metering (gas, water, energy), remote monitoring and control for different infrastructure elements within the city (parking, lighting poles, e-health, climate, transportation, traffic city congestion, waste management) that will require specific communication services and proper systems performance that must responds to the digital communication customers' needs, in terms of:

- Experienced user performance

- Traffic volume density
- Connection density
- Latency
- Mobility

The SmaLi-5G use case is considered to be as mMTC (massive Machine Type Communication) category from the perspective of the massive number of devices requiring communication services from the digital services providers.

The functional model and business role based on 3GPP TR 28.801and adapted to a telco provider vision regarding the next generation network evolution and services are presented in Figure 6:

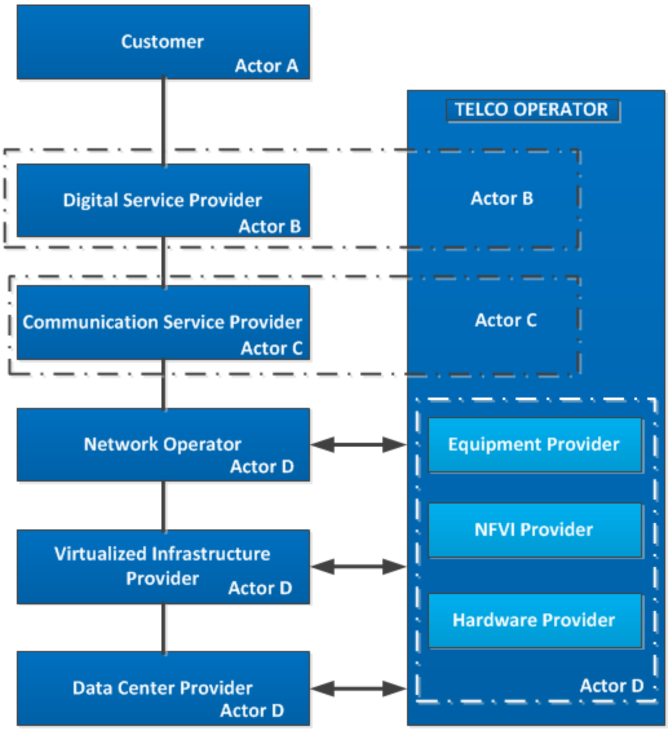

Figure 6: Functional model and business role perspective

Within these approaches, they are created/defined the specific roles and responsibilities and the mapping within the telco operator based on its capabilities. For the sake of clarity, the proposed general model may be adapted to the situation where not all the components are offered by the telco operator, as is the case of a multi-domain end to end network slice for example. The next table shows the relationship between 3GPP versus SLICENET stakeholder/role.

Table 9: 3GPP versus SLICENET stakeholder/role relationship

3GPP Stakeholder	Role in 3GPP	SLICENET Stakeholder	Role in SLICENET
CSC	Use communication services	Consumer (Actor A)	Use and consume specific SmaLi-5G resources in the slice
CSP	Provides communication	DSP (Actor B)	Provides digital communication services, with support of CSP and/or different “network slice” features exposed by the network operator for a

	services		specific SmaLi-5G communication.
		CSP (Actor C)	Service Provider or a network provider that offers communication services, within the network operator infrastructure.
NO	Provides network services	Network Operator (Actor D)	Hardware provider, PNF provider, and software system provider within the telco domain, with responsibilities of designing, building, configuring and operating the service in the “slice” for SmaLi-5G.
VISP	Virtual Infrastructure Provider services	Virtual Infrastructure Provider (Actor D)	NFVI provider, as design in MANO architecture, by designing, building and operating the virtual infrastructure.
DCSP	Data Center Services	Data Center Provider (Actor D)	Provides Data Center services, related to data center operation and integration of specific application.
NEP	Network Equipment Provider	Equipment Provider (Actor D)	The equipment provider in telecom operator infrastructure, physical and/or logical as series of VNFs (e.g. EPC/vEPC)
NFVI	Network Function Virtualization Supplier	NFVI Provider(Actor D)	Network virtualized functions offered for services related to the SmaLi-5G use case.
HW	Hardware provider	Hardware Provider (Actor D)	Dedicated hardware provided (compute, storage, network) as a base support for related NFVI and NEP functionality

In this scenario Actor A is the Smart City vertical that in fact will consume resources and services from the Actor B. Actor A will not be aware that he is using a Network Slice

Actor B is the Digital Service Provider (DSP) that plays the role of an umbrella for services to be offered to the Customer. It takes full responsibility role in the relation with the customer and may translate the Customer service requirements into network requirements, including the functions of designing and building the communication services Actor C is the Communication Service Provider (CSP), generally a telco operator that is in charge with operating the services and also defining the services and network architecture that will be exposed to actor B, including design, build and run phases. Actor D is the NO that provides services by operating its own network and takes responsibility in the design and management of the network. Actor C may interoperate and require network resources from multiple Actors D.

Actor D is the Virtualized Infrastructure Provider (VIP) offering virtual services through the virtualization layer.

Actor D is the Data Center Provider (DCP) that provides data center services, related to the hardware capabilities existing in the data center (x86 servers, layer 2 devices, physical data center elements).

In the SmaLi-5G Actor D is the Telco Provider.

3.5.3 eHealth Use-Case

From an analysis of the business modelling, as presented in the 3GPP TR, and in the context of SLICENET, a number of roles and stakeholders from the eHealth Use-Case can be mapped and identified.

Firstly, a recap of the eHealth scenario: The scenario for the ultrahigh definition video health use case begins with the continuous collection and streaming of patient data, when the emergency ambulance paramedics arrive at the incident scene, and can be summarised by the figure below.

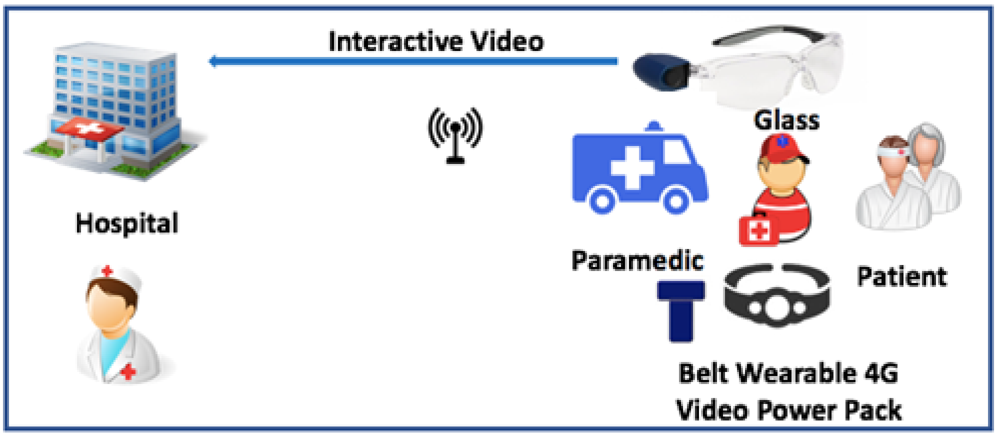

Figure 7: eHealth Use Case scenario (Source RedZinc, Q4Health Project)

3GPP TR 28.801 (pg. 16) describes generic roles and stakeholders for the management and orchestration of a network slicing service, and suggests that some of these roles can be performed by a single stakeholder or several.

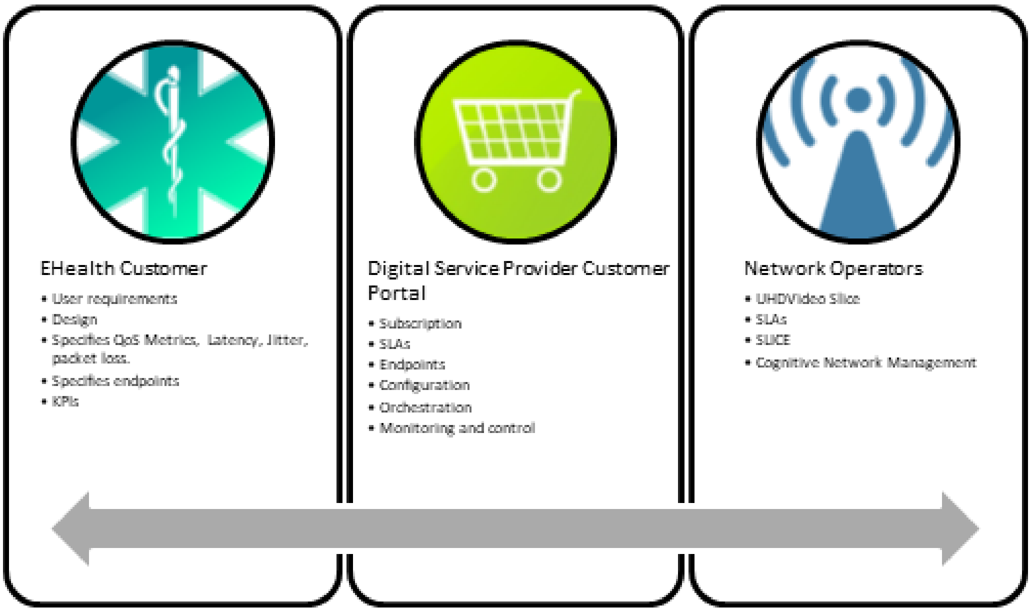

Figure 8: eHealth Use Case Roles and Stakeholders

Figure 8 above provides an overview of the roles and stakeholders, as conceived by the SLICENET eHealth Use Case, and the stakeholders from the 3GPP specification can be mapped to the eHealth Use Case, as presented in Table 10.

Table 10: Mapping the 3GPP TR28.801 stakeholders to SLICENET

3GPP Stakeholder	SLICENET Role	SLICENET Stakeholder
Communication Service Customer (CSC)	Uses and consumes the specific eHealth 5G SLICENET network slicing service.	SLICENET eHealth Customer: • Hospital clinicians, • Emergency services, • Paramedics • Patients
Communication Service Provider (CSP)	Provides a specifically tailored 'one-stop shop' eHealth 5G SLICENET network slicing service. Designs, builds and operates this specific eHealth network slicing service.	SLICENET eHealth 'one-stop shop' Digital Service Provider
Network Operator (NOP)	Provides core network slicing services. Designs, builds and operates the 5G SLICENET slicing service on top of its core networks.	SLICENET Network Operator
Virtualization Infrastructure Service Provider (VISP)	Provides virtualized network slicing infrastructure services. Designs, builds and operates network slicing virtualization infrastructure(s). Virtualization Infrastructure Service Providers may offer their virtualized network slicing infrastructure services to the SLICENET Digital Service Providers directly, i.e. separately from the core Network Operator.	SLICENET Digital Service Provider and/or SLICENET Network Operator
Data Centre Service Provider (DCSP)	Provides data centre services, such as Cognitive Network Management, as part of SLICENET Digital Services. Designs, builds and operates the SLICENET data centres.	SLICENET Digital Service Provider
Network Equipment Provider (NEP)	Supplies network equipment. For sake of simplicity, VNF Supplier is considered here as a type of Network Equipment Provider.	SLICENET Network Equipment Provider
NFVI Supplier	Supplies network function virtualization infrastructure to the Digital Service Providers	SLICENET Digital Service Provider
Hardware Supplier	Supplies SLICENET eHealth specific hardware.	E.g. ATSR have patented Acetech™ technology for Emergency Vehicle Control Systems, and who would supply that hardware either to the Digital Service Providers or directly to the emergency services customers

4 Architecture Considerations

4.1 Contextualization

The architectural concepts within the 5G networks end-to-end developments face a series of interpretations, starting from the technical use cases and possible technology implementations to the impressive sets of capabilities that should be exposed through a more pragmatic and easy-to-use methodology to the end consumers, being them mostly vertical industry players.

The general approach is that consumers, service owners, digital service providers, slice owners, and also the vertical's services owners are not aware of the contextualization of the low-level 5G technologies. The 5G infrastructure provides a series of capabilities that in the end offers to the customers services configuration capabilities, functionality, SLA, KPIs, availability, security and reliability, all abstracted, aggregated and exposed as a network slice construct.

The SLICENET logical architecture is defined as a SBA (software based architecture) approach including several key elements in the 5G ecosystem with various general functions as monitoring (service monitoring, slice monitoring etc.), intelligence (cognition, analytics, aggregation etc.), orchestration (security management, service, slice, resources and infrastructure orchestration etc.), control (slice provisioning, services optimization etc.) and data plane resources.

5G architectures introduce new concepts and new components that act as a valuable chain of functions and enablers for the type of services. The following sections provide an overview of those architecture principles that form the technological ground of the SLICENET framework: network slicing, One-Stop API, Plug & Play control, Cognition, Cross-Plane Orchestration, Multi-Domain network slicing.

4.2 Network Slicing

Network Slicing has definitely emerged as the key enabler in the paradigm shift from 4G to fully NFV/SDN-enabled 5G era. 5G mobile systems will in turn enable network operators and service providers to host vertical industry segments by introducing new services and enhance business collaborations between providers and customers at large. From a technical perspective, network slicing opens the opportunity for the creation of a new ecosystem for delivering customized and cost-efficient services to vertical segments.

In this context, following the visions from many leading 5G standardization bodies and initiatives, SLICENET targets a slicing framework for provisioning flexible, cost-efficient, scalable and tailored services in software-networking based 5G networks.

Vertical segments often have diverse and sometimes conflicting needs, which pose strict and challenging requirements to network operators and service providers on their control and management frameworks for the provisioning of suitable slices and services. Customization is therefore a key aspect for 5G network slicing to offer tailored services, and requires high degrees of programmability and flexibility of the slicing framework. SLICENET considers the involvement of vertical industry as a pillar for network slicing, and implements a "Vertical-In-The-Loop" approach to address the requirements of heterogeneous use cases and services at different stages and levels. First, involving the vertical partners in the consortium in the SLICENET framework design to gather and match specific needs and requirements. Second, and most important, targeting a mechanism for tailoring the provisioning of specific slice instances to the vertical needs by customizing slice offers for the given vertical customers. Moreover, vertical end-to-end services will benefit from a large coverage area across different operators and providers, as well as technical and administrative domains. In line with this, SLICENET builds on the premise that slice and service deployments need to be able (when required) to span across different domains, and therefore will be providing new automated mechanisms for provisioning end-to-end slices targeting a tight coordination of multi-domain control, management and orchestration planes. As a consequence, interoperability is key in SLICENET, and is enabled by the adoption of standard APIs and standard control and management architectures (e.g. at the SDN and NFV level) when applicable.

The multi-domain and truly end-to-end approach poses additional challenges and requirements in terms of heterogeneous resources to be managed and controlled by the SLICENET framework when building and provisioning end-to-end slices. Resources and service components from multiple providers have to be combined dynamically to offer augmented services for specific verticals, users, scenarios and environments. At the same time, resources have to be shared by multiple customers whilst guaranteeing isolation of vertical-tailored slices. This flexible approach for service provisioning, delivery and management is enabled in SLICENET by a combination of powerful SDN/NFV abstractions and resource awareness. Heterogeneous physical and virtualized resources (e.g. network, compute, storage), as well as service components like network functions (either physical or virtual) are abstracted at different layers and across domains following the SDN and NFV principles to hide the infrastructure complexity to the vertical actors and customers. On the other hand, the awareness of resource and service components capabilities, availabilities and logics allows the SLICENET framework to employ programmable SDN and NFV functions as well as programmable data planes in support of customized and vertical-tailored slice control, management and operation.

4.3 One-Stop API

One-Stop API is one of the key SLICENET novelties and is conceived as the main entry point of the slicing framework. It is the enabler for the engagement of verticals into slice design and provisioning, and it reflects the overall outcome of the multi-domain slicing concept as this is tailored to the requirements for the support of “Verticals-in-the-loop” concept.

The diversity of functional, performance and security requirements that may relate with a vertical, necessitates the support of a level of abstraction offered through the One-Stop API. The abstraction aims at enabling verticals express accurately the particular communication/service/application requests that are considered important for the delivery of the end-to-end functionality. The abstraction builds on top of a layered view of the architecture in terms of services utilising NSIs, the contained NSSIs and subsequently of the NFs that are required for the provision of each NSSI, with this layering spanning across domains. The overall view exposed via the One-Stop API aggregates the NSI/NSSI/NF offerings as a pool of selectable features that can be identified into a service creation request. These selectable features are expected to be offered in a way that allows verticals easily identify those aspects that are dictated by the expected requirements from the slice based service.

As there might be interdependencies and even conflicts among features, aggregation concepts should apply on the basis of identified slice templates/blueprints. For example, a vertical may define a specific service/application characteristic as a quite high level requirement. In case the vertical does not require to deal with the fine grained details about how this service requirement is decomposed in terms of interconnections and fine tuning among underlying components (e.g., among NSI/NSSI/NF, or the programmable data path), the One-Stop API should allow the selection of the service/application characteristic without requiring from the vertical to intervene - if not requested - with the underlying details. On the other hand, if there is need for fine tuning of the underlying interdependencies, this should be also possible in the context of the authorisation and the privileges with which a One-Stop API user is related.

Depending on the vertical role and apart from the fine tuning of a template/blueprint instantiation details, the option for building a new template from scratch should be also enabled through the One-Stop API. Optionally, new templates might be shareable so that these can be requested by other stakeholders. Having the service requirements identified by verticals, either from a high level perspective or via a more fine grained approach, the One-Stop API should decompose those requirements in the form of a set of service provisioning workflows as defined by the involved slice provisioning templates.

Embracing inter-domain plug and play options and the fact the SLICENET platform requires multiple entries - at least one per administrative domain - the One Stop API should cater for both vertical and horizontal management. The vertical management is mainly intended for the use by application and use case verticals for the selection of offered services, whereas the horizontal management space is intended for administrative roles that should be offered the possibility to provide service definitions based on the clearly defined inter-domain ecosystem as this is projected in terms of available intra and inter domain slice offerings.

4.4 Plug & Play

The innovative SLICENET "Vertical-In-The-Loop" approach envisages a truly customized runtime control, management and operation of end-to-end slice instances in support of vertical-tailored services. One of the key enablers of the "Vertical-In-The-Loop" runtime approach is the SLICENET Plug & Play control, which provides an innovative combination of customized control functions, APIs and tools to enable verticals to even plug their own control logics and specialize their slices according to their needs, offering significantly enhanced degree of flexibility for tailored services to end users.

The SLICENET Plug & Play control will indeed provide a new flavour of customization to end-to-end slice instances, at two main levels. First, it will enable from a slice provider perspective to activate all those specific per-slice SDN and NFV control functions needed to accommodate the vertical requirements, in terms of network functions composition, as well as performance and QoE. Second, from a slice consumer point of view, the SLICENET Plug & Play will allow the verticals to further control their slices and services by plugging their own control and management functions, enabling the deployment of specific 5G services in a truly customisable, dynamic and scalable way.

Moreover, as a design principle, the innovative Plug & Play control paradigm relies as much as possible on open and standard SDN and NFV APIs, aiming to enable a higher degree of extensibility of the SLICENET framework offering the possibility to third parties in general to plug their own control logics and functions.

From a technical perspective, the SLICENET Plug & Play control is not limited to the customization of slice control at the vertical space only. Indeed, in addition to the Plug & Play control exposed to the verticals, the multi-domain interactions across providers for building and composing end-to-end slices will be also exploiting Plug & Play functions. Assuming that end-to-end slices will always be offered to verticals by a single provider (which in turn can trade and negotiate part of the slice resources and network functions with other providers), a multi-provider (i.e. multi-domain) slice instance can be considered as fragmented into several single provider sub slices (see Figure 9). When provisioned, an end-to-end slice instance exposes a set of Plug & Play control functions and primitives to the vertical according to its needs and requirements. This Plug & Play is the result of the combination and abstraction of the Plug & Play control functions that each provider contributing to the end-to-end slice is offering and exposing.

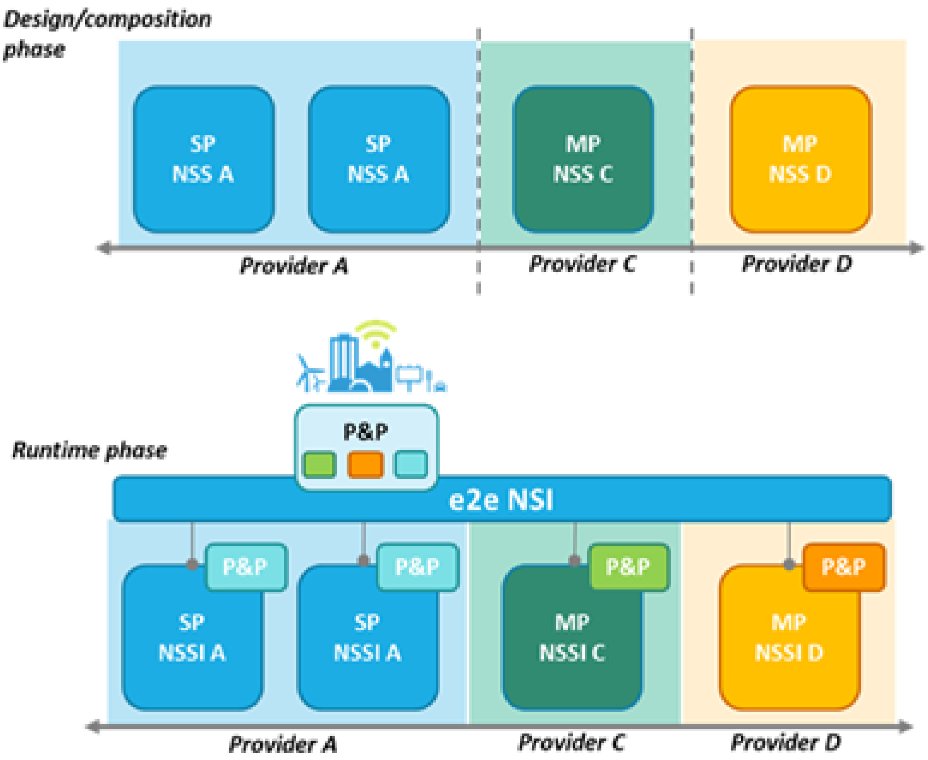

Figure 9: SLICENET Plug&Play approach in multi-provider scenarios

4.5 Cognition (Proactive Management)

SLICENET envisions an intelligent cost-effective network management, control, and orchestrations framework that can cope with the scale and pervasiveness of 5G networks, while minimizing human intervention. Autonomous management and control is evolving from self-healing, self-optimizing and self-configuring automation to artificially intelligent systems that can outlearn their current knowledge and apply newly gained wisdom to achieve network goals through self-decision-making. Network management in SLICENET must handle the unprecedented complexity allowed by the flexibility, pluggability, and hierarchical composition capabilities that 5G enables. It is no longer possible to prescribe what needs to be done per every conceivable system state, as there are too many options and heterogeneous cases. Thus, SLICENET proposes cognitive network management that utilizes machine learning to understand the network behaviour and proactively steer it towards its desired state. SLICENET advocates a declarative rather than an imperative approach to network management, where cognition is used to learn the best actions to achieve declared goals; moreover, the goals are abstracted in a user-centric manner to utilize the full potential of 5G slicing and fulfil the "Verticals-in-the-loop" SLICENET vision. SLICENET's QoE-oriented design targets to improve/optimise the perceived quality of user-facing slice-based/enabled services and applications in the verticals' businesses, utilizing a complete cognition-driven, intelligence-enabled control loop.

The technical approach of the SLICENET cognition framework is shown in Figure 10. Cognitive management is achieved through enhancing an autonomous computing control loop with machine learning capabilities. Monitor-Analyze-Plan-Execute (MAPE) is a common model to describe an autonomous computing loop, where the monitoring component obtains data from the controlled system through sensors, data is then transformed into information through analytics and optimization, self-decisions are converted into concrete execution plans, and finally the execution component orchestrates the plan via actuators. The MAPE loop is governed by policy and knowledge (K); the MAPE components utilize the knowledge-base for all aspects of operation, including configuration, rules, processes, available actions, goals, etc. In SLICENET the assumption is that the knowledge cannot be statically configured to cover all possible states, especially in cross-domain topologies; however, *it can be learned*. The SLICENET cognition framework combines Machine-Learning (ML) and advanced optimization to dynamically update the knowledge-base with newly gained wisdom and continuously refine the MAPE loop to achieve end-to-end QoE. An intelligent Policy Framework governs this dynamic process to ensure end-to-end system behaviour complies to all rules and restrictions.

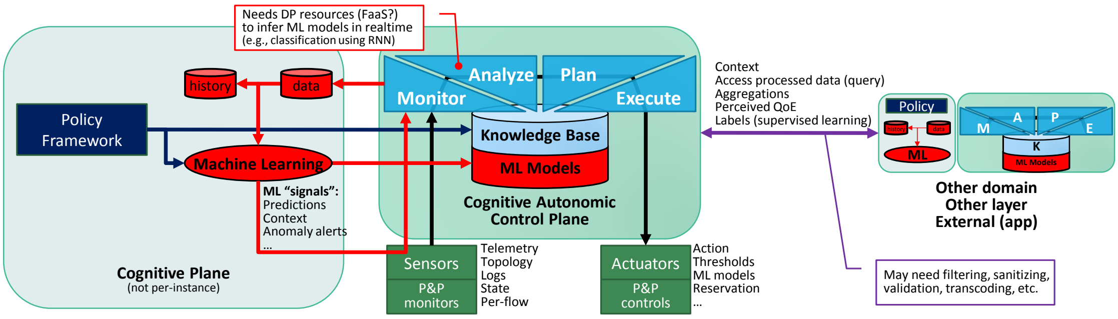

Figure 10: SLICENET Cognition Framework

To achieve full system comprehension, SLICENET sensors collect and aggregate multi-dimensional data, including state, logs, topology, real-time telemetry, and network flows; moreover, Plug & Play sensors allow for dynamically adjusting the collection to achieve QoE goals with minimal overhead. SLICENET actuators combine common tools, such as threshold setting, resource reservation, and direct controls, with machine leaning control models, such as Recurrent Neural Network (RNN) models. Machine learning algorithm can also act as advanced sensors, generating ML "telemetry signals", such as predictions, alerts, and context. SLICENET takes a services approach to obtain the resources needed for the machine learning life-cycle; the learning phase is realized on enterprise-cloud-based clusters that implement data warehouses and ML engines, while the inference phase is realized inside the network by applying a function-as-a-service paradigm.

Vertical and multi-domain slicing is realized through interaction of multiple cognitive MAPE-K loops; these exchange not only raw data but also information (e.g., queries to persisted processed and aggregated data), learnt insights (e.g., context and QoE perception), and wisdom (e.g., learnt models). SLICENET cognition is based on collaboration between cognitive loops, including external entities, such as user behaviour modelling, to maintain end-to-end slice goals. In particular, while unsupervised context learning is possible, sharing context labels can improve the learning quality, and sharing the context itself allows better QoE optimisations. The information exchange between loops is governed by policies to make sure data properly filtered, validated, and sanitized as it crosses domain and/or hierarchy boundaries. This multiple-loop approach also eliminates the need to transfer (big-)data between domains, which is prohibitive both from resource consumption and privacy perspectives.

4.6 Cross-Plane Orchestration

The purpose of the Cross-Plane Orchestration is to provide a set of coordination functions across several logical layers and constructs (i.e. service, slice, resource, and infrastructure) with the aim orchestrating the provisioning of end-to-end slices.

The cross-plane orchestration in SLICENET makes use of recursive orchestration abstractions that hide complex operations in multiple administrative domains (i.e. more than one network provider), multiple technologic domains (i.e. RAN, WAN and datacentres), and multiple layers (i.e. services, slices and resources). The advantage of orchestration abstractions is twofold: i) to allow recursive service composition to establish an end-to-end slice across multiple administrative domains, ii) and to hide workflows of multiple operations in a single atomic operation.

Regarding the first advantage, when a customer requests a service from a network operator A, the latter might need to request services from other network operators to fulfil the service. With that in mind, the network operator A may request a network operator B for a service that will be used in conjunction with the part of the service that is instantiated on the network operator A managed infrastructure. From the point-of-view of the client, he is unaware of the service provided by network operator B and he only needs to interact with network operator A, who is the one he has a business relationship. In turns, network operator B could also request services from other network operators to fulfil the service requested by network operator A (see Figure 11).

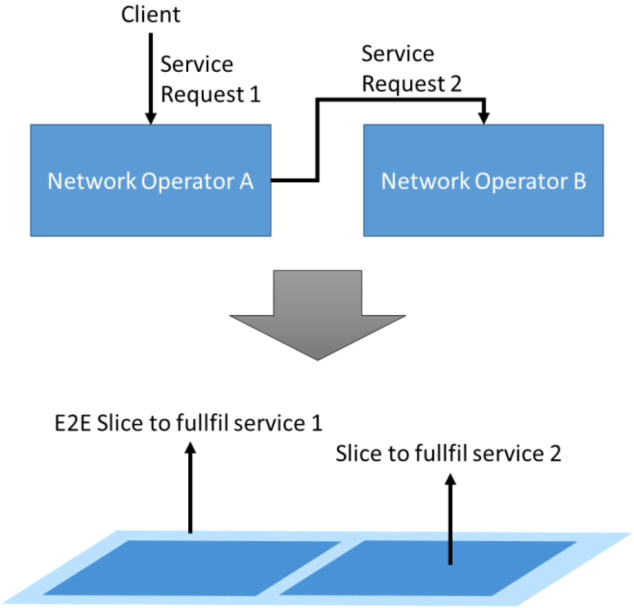

Figure 11 Recursive Service requests to instantiate end-to-end slices

As to the second advantage, the process of scaling a slice can be seen as a single operation when seen from the northbound interfaces of the slice orchestrator, but in reality, this single operation is translated to multiple operations. This decomposition depends on the number of associated resources deployed across multiple domains. Another example may be the case of expanding the geographical area covered by the RAN, this expansion might lead to the necessity of increasing the resources on the core network side and in turns associated network services might also need to scale out. This scaling procedures in different domains is very complex when viewed as whole, but by dividing the problem by layers each orchestrator can focus on a smaller part of the process.

One implicit advantage of this orchestration abstraction is that it simplifies the orchestration of sub-slices deployed across multiple administrative domains to establish a single end-to-end slice. Network operators are very conservative in exposing their infrastructure, therefore it is not possible for one network operator to directly manage the infrastructure of another network operator even if the latter is providing a service to the former. By establishing these abstractions, which are able to hide the implementation of a given service, it is possible for a single network operator to effectively manage a service that is built by components owned by other network operators with limited functional impact.

In SLICENET two types of orchestration are defined: horizontal orchestration and vertical orchestration (see Figure 12). It is called horizontal orchestration when it refers to multiple domains, either administrative or technologic, and the orchestration is performed within the same layer or for the purpose of the same logical construct, e.g. for an end-to-end slice two network operators may perform horizontal orchestration by using the service layer as the common layer. Additionally, it is called vertical orchestration when it refers to multiple layers, e.g. a single network provider performs vertical orchestration when deploying a service instance where the service is composed by one or more slices and each slice is composed by multiple resources.

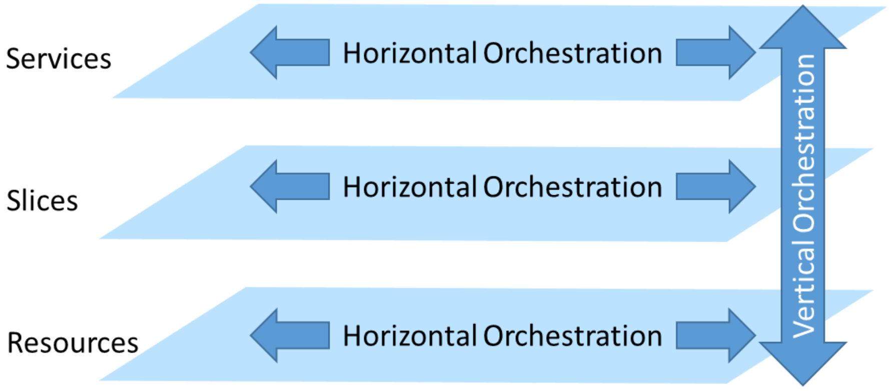

Figure 12 Horizontal and Vertical Orchestrations

Given the three layers considered in SLICENET, i.e. services, slices and resources; three layers of orchestration are therefore considered the building blocks of the Cross-Plane Orchestration: service orchestration, slice orchestration and resource orchestration.

5 Technical Use-Cases & Requirements

In order to better explain the overall scope of the SLICENET system, a set of technical use-cases, inspired on the use-cases defined in 3GPP 28.801, are described in this chapter of the document.

The technical use cases are grouped so to explicitly address the identified Slicing key issues as described in section 2.3, namely:

- slice creation,
- slice configuration,
- slice FCAPS management,
- Self-Optimized Network (SON) applied to Slices,
- multi-domain slicing,
- customization of slice management exposure (Plug&Play) and
- cognition-based slice management

Besides helping on defining the scope of the SLICENET system architecture, the described technical use-cases are also very important to identify the technical requirements that must be addressed during the architecture phase. The set of technical requirements are listed in Annex A of this document.

5.1 Creation of Network Slice Instance

5.1.1 Create Network Slice Instance with (shared) Network Slice Subnet Instance

Table 11 reports the sequence of actions required for the execution of the **creation of a Network Slice Instance with (shared) Network Slice Subnet Instance** Technical Use Case (TUC).

Table 11: Create Network Slice Instance with (shared) Network Slice Subnet Instance Steps

Step	Impacted Architecture Plane	Description
1	Management Plane	Create NSI by selecting from available templates and specify configuration options as presented by the template
2	Management Plane	Identify related NSSs as indicated by NS template
3	Management Plane	Identification of NSSs location based on create request parameters relating to location
4	Management Plane	Decomposition of NS descriptor into NSS descriptors and calculation of configuration parameters per NSS
5	Management Plane	Check if any existing NSSI can be re-used and if there is no conflict between existing and additional configuration
6	Control Plane	Use the NSS descriptor to trigger the NSSI creation by mapping into actions towards subnet interfaces (iteration)
7	Control Plane	NSSI instantiation is ordered to management plane
8	Management Plane	Orchestration of network services takes place to instantiate and configure the NFs and infrastructure

9	**Data Plane**	NSSI NFs and related network links are created

5.1.2 Create end-to-end NSI across multiple network segments

Table 29 reports the sequence of actions required for the execution of the **creation of an end-to-end Network Slice Instance (NSI) across multiple network segments** TUC.

Table 12: Create end-to-end NSI across multiple network segments Steps

Step	**Impacted Architecture Plane**	**Description**
1	**Management Plane**	Capabilities of data plane are exposed to upper layers
2	**Control Plane**	Control specific functions are exposed to management plane
3	**Management Plane**	Create NSI order is received in terms of template and specify configuration options
4	**Management Plane**	Identification of the network elements and functions to be configured and deployed across the different infrastructure segments
5	**Management Plane**	Orchestration of the multiple controllers for NSI/NSSI configuration takes place
6	**Control Plane**	NSSI configuration is triggered
7	**Data Plane**	NSSI NFs and related network links are created
8	**Control Plane**	Inter-NSSI connectivity is triggered to compose end-to-end slice
9	**Data Plane**	Network paths interconnecting NSSIs are created

5.2 Network Slice Instance Change Capacity

5.2.1 Network Slice Instance Change Capacity (OSA ordered)

Table 13 reports the sequence of actions required for the execution of the **Network Slice Instance Change Capacity upon orders received by One-Stop API** TUC.

Table 13: Network Slice Instance Change Capacity (OSA ordered) Steps

Step	**Impacted Architecture Plane**	**Description**
1	**Management Plane**	Change capacity order from One Stop API
2	**Management Plane**	Translate order into re-configure existing NSSI/NF or add/remove NSSI/NF
3	**Management Plane**	Orchestrate change action and identify NSSIs to update
4	**Control Plane**	Trigger the NSSI updating by taking into account P&P rules

5	**Control Plane**	NSSI orchestrates change action and identify NFs to re-configure or add/remove
6	**Control Plane**	If any NF is shared and there is anticipated negative impact on other NSSIs, then Change Capacity order is refused and rollback is orchestrated for NSSI
7	**Management Plane**	Otherwise, either orchestration to add/remove NFs or NFs re-configuration takes place
8	**Data Plane**	New NFs are added while previous allocated NFs are re-configured or removed
9	**Data Plane**	Network links interconnecting configured NFs are updated or removed

5.2.2 Network Slice Instance Change Capacity (Cognitive ordered)

Table 14 reports the sequence of actions required for the execution of the **Network Slice Instance Change Capacity upon orders received by Cognitive** TUC.

Table 14: Network Slice Instance Change Capacity (Cognitive ordered) Steps

Step	**Impacted Architecture Plane**	**Description**
1	**Intelligence Plane**	Change capacity order from Cognitive for NSI
2	**Management Plane**	Translate order into re-configure existing NSSIs/NFs or add/remove NSSIs/NFs
3	**Management Plane**	Orchestrate change action and identify NSSIs/NFs to update
4	**Control Plane**	Trigger the NSSI updating
5	**Control Plane**	NSSI orchestrates change action and identify NFs to re-configure or to add/remove
6	**Control Plane**	If any NF is shared with other NSIs and there is anticipated negative impact on any NSI, then Change Capacity order is refused and rollback is orchestrated for NSSI
7	**Management Plane**	Otherwise, either orchestration to add/remove NFs or NFs re-configuration takes place
8	**Data Plane**	New NFs are added while previous allocated NFs are re-configured or removed
9	**Data Plane**	Network links interconnecting configured NFs are updated or removed

5.3 Network Slice Instance Activation

5.3.1 Network Slice Instance Activation (OSA ordered)

Table 15 reports the sequence of actions required for the execution of the **Network Slice Instance Activation upon orders received by One-Stop API** TUC.

Table 15: Network Slice Instance Activation (OSA ordered) Steps

Step	Impacted Architecture Plane	Description
1	**Management Plane**	Activation order from One Stop API
2	**Management Plane**	Translate order into activation of configured NSSI/NF/TN
3	**Management Plane**	Orchestrate activation action and identify NSSIs to activate
4	**Control Plane**	Trigger the NSSI activation
5	**Control Plane**	NSSI activation action is orchestrated and Network Functions / Transport Nodes to activate are identified
6	**Management Plane**	The NSSI orchestrator manages the sequence of NF/TN activation and their state is updated accordingly
7	**Data Plane**	Allocated NFs / TNs are activated
8	**Data Plane**	Network links interconnecting NFs are activated

5.4 NSI FCAPS management

5.4.1 Network Slice Instance Fault Management

Table 16 reports the sequence of actions required for the execution of the **Network Slice Instance Fault Management** TUC.

Table 16: Network Slice Instance Fault Management Steps

Step	Impacted Architecture Plane	Description
1	**Data Plane**	Fault generated by NFs and/or Infrastructure
2	**Management Plane**	Fault reaches NFs management and /or Infrastructure management (VNFM, EMS, VIM, …)
3	**Management Plane**	Fault is notified to all concerned NSSIs Fault Mgr
4	**Management Plane**	Fault is notified to all concerned NSIs Fault Mgr
5	**Intelligence Plane**	Fault is notified to Cognitive for all concerned NSIs
6	**Management Plane**	Update NS status and network resources inventory

7	**Management Plane**	Fault is exposed to Slice consumer

5.4.2 Configuration management

5.4.2.1 Configuration management supporting network slice (OSA ordered)

Table 17 reports the sequence of actions required for the execution of the **configuration management of network slices made of RAN NFs upon orders received by the One-Stop-API** TUC.

Table 17: Configuration management supporting network slice (OSA ordered) Steps

Step	Impacted Architecture Plane	Description
1	**Management Plane**	RAN configuration order from One Stop API for NSI
2	**Management Plane**	Identification of RAN NSSIs to reconfigure
3	**Management Plane**	Configuration module triggers RAN NSSI specific configuration actions
4	**Management Plane**	RAN NSSI Configuration Management identifies what to configure: radio interface parameters or RAN virtualized component to deploy
5	**Management Plane**	Management of identified configurations
6	**Data Plane**	Radio link parameters are set or NF is reconfigured or RAN virtualized component is deployed

5.4.2.2 Slice specific information configuration for CN (OSA ordered)

Table 18 reports the sequence of actions required for the execution of the **configuration management of network slices made of Core Network NFs upon orders received by the One-Stop-API** TUC.

Table 18: Slice specific information configuration for CN (OSA ordered) Steps

Step	Impacted Architecture Plane	Description
1	**Management Plane**	RAN configuration order from One Stop API for NSI
2	**Management Plane**	Identification of RAN NSSIs to reconfigure
3	**Management Plane**	Configuration module triggers RAN NSSI specific configuration actions
4	**Management Plane**	RAN NSSI Configuration Management identifies what to configure: radio interface parameters or RAN virtualized component to deploy
5	**Management Plane**	Management of identified configurations
6	**Data Plane**	Radio link parameters are set or NF is reconfigured or RAN virtualized component is deployed

5.4.3 Accounting for Slice Instance

Table 19 reports the sequence of actions required for the execution of the **accounting of network slices** TUC.

Table 19: Accounting for Slice Instance Steps

Step	Impacted Architecture Plane	Description
1	**Data Plane**	LCM events generated by NFs and/or Infrastructure
2	**Management Plane**	LCM events reaches NFs management and /or Infrastructure management (VNFM, EMS, VIM, …)
3	**Management Plane**	LCM events are collected for the NSSIs
4	**Management Plane**	LCM events are collected for the NSIs (Cross-plane orchestrator generated NSI LCM events are also included)
5	**Management Plane**	NSI LCM events are exposed to e.g. BSS where billing is produced (BSS is out of SLICENET scope)

5.4.4 Network Slice Instance Performance Management

Table 20 reports the sequence of actions required for the execution of the **Network Slice Instance Performance Management** TUC.

Table 20: Network Slice Instance Performance Management Steps

Step	Impacted Architecture Plane	Description
1	**Data Plane**	Performance statistics of infrastructure resources and NFs are generated.
2	**Management Plane**	An order of measurement jobs (either default or custom) is present in relation to NSSIs pertaining the NSI for which performance data needs to be collected.
3	**Management Plane**	Each NSSI orders to fetch the statistics from associated NFs.
4	**Management Plane**	NFs and/or Infrastructure management (VNFM, EMS, VIM, …) reads the statistics from NFs.
5	**Management Plane**	Statistics of all NSSIs are collected only for the relevant NSI.
6	**Management Plane**	Performance level data of an NSI is generated based on agreed Services.
7	**Management Plane**	Performances are exposed to the service consumer.

5.4.5 Network Slice Instance Security Management

Table 21 reports the sequence of actions required for the execution of the **Network Slice Instance Security Management** TUC.

Table 21: Network Slice Instance Security Management Steps

Step	Impacted Architecture Plane	Description
1	**Data Plane**	NFs for security monitoring (e.g. DDoS attacks) are deployed and configured
2	**Management Plane**	NFs management and /or Infrastructure management (VNFM, EMS, VIM, …) configures access and management rights to deployed infrastructure elements and NFs
3	**Management Plane**	Access and authentication credentials are exposed to upper layers per each NSI/NSSI
4	**Management Plane**	Access and authentication credentials are stored for further configuration possibilities in security management layer at NSI level.
5	**Data Plane**	A security threat is detected by relevant NFs. Threat may be resolved at the DP
6	**Management Plane**	NFs and or/ Infrastructure management is notified about security threat.
7	**Management Plane**	Upper layers are notified to determine affected NSIs/NSSIs.

5.5 SON evolution for network slice management

5.5.1 Automated Optimization of a Network Slice Instance

Table 22 reports the sequence of actions required for the execution of the **automated Optimization of a Network Slice Instance** TUC.

Table 22: Automated Optimization of a Network Slice Instance Steps

Step	Impacted Architecture Plane	Description
1	**Intelligence Plane**	Policies are added or modified as a result of the machine learning mechanisms.
2	**Control Plane**	Policies are checked
3	**Control Plane**	QoE optimization algorithms determine if modifications onto deployed NSI are needed to maintain QoE levels.
4	**Management Plane**	Orchestrate change action and identify NSIs to update
5	**Control Plane**	Trigger NSIs/NSSIs re-configuration

6	**Management Plane**	Orchestration to add/remove NFs or NFs re-configuration takes place
7	**Data Plane**	New NFs are added while previous allocated NFs are re-configured or removed
8	**Data Plane**	Network links interconnecting configured NFs are updated or removed
9	**Management Plane**	**N**ew NSI/NSSI configuration and resources are reported
10	**Management Plane**	**N**ew NSI/NSSI configuration is exposed

5.5.2 Automated Healing of a Network Slice Instance

Table 23 reports the sequence of actions required for the execution of the **automated healing of a Network Slice Instance** TUC.

Table 23: Automated Healing of a Network Slice Instance Steps

Step	**Impacted Architecture Plane**	**Description**
1	**Data Plane**	A failure is detected at the data plane, either infrastructure or NF
2	**Management Plane**	Failure is reported to the NF and/or infrastructure management. Automated protection/restoration mechanisms at the data plane may be triggered
3	**Data Plane**	Faulty infrastructure/NFs are isolated and alternative resources are configured
4	**Management Plane**	New configuration is reported to upper layers
5	**Management Plane**	Cross-layer fault recovery mechanisms may be triggered if data plane does not support automated protection/restoration
6	**Management Plane**	Affected NSI/NSSIs are determined
7	**Management Plane**	Orchestration of alternative infrastructure resources and NFs to overcome failure is performed
8	**Control Plane**	Policy framework and P&P rules are checked
9	**Control Plane**	Resources and NFs (re-)configuration is triggered
10	**Data Plane**	Resources and NFs are re-configured

5.6 Multiple administrative domain Slice

5.6.1 Create an end-to-end NSI across multiple operators

Table 24 reports the sequence of actions required for the execution of the **creation of an end-to-end NSI across multiple operators** TUC.

Table 24: Create an end-to-end NSI across multiple operators Steps

Step	Impacted Architecture Plane	Description
Pre-conditions: 1. Management and Control information among peer domains are shared and on-boarded as NSI templates in the cross-plane orchestrator catalogue.		
1	**Management Plane**	Customer requests NSI spanning multiple operators.
2	**Management Plane**	NSI is decomposed in NSSIs per involved domains.
3	**Management Plane**	Instantiation of Other-Domain NSSIs is ordered.
4	**Control Plane**	Intra-domain NSSI creation is triggered.
5	**Control Plane**	NSSI instantiation is ordered to management plane.
6	**Management Plane**	Orchestration of Network Services takes place to instantiate and configure the NFs and infrastructure.
7	**Data Plane**	NFs and network links are configured.
8	**Data Plane**	Inter-domain network resources are configured to interconnect the involved domains.
9	**Data Plane**	Active probing and monitoring functions are deployed to guarantee cross-domain SLAs

5.6.2 Monitoring end-to-end NSI across multiple operators

Table 25 reports the sequence of actions required for the execution of the **monitoring of an end-to-end NSI across multiple operators** TUC.

Table 25: Create an end-to-end NSI across multiple operators Steps

Step	Impacted Architecture Plane	Description
1	**Data Plane**	Active probe and monitoring functions send collected data to Orchestrator
2	**Management Plane**	Monitoring data is aggregated per SLA and matched against requirements
3	**Management Plane**	Aggregated SLA data of multi-domain services is sent to requesting

		domain orchestrator (the one initiating the request)

5.6.3 Management support to facilitate UE roaming between Network Slice Instances in different administrative domains

Table 26 reports the sequence of actions required for the execution of the **management support to facilitate UE roaming between Network Slice Instances in different administrative domains** TUC.

Table 26: Management support to facilitate UE roaming between Network Slice Instances in different administrative domains Steps

Step	**Impacted Architecture Plane**	**Description**
1	**Management Plane**	Abstracted information of NSIs is exchanged to determine roaming agreement among domains
2	**Management Plane**	Orchestration of network functions and infrastructure elements takes place to support coverage area per domain
3	**Data Plane**	Infrastructure resources and NF are (re-)configured to allow for data/function mobility across the physical/virtual NSIs of each operator
4	**Control Plane**	Control functions trigger resource handover to enable UE mobility across operators with a roaming agreement

5.7 Management exposure - Limited level of management exposure for multiple Network Slice Instances (related to P&P)

5.7.1 Basic level of control exposure for NSI

Table 27 reports the sequence of actions required for the execution of the **Basic level of control exposure for NSI** TUC.

Table 27: Basic level of control exposure for NSI Steps

Step	**Impacted Architecture Plane**	**Description**
Pre-conditions: 1. NSI created (i.e. slice resources and NFs provisioned) 2. Slice customer and provider agreed on basic level of control exposure for NSI		
1	**Management Plane**	Once the NSI is provisioned by cross-plane orchestrator, the dedicated P&P control instance is activated through the related P&P management instance
2	**Management Plane**	The P&P control instance is given access to APIs for collection of KPIs related to NFs performance
3	**Management Plane**	The P&P control instance is given access to APIs for collection of KPIs related to slice performance

4	**Control Plane**	**A** correspondent set of control primitives for NFs and slice KPIs monitoring is activated and exposed to slice customer

5.7.2 Limited level of control exposure for NSI

Table 28 reports the sequence of actions required for the execution of the **Limited level of control exposure for NSI** TUC.

Table 28: Limited level of control exposure for NSI Steps

Step	**Impacted Architecture Plane**	**Description**
Pre-conditions: 1. NSI created (i.e. slice resources and NFs provisioned) 2. Slice customer and provider agreed on limited level of control exposure for NSI 3. Steps of CTRL_MGMT_EXP_NSI_1 have been already implemented		
1	**Data Plane**	**T**he P&P control instance is given access to programmable NF APIs for configuration purposes
2	**Control Plane**	**T**he P&P control instance is given access to a limited set of SDN APIs for customized slice run-time operation
3	**Management Plane**	**T**he P&P control instance is given access to a limited set of NFV APIs for managing a restricted set of lifecycle aspects of VNFs
4	**Control Plane**	**A** correspondent set of control primitives for programmable NFs configuration, SDN logics, and limited NFV management is activated and exposed to slice customer

5.7.3 Extended level of control exposure for NSI

Table 29 reports the sequence of actions required for the execution of the **Extended level of control exposure for NSI** TUC.

Table 29: Extended level of control exposure for NSI Steps

Step	**Impacted Architecture Plane**	**Description**
Pre-conditions: 1. NSI created (i.e. slice resources and NFs provisioned) 2. Slice customer and provider agreed on extended level of control exposure for NSI 3. Steps of CTRL_MGMT_EXP_NSI_1 and CTRL_MGMT_EXP_NSI_2 have been already implemented		
1	**Management Plane**	The P&P control instance is given full access to NFV APIs for VNF lifecycle management
2	**Management Plane**	The P&P control instance is given access to slice management APIs
3	**Management Plane**	The P&P control instance is given access to slice orchestration APIs

4	**Control Plane**	A correspondent set of control primitives for NFV and slice management/orchestration is activated and exposed to slice customer

5.8 Cognition

5.8.1 Data Collection for Cognition

Table 30 reports the sequence of actions required for the execution of the **Data Collection for Cognition** TUC.

Table 30: Data Collection for Cognition Steps

Step	**Impacted Architecture Plane**	**Description**
1	**Management Plane**	Define monitoring policy
2	**Data Plane**	Performance statistics of infrastructure elements and deployed NFs are generated
3	**Monitoring Plane**	Collect resource metrics/KPIs, report them in a structured way and signal on preconfigured events (such as reaching B/W watermarks for specific Slice)
4	**Monitoring Plane**	Discover and report topology configuration
5	**Monitoring Plane**	Obtain and stream service-level (structured) logs
6	**Monitoring Plane**	Provide preconfigured full/partial collection of network flows and stream the data
7	**Intelligence Plane**	Receive monitored data
8	**Intelligence Plane**	Process (clean, reformat, transform, convert and annotate) and filter structured data
9	**Intelligence Plane**	Store both structured and unstructured data
10	**Intelligence Plane**	Forward information to interested subscribers
11	**Intelligence Plane**	Obtain data through queries from other data aggregators
12	**Intelligence Plane**	Answer queries of other data aggregators (verify permissions, clean the data)

5.8.2 QoE Classification via Machine Learning

Table 31 reports the sequence of actions required for the execution of the **QoE Classification via Machine Learning** TUC.

Table 31: QoE Classification via Machine Learning Steps

Step	Impacted Architecture Plane	Description
1	Management Plane	Define desired QoE via One-Stop-API
2	Management Plane	Define slice SLAs and monitored KPIs
3	Management Plane	Configure monitoring policy
4	Data Plane	Performance statistics of infrastructure elements and deployed NFs are generated
5	Monitoring Plane	Collect and report resource metrics/KPIs
6	Intelligence Plane	Collect historical KPI data
7	Management Plane	Provide QoE labels and app context via One-Stop-API as level of perceived QoE (either from DSP or CSP)
8	Management Plane	Management Plane receives the perceived QoE
9	Control Plane	Report SLA control adjustments and known violations
10	Management Plane	Decompose, compile and dispatch labels
11	Intelligence Plane	Label historical data
12	Intelligence Plane	Learn KPI classification by QoE labels. Learn KPI to QoE correlation
13	Intelligence Plane	Install classification model (via policy framework)
14	Control Plane	Apply model to infer QoE for current KPIs
15	Control Plane	Adjust configuration to achieve QoE
16	Intelligence Plane	Refine and update model with new data and labels

5.8.3 Proactive Fault Management via Machine Learning

Table 32 reports the sequence of actions required for the execution of the **Proactive Fault Management via Machine Learning** TUC.

Table 32: Proactive Fault Management via Machine Learning Steps

Step	Impacted Architecture Plane	Description
1	**Management Plane**	Define fault data collection policy and learning goals
2	**Management Plane**	Receive fault data collection policy and learning goals
3	**Data Plane**	Fault events are generated
4	**Management Plane**	FCA module reports faults to Cognitive Monitoring
5	**Intelligence Plane**	Collect fault stats history and related KPIs (multiple slice instances)
6	**Intelligence Plane**	Compute and persist model of fault probabilities per measured KPIs
7	**Intelligence Plane**	Compute minimal set of KPIs needed to monitor fault with good enough probability
8	**Management Plane**	Instantiate slice and define proactive fault management goals (level) and actions
9	**Intelligence Plane**	Compute KPI to action rules and create per-slice policy
10	**Monitoring Plane**	Monitor KPIs per defined policy and raise applicable alerts
11	**Control Plane**	Execute corrective proactive fault management action when KPI match policy
12	**Management Plane**	Manage further actions (could be cross-domain)
13	**Intelligence Plane**	Refine and update model per new fault data

6 System Architecture

This chapter starts by providing an overview of the main requirements for the SLICENET system architecture design (section 6.1). Based on this information, section 6.2 provides the initial SLICENET system architecture and section 6.3 the main reference points. Finally, in section 6.4 are illustrated and described a group of workflows related with the NS lifecycle, that is, a high-level view of how the SLICENET functional architecture will work.

6.1 Architecture Design Methodology and Requirements Overview

The key innovation aspects of the SLICENET project (Network Slicing, One-Stop API, Plug & Play, Cognition and Cross-Plane Orchestration), which are detailed in section 4 of this document, together with the vertical use-cases, were the foundation for the identification of the TUCs and system requirements, presented in section 5. The identified requirements and TUCs are the input for the design of the system architecture presented in this section.

As identified in the system requirements, the SLICENET architecture must deliver a slicing framework which is able to handle the complete NS journey. In practical terms, this means that it should allow the design and onboard of the NSs by the NO, as well as to enable the DSP (or DSC) to subscribe, configure, control and decommission the NS. Additionally, it should encompass the capability to monitor and assure that the NS is delivered to the DSP (or DSC) according to the established KPIs.

Figure 13 provides an overview of the phases and activities that are required to manage and deliver a NSI to a DSP/DSC (based on the 3GPP approach).

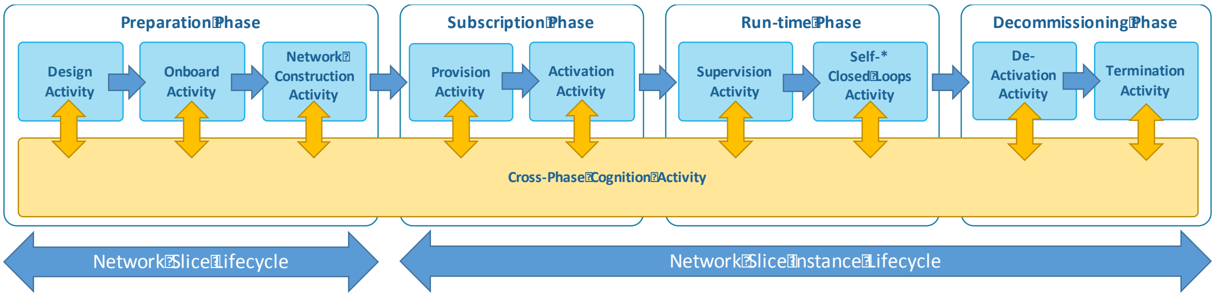

Figure 13: Network Slice Phases and Activities

The following phases and activities are defined:

- **Preparation Phase** – during this phase all the required procedures to prepare the NS subscription by the DSP/DSC are prepared. Its composed by the following core activities:
  - **Design Activity**: NS design, including the templates and blueprints, by the NO, either due to business requirements to serve potential DSP/DSC, or due to internal necessities;
  - **Onboard Activity**: after designing the NS and all its resource dependencies, the produced templates are onboarded to the architecture components (catalogues) and made available to be invoked;
  - **Network Construction Activity**: the required network resources are deployed/instantiated and configured according to the NS templates;
- **Subscription Phase** – during this phase the NSs are subscribed by the DSP/DSC. Its composed by the following core activities:
  - **Provision Activity**: guarantee that all required resources and services to deliver the NS are deployed and configured according to the DSP/DSC request;
  - **Activation Activity**: the provisioned resources and services are activated and, as a result, the NSI starts running and is available;

- **Run-time Phase** – during this phase are included all the required procedures to keep the NSIs running healthy. Its composed by the following activities:
    - **Supervision Activity**: monitor all the required information to evaluate the status of the NSIs; includes reporting information towards the NO system administrator, as well as to the DSP/DSC;
    - **Self-* Closed-Loops Activity**: encompasses the policy-driven self-* closed loops in order to automatically and proactively manage and/or control the NSIs; it can include, for example and among others, self-protection, self-healing, self-configuration and/or self-optimization closed-loops;
- **Decommission Phase** – during this phase the running NSI is terminated. Its composed by the following activities:
    - **De-Activation Activity**: all the NSI related resources and services configurations are removed;
    - **Termination Activity**: the registered NSI is removed from the inventories and no longer available;
- **Cross-Phase Cognition Activity** – this is a transversal phase/activity, potentially impacting the activities in all the other phases/activities. It is responsible for all the intelligence-related procedures in the system, affecting, for example, NS design (e.g. filling templates with QoE related parameters and/or models), provisioning (e.g. selecting the best providers, configuring thresholds, etc.) and run-time (e.g. updating or creating new policies for the self-* closed-loops).

6.2 Logical Architecture

The high-level perspective of the logical architecture, also named as “Level 0”, is presented in Figure 14. Three main pillars are used to structure the “Level 0” of the logical architecture: **planes**, **sub-planes** and **layers**.

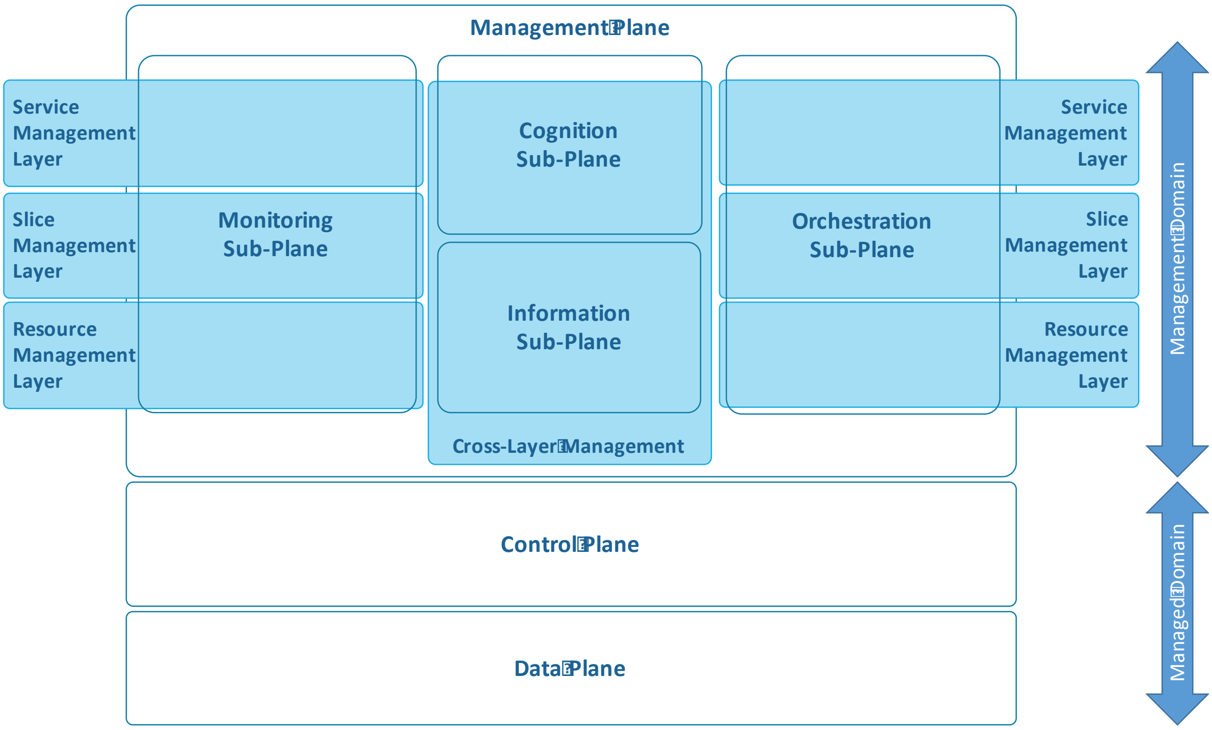

Figure 14: SLICENET “Level 0” Logical Architecture

The architecture **planes** refer to very large groups of components that are used in different functional phases of the system. Three planes are adopted from the digital services industry and used in SLICENET:

1. **Management Plane**: encompasses all the mechanisms related with the design, deployment, provisioning, configuration, supervision and decommissioning of network resources, slices and services;
2. **Control Plane**: includes the mechanisms required to guarantee that the configurations applied by the management plane are respected and executed during the slice and service delivery/run-time;
3. **Data Plane**: responsible for processing and forwarding packets between network elements.

The management plane is also structured in several internal **sub-planes**. In detail, the following sub-planes are identified:

1. **Information Sub-Plane**: maintains all the required knowledge, such as service/slice templates and service/slice registries that are required by the other sub-planes and planes of the system. As a logically centralized sub-plane, it avoids replication and incoherent information to be spread in several architecture components. All the architecture sub-planes interact with the Information Sub-Plane;
2. **Orchestration Sub-Plane**: ensures that the required actions are fulfilled in the correct order to deploy and/or configure the three SLICENET logical abstractions – resources, slices and services. It is the sub-plane responsible for actuating on the network elements, as well as to request the administrative domains involved in the end-to-end slice the instantiation of their slices;
3. **Monitoring Sub-Plane**: collects, filters and enriches network information that will be used to understand the performance and usage of the network slices and services. It gathers information, such as counters, events and/or alarms, from the running physical/virtual network resources, slices and services. This sub-plane is critical for providing information to the intelligence procedures deployed in the Cognition Sub-Plane, as well as to allow the NO administrator and/or the DSP/DSC to view the slice and service status, as well as to retrieve performance information from the peer administrative domains;
4. **Cognition Sub-Plane**: guarantees the proactive management (Performance, Fault, SLA, etc.) procedures of the network slice and services. Towards this goal, it pre-processes and aggregates the collected network data in order to allow the automation procedures through policy-based closed loops without human intervention. More importantly, it also incorporates autonomous network procedures through analysis mechanisms based on Artificial Intelligence techniques.

Finally, within the management plane, three different **layers** are defined to abstract the resources, slice and service details. This guarantees a clear separation of concerns between the logical architecture components responsible for managing the services, the slices and the resources. In summary, the following layers are defined:

1. **Resource Management Layer**: responsible for the network resources lifecycle management procedures, either they are physical (PNF), virtual (VNF) or SDN resources;
2. **Slice Management Layer**: responsible for the NS lifecycle management procedures, being able, for example, to decompose a NS instantiation request to the required network resources, as well as to collect and process information about the running NSs;
3. **Service Management Layer**: responsible for the service lifecycle management procedures, including, for example, the exposition of NSs as a service to the DSP/DSC and the decomposition of the customer service into one or more NSs.

Within the Monitoring and the Orchestration Sub-Planes, the Service, Slice and Resource Management Layers are independent and have their own architecture components to deal with the monitoring and orchestration procedures at each layer (as further described in section 6.2). On the other hand, the Information and Cognition Sub-Planes are an exception and are not split in different layers. In this case, since their architecture components require Service, Slice and Resource level information, they are Cross-Layer.

One of the most important concepts of SLICENET is the is the capability to apply cognitive management through enhancing an autonomous, policy-based, closed loop with machine learning capabilities. Two main policy-based autonomous closed loops are foreseen within the SLICENET system architecture:

1. **Cognition Closed Loop & Autonomous Management Closed Loop**:
   a. this policy-based closed loop involves the Monitoring, Cognition and the Orchestration Sub-Planes and is used for longer term proactive management procedures, such as QoE/performance scenarios;
   b. besides implementing the autonomous management closed loop described in a), this loop also includes the intelligent, e.g. machine-learning, procedures which, as a result, can update or create new policies for both the autonomous management closed loop (described in a)) and for the autonomous control closed-loop (described in the next paragraph);
2. **Autonomous Control Closed Loop**:
   a. this policy-based closed loop involves the Monitoring, Control and Orchestration Sub-Planes and is used mostly for real-time sensing and actuation on the network. It is based on policies and models fed by the Cognition Sub-Plane.

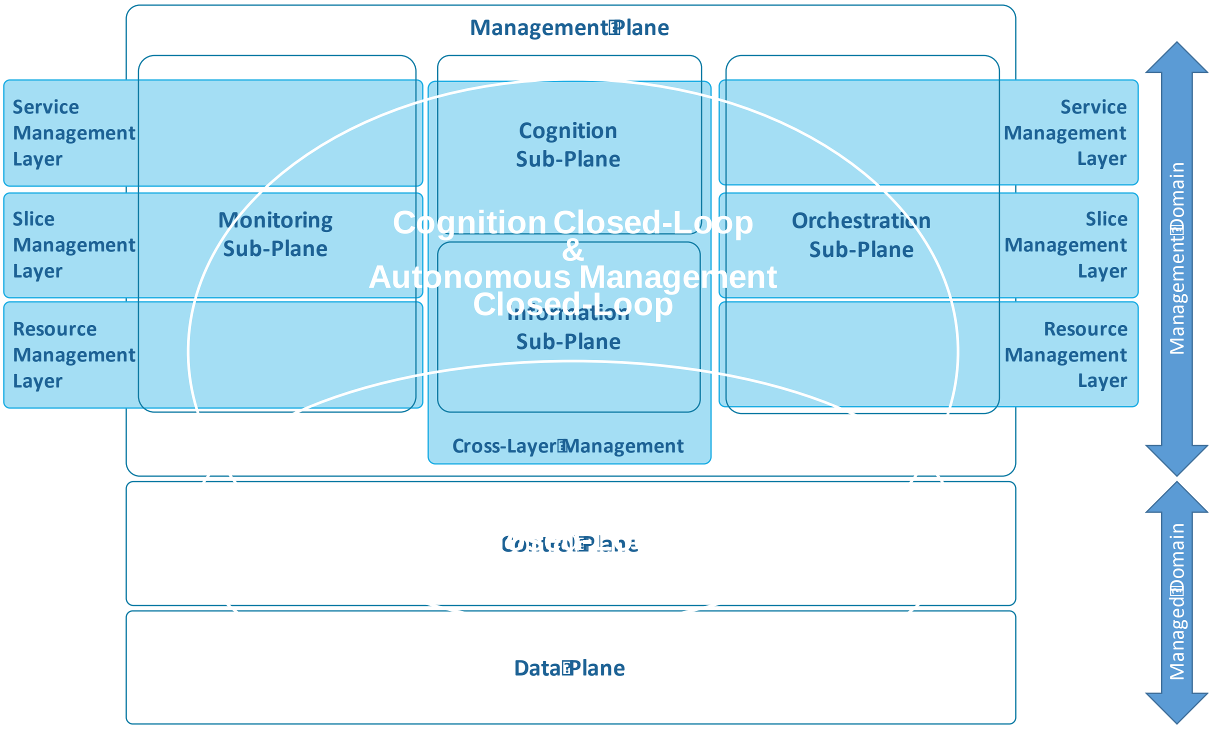

Figure 15: SLICENET Cognition, Autonomous and Control Closed-Loops

Further and more detailed information about the SLICENET closed-loops will be given in subsequent deliverables of the project.

Figure 16 goes one step further and depicts the "Level 1" system architecture. This perspective provides the inner components within each Plane, Sub-Plane and Layer. It is important to mention that the identified components in this "Level 1" architecture perspective are not definitive and can be changed (expanded, collapsed, etc.) during the project lifetime in more, low-level, detailed activities.

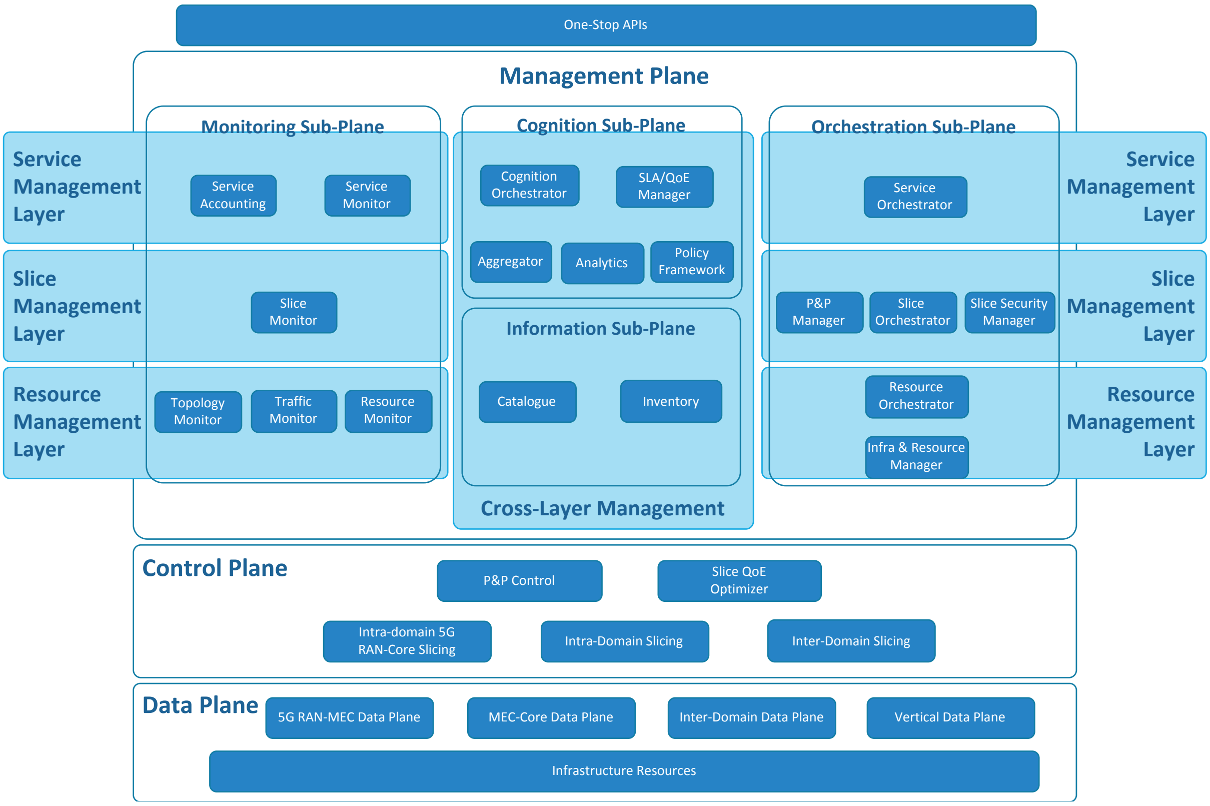

Figure 16: SLICENET "Level 1" Logical Architecture

In the following sections detailed descriptions about each "Level 1" architecture component is given.

6.2.1 Information Sub-Plane @Management Plane

The Information Sub-Plane is responsible for handling the service and slice related information that is required by the other components of the system architecture, therefore guaranteeing coherence of information and avoiding duplication. Two components are envisaged within this Sub-Plane: i) Service & Slice Catalogue (section 6.2.1.1) and ii) Service & Slice Inventory (section 6.2.1.2).

6.2.1.1 Catalogue

The Service Catalogue component role is characterized by providing persistency to Service templates or descriptors that can be used by Service Orchestrators to advertise and manage their complete lifecycle. Each service template shall hold the necessary information to characterize each service from the customer (e.g. QoS, SLAs) and operational perspectives (e.g. service operations, service components). Moreover, since each service may be realized by a federation of sub-services from distinct Network Operators, the Service Catalogue role shall include the federation of Service Catalogues using mechanisms to support service advertisement and validation.

Like the Service Catalogue, the Slice Catalogue component provides persistency to Slice templates or descriptors that can be used by the Slice Orchestrator to compose a Slice using the available building blocks. Each building block consists of an independent segment of virtualized resources that needs to be interconnected to create an isolated and unified logical infrastructure.

To manage the complete lifecycle of a single slice, the Slice Orchestrator interacts with the Resource Orchestrator, which is responsible for the respective lifecycle management of different virtual resources. Thus, it is expected that these management systems also implement catalogues that need to be integrated with the Slice Catalogue. To avoid storing redundant information, the Slice template shall keep the minimum needed information about the different building blocks that it needs to compose a single slice.

6.2.1.2 Inventory

The component realizing the Service and Slice inventory roles shall persist all the information regarding service and slice instances, which includes all necessary resource information taking into account the federation of inventories throughout the SLICENET framework, e.g. a Service Network Functions Forwarding Graph in a slice. Moreover, since a service may be composed by service instances in other administrative domains, it also includes the necessary information for the aggregation and management of multi-domain services and slices.

The Inventory functionality also includes storing information regarding the infrastructure to enable orchestration capabilities at all layers, i.e. service, slices and resources. Also associated with this capability regarding the infrastructure, is the resource allocation functionality which allows orchestration components to request specific resources. Moreover, this process of resource allocation may be influenced by orchestration components using parameters such as geographic locations, specific resource requirements or even network applications categorization.

Finally, since various components may also provide inventories, the Service & Slice Inventory component shall include the federation of inventories capabilities to avoid duplication of stored information.

6.2.2 Orchestration Sub-Plane @Management Plane

The Orchestration Sub-Plane is the one that coordinates the required sequence of actions to deploy and/or configure the SLICENET services, slices and resources across a single and/or multiple administrative domains. Six components are foreseen within this sub-plane: i) Service Orchestrator (section 6.2.2.1, ii) Slice Orchestrator (section 6.2.2.2), iii) Resource Orchestrator (section 6.2.2.3), iv) P&P Manager (section 6.2.2.4), v) Infrastructure & Resource Manager (section 6.2.2.5) and vi) Slice Security Manager (section 6.2.2.6).

6.2.2.1 Service Orchestrator

A service represents the realization of a product in a technology-agnostic way, which can be customized to fulfill each customer's requirements, i.e. SLA and QoE. The role of the Service Orchestrator is to coordinate the actuation management processes to fulfil and guarantee continuous delivery of services to individual customers in a multi-tenant environment. To realize this role, the Service Orchestrator shall expose the necessary interfaces which customers can use to consult available services, request new services, update and monitor the state of existing ones.

Since SLICENET addresses multi-domain scenarios, where services may cross several Network Operators Infrastructures, a single service may be decomposed in multiple services in a recursive manner. Thus, the Service Orchestrator shall be able to communicate with other Service Orchestrators to request services instantiation.

6.2.2.2 Slice Orchestrator

Services in SLICENET are realized through Slices which consist of isolated virtual infrastructures deployed and managed at runtime with the use of the Slice Orchestration functionality. The latter is the slice-related counterpart of Service Orchestration, which means coordinating one or more technology-independent virtualization domains by interacting with their respective management systems. The coordinated integration of distinct domains enables building isolated and unified slices that can be managed during their complete lifecycle by a Slice Orchestrator. To achieve this, the Slice

Orchestrator component will use aggregation capabilities to combine information and operations at a slice-level.

6.2.2.3 Resource Orchestrator

SLICENET establishes a hierarchy covering from services to resources, where Services are composed by slices and in turn, slices are an aggregation of resources deployed over a shared infrastructure. Thus, Resource Orchestration represents the lowest level of orchestration within the Cross-Plane Orchestration and provides the building blocks, e.g. VNFs, for Slice Orchestration. In this role, it can be included ETSI MANO management components, SDN components for the applications lifecycle management and other applicable functionalities.

The previously mentioned functionalities may be conditioned by slice orchestration ones, where some resource parameters may be characterized using input coming from the top layers, e.g. a specific implementation of a VNF may be chosen based on service and/or slice requirements.

6.2.2.4 P&P Manager

The SLICENET Plug & Play control is conceived as a set of control and management primitives and APIs exposed towards slice consumers to have direct run-time control over their slice instances, thus enabling their customization. As any other control tool or component, it requires its own management mechanisms and procedures in order to be properly activated, configured and used.

As described in section 6.2.5.2, a dedicated SLICENET Plug & Play instance is intended to be activated for each slice or sub-slice part of an end-to-end slice. During the end-to-end slice instance provisioning, it is envisaged that the Plug & Play manager takes care of the activation of the specific vertical tailored Plug & Play control instance, applying all those customization and configurations required to expose to the vertical (or the generic slice consumer) the correct level of control over the slice resources and service logics.

To fulfil these management operations, the Plug & Play Manager needs to interact with Service and Slice orchestration components within the Orchestration Sub-Plane. Indeed, according to the end-to-end slice requirements and capabilities included in the given slice template (as stored in the Slice Catalogue), possibly augmented with dedicated updates or refinements specific per slice customer and slice instances applied through the One-Stop-API, the Plug & Play Manager triggers the activation of a new Plug & Play control instance.

In summary, the Plug & Play Manager is in charge of providing at least the following lifecycle management operations for each Plug & Play control instance:

- Activation: this allows to deploy a new instance of Plug & Play when a new end-to-end slice (or sub-slice) is created. This operation is triggered by service and slice orchestration components in the context of an end-to-end slice provisioning.
- Plug of specific control and management drivers: this operation is required to restrict the access to only those slice control and management APIs and logics to which the vertical (or the slice consumer) should have access, according to the slice capabilities and vertical requirements set into the slice template and through the One-Stop-API.
- Configuration of vertical tailored abstraction: this operation provides a vertical tailored abstraction of the SLICENET control and management primitives and logics, and allows on the one hand to protect the slice providers' proprietary control information models, interfaces and procedures, and on the other to reduce the complexity exposed to the slice consumer.
- Deactivation: whenever an end-to-end slice instance is de-commissioned, the related Plug & Play control instance has to be deactivated in order to deny the vertical (or slice consumer) the access to the SLICENET control and management logics.

6.2.2.5 Infrastructure and Resource Manager

A RAN slicing architecture allows radio resource management (RRM) policies to be enforced at the level of physical resource blocks (PRBs) through providing the virtualized resource blocks (vRBs) via a resource visor toward each slice. A RAN slicing runtime system presented in [78] provides a flexible execution environment to run multiple virtualized RAN instances with the required level of isolation and sharing of the underlying RAN modules and resources. Specifically, it allows slice owners to create and manage their slices, to perform custom control logics such as handover decisions and to operate on a set of virtual resources, e.g. resource block or spectrum.

Network slicing and RAN and CN lifecycle management is applicable for both slice owners and infrastructure providers, allowing different levels of sharing and isolation across resources and network functions, as well as a fine-grain service management and orchestration on per slice basis, shared or slice-specific, in particular through the design and runtime phases. Disaggregated RAN and CN support is applicable at the level of the Infrastructure provider, enabling a flexible deployment of infrastructure in order to increase capacity, coverage, and multiplexing gain in terms of OPEX. Last, support for resource abstraction and coordination applies to both slice owners and infrastructure providers, allowing resource isolation and performance guarantees along with a maximizing resource utilization and increase in terms of total number of supported slices.

The management of infrastructure and resources should cover both the physical and the virtual layers, and the Virtualised Infrastructure Management (VIM) is required and highlighted. The VIM is responsible for managing the NFV infrastructure (NFVI) of a 5G operator. It manages a Resource Repository of NFVI hardware resources (compute, storage, networking) and software resources (hypervisors, software images), and keeps tracking the allocation of virtual resources to physical resources through a Resource Inventory to allow managing the NFVI resources and optimize their use.

6.2.2.6 Slice Security Manager

The Slice Security Manager component provides functionality required to secure individual slice instances from each other. Depending on the negotiated slice features, specific network paths within a slice may need to be encrypted end-to-end, using a suitable encryption scheme. To this end, the Slice Security manager will trigger the deployment of predefined de/encryption functions in proximity to, or within the customer's NFVs. The component will interact with other architecture components to distribute required keys.

The Slice Security Manager will similarly support the management of security related network functions, (e.g., DPI, IDS) utilized by the control loop dealing with strategies for mitigating attacks that may degrade or disrupt the slice operation.

6.2.3 Monitoring Sub-Plane @Management Plane

The Monitoring Sub-Plane is the one that performs the architecture sensing functionalities, that is, collection, filter and enrichment of counters, events and alarms retrieved from the network resources that compose the slice delivered to the vertical. Six components are foreseen within this sub-plane: i) Resource Monitor (section 6.2.3.1), ii) Topology Monitor (section 6.2.3.2), iii) Traffic Monitor (section 6.2.3.3), iv) Slice Monitor (section 6.2.3.4), v) Service Monitor (section 6.2.3.5) and vi) Service Accounting (section 6.2.3.6).

6.2.3.1 Resource Monitor

The Resource Monitor module is the responsible entity for collecting monitoring information from the resources composing the data plane, thus enabling the monitoring functionalities of the upper level modules of the Monitoring Framework. To this end, the Resource Monitor has to implement the interface that interconnects the monitoring and the data planes. Such interface will need to support the various technologies that are involved in the different segments of the data plane. More specifically, monitoring the operator's infrastructure involves sensing its resources either they are virtual (VNFs), SDN-based and/or physical (PNF), collecting raw data regarding their alarms, events and performance counters. The collected resource raw data is stored in this module, which should be big data compliant (store and manage large volumes of data).

6.2.3.2 Topology Monitor

Topology Monitor collects and reports user mobility and infrastructure/network topological information to enable cognition based on the awareness of location, topology and/or mobility. Firstly, the UE's mobility should be tracked to allow mobility management in the 5G network for handover control and location management, as required especially in the 5G eHealth ambulance use case. Secondly, dynamic changes in both physical and virtual infrastructures in the network should be monitored to allow topological context awareness, and the topologies in both physical and virtual layers should be correlated to facilitate cross-layer infrastructure-aware actions. Thirdly, this real-time topological view would enable the on-demand deployment of network functions at the most appropriate location, especially to facilitate QoE optimization when QoE actuators are deployed as the result of the planning made by the intelligence.

6.2.3.3 Traffic Monitor

A Traffic Monitor at the flow level in SLICENET is required as an enabling component for performance management, QoS monitoring, QoE optimisation and/or some other cognition-related tasks that are concerned with the behaviour of traffic flows. The Traffic Monitor is expected to collect and report a range of flow-level metrics and states for the concerned flows traversing the network. With the different roles in the system taken into account, two kinds of Traffic Monitors may be required: Infrastructure Provider's Traffic Monitor, and Network Operator's Traffic Monitor. An Infrastructure Provider's Traffic Monitor should be able to detect and differentiate traffic flows belonging to multiple tenants (i.e., Network Operators) and thus multi-tenancy awareness is required. Meanwhile, an Network Operator's Traffic Monitor should be tailored to support the traffic monitoring in a particular type of network. In the context of SLICENET, LTE/5G networks are focused on and hence LTE/5G traffic awareness is featured. In both cases, a Traffic Monitor should be hooked to the data plane of the SLICENET architecture to provide the monitoring functions in the Monitoring sub-plane.

6.2.3.4 Slice Monitor

The Slice service characteristics as defined by the SLA will be used to resolve any differences between SLA and the actual slice performance. QoS levels that require reaction will be subject to intra-domain or inter-domain resolution and the events will be forwarded to the appropriate intra or inter domain management modules. Individual Network Slice level performance data is also useful to decide to scale up or down services within those slices. Performance data (or events) includes: user and control traffic load data, QoS/SLA data, e.g. indicating whether services were provided at expected QoS/SLA level. Alarms notifications can be individually enabled. Events and alarms from a shared Network Slice contain enough information to be attributed by the 3GPP management system to one of the Complete 3GPP Network Slices that contain this shared Network Slice.

The task of monitoring a slice is much aligned with the Resource Monitoring role in sensing resources, in this case a subset of the resources available to the Resource Monitoring, and transforming them by aggregating raw data in a batch and/or event processing fashion. Thus, enabling this module to provide an overview regarding slice performance.

6.2.3.5 Service Monitor

As services are expected to rely on NSIs, service monitoring will be collecting the fault and performance information provided by the NFs relating to all NSIs involved in a Service Instance definition. This monitoring phase englobes the aggregation of previously monitored data (resource and slice level), by using the same aggregation methods used in Slice Monitoring but paying attention to the services that such data belongs to. This enables a clear distinction between data that belongs to a service that crosses different slices (multi-domain slices) and data that lives within a certain slice and is part of an end-to-end network service. This type of monitoring enables SLICENET to support and enforce SLAs at a service and slice level.

6.2.3.6 Service Accounting

The components realizing the single domain accounting are responsible for handling accounting of slices and its components. The accounting function is based on data retrieved from the inventory of resources at several layers (slice, NF) available in the management plane. Events are generated from the operation of resources and diverted to this accounting module, which then exposes them to any billing system (which could be the BSS itself or other billing system) is produced.

6.2.4 Cognition Plane @Management Plane

The Cognition Sub-Plane is the main architecture area in SLICENET. It deals with all the data pre-processing procedures to create slice and service metrics, as well as with the policy-based self-control loops (e.g. Self-Optimization, Self-Healing) and, most importantly, with the artificial intelligence techniques. Five components are envisaged for this sub-plane: i) Policy Framework (section 6.2.4.1), ii) Cognition Orchestrator (section 6.2.4.2), Analytics (section 6.2.4.3), Aggregator (section 6.2.4.4) and v) QoE/SLA Manager (section 6.2.4.5).

6.2.4.1 Policy Framework

Policies represent rules and restrictions that govern the behavior of a system. Herein an important type of behavioral policies is considered, obligation policies, where the Policy Framework components guide the decision-making process regarding what the system should or should not do. Most decisions that are taken across the whole system are determined by rules that implement policies at various levels, ranging from decisions at the resource level, taken in real-time using locally generated information, to business decisions that are based in information aggregated in time and space, from various sources.

Different policy paradigms may be considered, e.g. imperative and/or declarative, to be mapped in distinct management layers, i.e. service, slices and resources. Although related they are distinct in nature, and a change on a policy will have effects that ripple across the whole system, and eventually lead to changes in policies that depend of it. Usually, these dependencies are managed only in restricted policy groups, typically in single-vendor environments, and cannot be managed E2E.

To enable this, the following processes shall be supported: policy design, policy validation, language translation, conflict resolution and execution. A brokering mechanism might also be needed to interact with other policy systems in inter and intra-domains scenarios.

6.2.4.2 Cognition Orchestrator

The Cognition Engine enables the Artificial Intelligence capabilities in SLICENET, it manages the platform required for running the AI analysis processes that are capable of learning QoE management models and policies and are capable of applying these models autonomously, making decisions that impact the network/slice/service status. The module makes use of defined templates to provide the necessary resources for deploying and managing the analytics and for consuming the data provided by Monitoring Aggregation.

This module provides a data analytics platform for machine learning either inside the provider virtual infrastructure or through 3rd party cloud services. It also provides a scalable platform for inference of analytic learnt models (e.g., large deep learning neural network models) to be used by slice QoE control. This cognitive orchestration functionality might be especially useful when applied at the edge, where resource constraints play a vital role.

6.2.4.3 Analytics

The Analytics functionality provides the Artificial Intelligence capabilities in SLICENET. It is composed of a set of AI processes that are capable of making autonomous decisions that impact the network/slice/service status during the entire slice lifecycle, from planning to deployment and to maintenance. The module must be able to provide a standard methodology, or a set of methodologies, to enable SLICENET to apply several analysis processes over the gathered data. This approach allows a generic module that is able to build predictive models, apply machine learning techniques or perform any other statistical treatment, while delivering the desired output, models that can be used by Artificial Intelligence agents, i.e. Cognition Engine inference platform, implementing a number of specialized analysis processes, which are purpose-built from a common “portfolio” of analysis tools.

For this matter, the Analytics makes use of the defined policies, within the Policy Framework, and the models available to suggest actions to be applied to the network/slice/service and in a later stage to apply them autonomously. In an even later stage, Analytics are applied to the Cognition Engine, Policy Framework and even the Analytics Engine itself, in order to self-adjust and self-optimize rules that will lead to new analytic processes and to modify existing policies. This set of AI capabilities makes the Analytic Engine the brain of the Intelligence Framework.

6.2.4.4 Aggregator

Monitoring aggregation provides the data pipeline and (big-)data storage for application of cognitive analytics to achieve end-to-end QoE. It acquires data from the service, slice, topology, and resource monitoring modules, prepares the data, and persists it for analytic processing. It must be able to handle high volumes of data both at the transport layer (messaging system) and at the storage system.

The messaging system must be scalable (worker queues and batching), must handle various kinds of data (structured, unstructured) and several methods of interaction (batch queries, streaming, notification). Collection includes several data preparation steps – parsing, shaping/formatting, filtering, validation/cleansing, anonymizing, annotating/enriching, transforming, and more. The messaging system must also apply access control to move data without violating privacy and security permissions. It should support forwarding relevant data to multiple interested subscribers (which may have different ACLs and different processing requirements) including real-time data consumers. The storage system must be scalable and must support efficient data access and queries. At minimum, this includes storing data in time-series databases and indexing.

Monitoring aggregation is driven by policies (via the Policy Framework) and are optimized by the Analytics module; in particular, resource efficiency dictates that only the minimal amount of data that is required for QoE management should be collected and persisted. Choosing the correct data, compacting it through aggregation and discarding unneeded information are an important optimization task and part of the cognitive-based management in SLICENET.

An additional role of monitoring aggregation is to expose slice information to higher layer consumers. Aggregation can be used to hide details, provide processed statistics, and reduce the amount of data reported.

6.2.4.5 QoE/SLA Manager

The role of QoE/SLA Manager is to establish a contract between the network provider and the consumer for a requested service SLA and QoE. While the SLA may be fixed per service type, the QoE can be customized by the customer regarding functional requirements as perceived by the consumer. This role is realized by mapping the desired quality-of-experience into an end-to-end SLA and managing the SLA across all service components. The QoE and SLA shall be technology and provider agnostic to allow service realization across different providers and different administrative domains. The component realizing the QoE management shall apply models and policies, as prescribed by the cognitive plane, to decompose the end-to-end QoE goals into per-slice and per-domain requirements, providing a self-configured concrete plan for the service instance (e.g., through population of configuration parameters in the service template). The plan is deployed and managed by components in charge of service and multi-domain orchestration and the decomposed QoE goals are managed through local domain slice QoE managers and through peer QoE/SLA managers in other domains.

6.2.5 Control Plane

6.2.5.1 Slice QoE Optimizer

Fulfilling the current and future trend of QoE-centric control and management, spurred by the evolving needs of mobile and transport network infrastructures as we move towards 5G, the Service/Slice QoE optimisation functionality is responsible for the maintenance and maximization of the QoE of the deployed NSIs and the services running on top under dynamic infrastructure conditions.

In this regard, the Service/Slice QoE optimisation functionality is intended to provide a per-slice optimisation actuation framework with the scope of QoE guarantees. To this end, it needs to implement several optimisation algorithms with the focus on determining the infrastructure resources and network functions to be deployed, re-configured or released within the bounds defined by the characteristics of the NSI and the current state and utilization of the underlying physical and virtual data plane. In order to achieve this, the management plane defines the overall QoE optimisation goal in terms of declarative policies, that is, a set of operative guidelines that lead the optimisation procedure in deciding the best actions for QoE guarantees. In accordance with the policies, and the outputs determined by the optimisation algorithms themselves, the components implementing the Service/Slice QoE optimisation functionality interact with the core functionalities of the control plane to trigger the (re-)configuration of infrastructure resources and functions to enforce the optimisation decisions.

6.2.5.2 P&P Control

The SLICENET Plug & Play represents a key novelty within the whole SLICENET platform, being it the main enabler for end-to-end slice customization by giving the opportunity to verticals and slice consumers to directly control their slice instances.

The Plug & Play functionalities are intended to provide an on-demand composition, integration and abstraction of slice instance control and management capabilities for direct use of slice consumers. Indeed, the SLICENET Plug & Play control has to be considered as a per-slice instance set of control functions to be consumed, and even augmented with own control logics and tools, by slice consumers. This allows the Plug & Play to be extremely tailored to the vertical needs on the one hand, and to the slice provider non-disclosure policies (in terms of exposing access to internal resources and services towards external entities) on the other.

While these Plug & Play features are included as part of the SLICENET control plane platform, they are intended to expose towards slice consumers both control and management functions:

- very basic monitoring only option, where the slice provider offers only means to monitor slice KPIs (e.g. in terms of performance, resource availability, etc.), while slice configuration and customization is chosen from a catalogue of pre-designed slice templates

- limited control option, where the slice consumer can have also access to a limited set of SDN and NFV control and configuration primitives to customize the slice runtime operation
- extended control option, where the slice consumer can also access the slice instance lifecycle management, thus opening and offering the full operation of the slice

6.2.5.3 Inter-domain Slicing

While an inter-domain service can have multiple components the only truly multi-domain component that requires coordination is the network part. So, an inter-domain service is composed as an intra-domain service, with the SLICENET software being aware of the junction points between the adjacent operators. These junction points are involved in the service as they are specified in the inter-domain slice. Within an autonomous system, the local domain orchestration makes the necessary orchestration for the network resource coordination on handover of data across domains as they are decomposed into components for component activation. The concatenation of the components when active (including VNFs, network graphs, and value-added connectivity) becomes the end to end multi domain service.

6.2.5.4 Intra-domain multi-tenant slicing

The “intra-domain multi-tenant slicing” functionality is thought to arrange tasks based on the enforcement of rules and policies required for the Life Cycle handling of logical and physical resources in end-to-end slices within the same administrative domain.

The main scope is to facilitate any additional service decomposition and orchestration (upon the ones already performed at upper layer) that should be conditionally triggered towards the subnet adapters intended to control the Network Function (NF) and Transport Node (TN) resources.

The main functional scope is the following:

- manage the reception of slice subnet descriptors from the upper “cross-plane slice orchestrator” that is decomposing an e2e slice;
- perform a conditional mapping of subnet descriptors into actions towards the subnet adapters interfacing the “intra-domain 5G RAN-Core slicing” component.

This functionality of the control plane provides the flexibility means such to control the programmability of shared subnets data plane resources for achieving partitioning of NFs in network slices and their chaining into network services. Such a handling of virtual and physical network resources aims to provide the network operations that are characterizing a slice, like the support of multi-tenancy and the isolation of characteristics (performance, security) among different instances of slices and services.

6.2.5.5 Intra-domain 5G RAN Core Slicing

RAN slicing is a challenge with respect to providing different levels of isolation and sharing for slice owners who wish to customize their service across UP and CP, and also to increase the resource utilization of RAN infrastructure. In what follows next, Control Logic (CL) refers decisions making for a particular CP/UP function, e.g. about handovers performed by the CP. RAN functions are pipelined to compose the desired RAN module, i.e., monolithic or disaggregated RAN instances, either via multiplexed or customized CP and UP functions as per slice requirements. The runtime system acts as the intermediate between customized slices and underlying shared RAN module and infrastructure providing a unified execution environment with substantial flexibility to achieve the required level of isolation and sharing.

6.2.6 Data Plane

6.2.6.1 Vertical Data Plane

The Vertical Data Plane segment is integrated in the SLICENET architecture as it is the link that provides the extension of the data plane functionality, responding to the request that SLICENET acts an end-to-end facilitator for the services that are expected to be consumed by the vertical's customers, as a slicing-friendly 5G integrated infrastructure.

6.2.6.2 5G RAN-MEC Data Plane

The Low Latency MEC (LL-MEC) platform is made up of two main components: the LL-MEC platform and data-control APIs. The LL-MEC provides two main services: i) ***native IP-service endpoint*** and ii) ***realtime radio network information to MEC applications*** on per user and service basis, and can be connected to a number of underlying RANs and CN gateways. The control and data plane APIs act as an abstraction layer between the RAN and the CN data plane, and the LL-MEC platform. The OpenFlow and FlexRAN protocols facilitate the communication between the LL-MEC and underlying RAN and CN. With LL-MEC, coordinated RAN and CN network applications can be developed on the top of LL-MEC SDK allowing to monitor and control not only the traffic but also the state of network infrastructure. Such applications could vary from elastic application that obtain user traffic statistics, to low latency applications that redirect user traffic (local breakout) and apply policies to setup the data path. All the produced RAN and CN data and APIs are open to be consumed by 3rd parties.

6.2.6.3 MEC-Core Data Plane

The data plane between the edge network and the core network is the backhaul link of the 5G infrastructure. The programmability of this link will be investigated to allow QoS assurance in this segment to contribute to meeting the end-to-end QoS requirements from the slice-based service. The monitoring capabilities in this segment in terms of QoS metrics will also be explored, and the requirements for sensors to be deployed in the data plane will be examined.

In the MEC segment itself, a MEC subsystem will be established, including the MEC data plane (as part of the backhaul), MEC platform and MEC Apps hosting, in line with the ETSI MEC standard [ETSI-MEC].

6.2.6.4 Inter-Domain Data Plane

The data plane connectivity between Telecommunications Operators is already well-established business and technical agreement, what SLICENET addresses is added features, namely in traffic differentiation in these inter-connection points.

To ensure that data plane traffic does not lose accorded SLA network behavior parameters the border devices must be able to support some level of programmability to allow addition of queues for prioritization, the marking of packets with the appropriate traffic differentiation flags and the traffic routing to the proper destination to take advantage of pre-established internal mechanisms such as MPLS systems.

6.3 Reference Points, Interfaces/API Definitions

As SLICENET addresses 5G slicing concepts, the 5G reference architecture model, as defined in terms of service-based interfaces within the Control Plane and reference points among Network Functions, is followed for defining a similar model for the various functions that are provided by SLICENET sub-planes but it is also expanded by the introduction of a service/information bus that is expected to allow for high granularity and agility based on the concept of a loosely coupled service architecture.

According to this pattern, SLICENET architecture defines a service bus (Figure 19) that intends to provide a means for dynamic registration, discovery and utilization among all planes' service/functional components. During Slice orchestration and operation, a set of services are expected to be involved for the provision of the NSI specific features. This service micro-orchestration (intra/inter-plane) is expected to be triggered by use of business/domain process execution language derivative in the context of an intra/inter-plane workflow execution framework. In this way, the various planes will be possible to be extended with future functions that can be part of workflows through a dynamic discovery mechanism.

Considering the modularity that Slicing might be requiring, an early assumption is that SLICENET sub-planes will be implemented in a modular way so that the enclosed function blocks/objects can be instantiated per Slice Instance (NSI) allowing for high granularity and sufficient scalability as well as for a dynamic and dedicated management space per NSI. In such case the service/information bus concept aligns perfectly with the design approach, but even if the modular approach will not be a design choice and the sub-planes will be implemented in a monolithic way, i.e. one function block for all NSIs, the service/information bus can also cater for the support of the communications and service invocations among the functional blocks. In the simplest case the service/information bus can be based on a modern message queue/streaming platform.

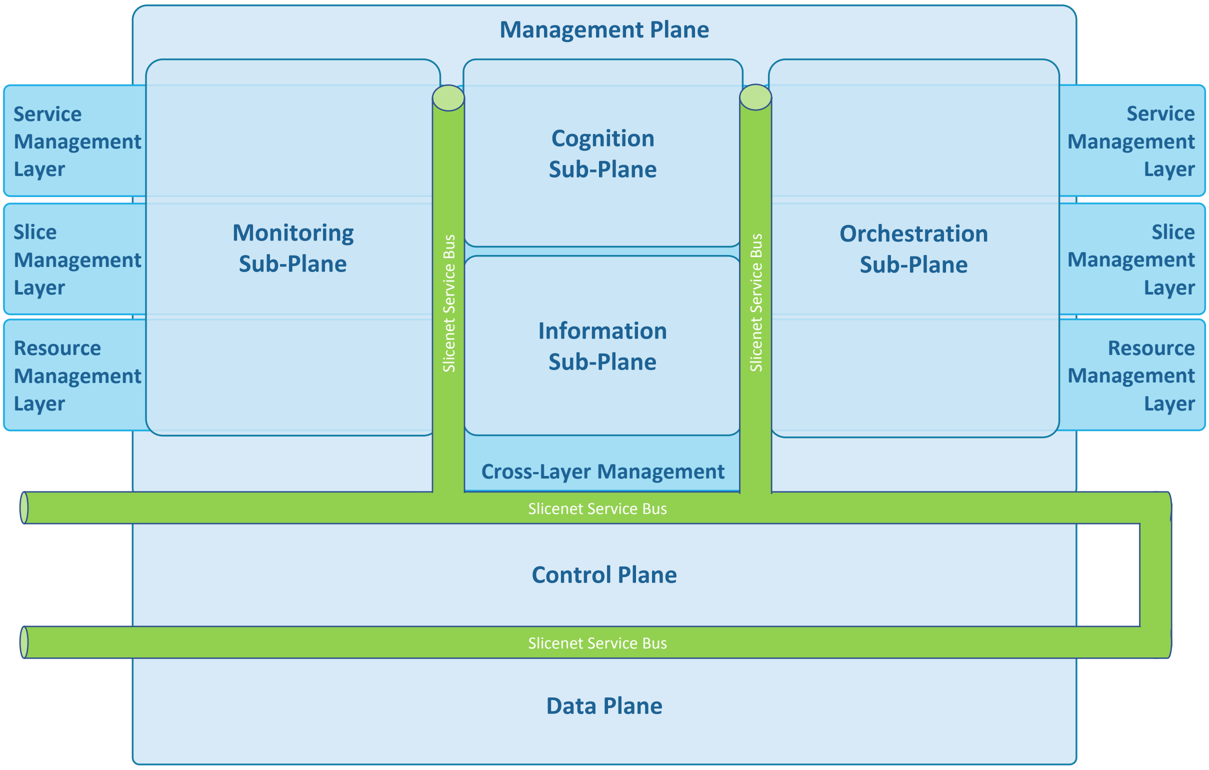

Figure 17: Architecture Service Bus

In order to enable this forward compatible approach, however, there is need for defining abstract reference points per every function inside every plane. The reference points need to be compliant with an expandable information model so that future extensions can be accommodated easily in subsequent workflow definitions. Reference points are considered bidirectional so that they can both consume from and provide services to other functions according to the execution of the workflows into which they are involved.

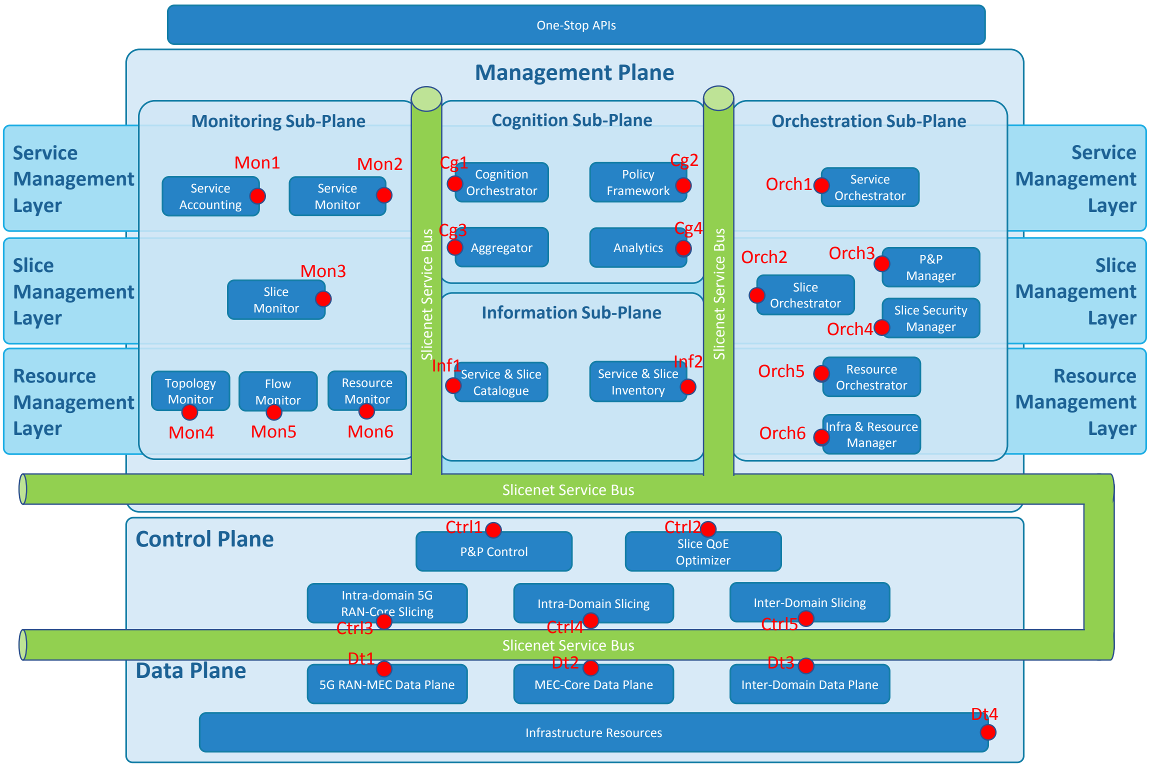

Figure 18: Reference Points

In the above figure (Figure 18) a depiction of the foreseen reference points is highlighted by use of red circles at the edges of function blocks. During workflows execution the reference points are utilized through the service bus to invoke functionality and retrieve information in the context of the supported information and functional model that has been instantiated in the workflow semantics. Reference points can be subject to discovery in case the modular approach is followed. In this context queries identifying the function group along with a parameters set can be used to specialize the resolution of the candidate function instance. For example, a workflow may be requiring ***monitoring functionality,*** regarding ***topology information*** with specific ***flow*** characteristics. This leads to the resolution of a function instance from the Topology Monitoring function block inside the Monitoring Plane that provides flow based support. Once the function instance is identified, it is defined as the service endpoint to be invoked during the related step in the workflow execution. The function supports internally the mapping of the service model onto the internal implementation details. In the case of the monolithic design of the function blocks, the service endpoint is unique, however, slice specific information might need to be resolved before the actual service invocation.

Reference points and functions will be subject to be administered under the command of the One-Stop API in the context of the horizontal management space (6.3.1) for the provision and support of NSSs per administrative domain.

As slice operation is expected to span across different administrative domains, an inter domain Service Interface (SI) is defined. This interface is expected to allow the utilization of the services that are provided by NSSIs in one domain, by the NSSIs that are instantiated in other domains supporting Inter-domain communication among control components during slice operation.

6.3.1 One-Stop API

The One-Stop API will be arranged into two management spaces. A vertical one intended to be utilized by application and use case verticals and a horizontal one to be utilized by administrative domains. The difference between the views produced for each of the two spaces relates with the roles assigned to the users. One role is provided with the service characteristics (offerings) to be used for selection among the slice templates whereas the other relates with the way these offerings are provided in terms of integration of available resources.

In order for this separation to be possible, it is necessitated that each Network Function is represented also in two different ways (Figure 19):

- one listing the technical details (Fault, Configuration) that need to be considered for combining an NF with other NFs for the definition of one NSS. These details will be aggregated at the NSS level to allow proper inclusion of the NSS into NSs.
- and the other listing the qualitative details that can be used to influence the features of any higher level synthesis an NF is related.

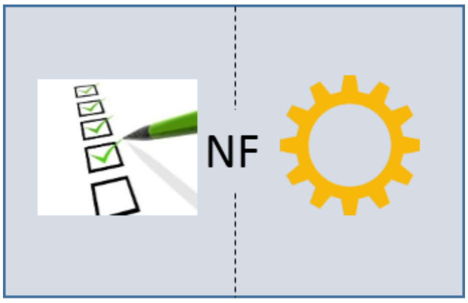

Figure 19: NF Dual View

This dual view model is inherited to higher level aggregations that in turn are producing the result of the synthesis of the contained NF views with any restrictions or dependencies resolved. For example, an NF with an upper limited feature that is combined with another NF with this feature being not limited will result into an NSS with this characteristic dictated by the “less” capable ingredient (Figure 20).

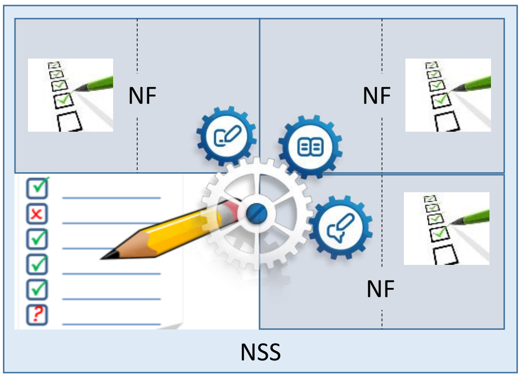

Figure 20: Aggregation of NFs in NSS

Finally, a Network Slice Template is created following the same aggregation approach over NSSs. In this context of horizontal management the One-Stop API should provide the following functionalities:

- NF registration and listing
- NF sharing and FCAPS annotation
- NSS synthesis from existing NFs, aggregation of contained FCAPS
- NF and NSS time-frame based reservation
- Service based cross domain advertisement and discovery
- NS Template compilation from intra/inter-domain NSSs
- NS Template sharing and FCAPS annotation
- Definition of NF/NSS/NS instances provisioning/configuration/maintenance/deletion workflows

The One-Stop API, as used by application and use case verticals (domain customers), will be based on the provisioning of a marketplace populated with NS Templates as these can be listed in the context of the features they support. The service offerings via the marketplace will be defined by each administrative domain on the basis of the available NS templates from NSS templates supported by the specific domain but also from services that can be used from other administrative domains. Service selection will be provided along with a number of configurable parameters that are exposed by NS and NSS templates. Thus the customer will be able to define location, performance, QoS, QoE and scaling parameters for the customization of the ordered service instance. The One-Stop API will then transfer the service request to the slice orchestration sub-plane (Figure 21).

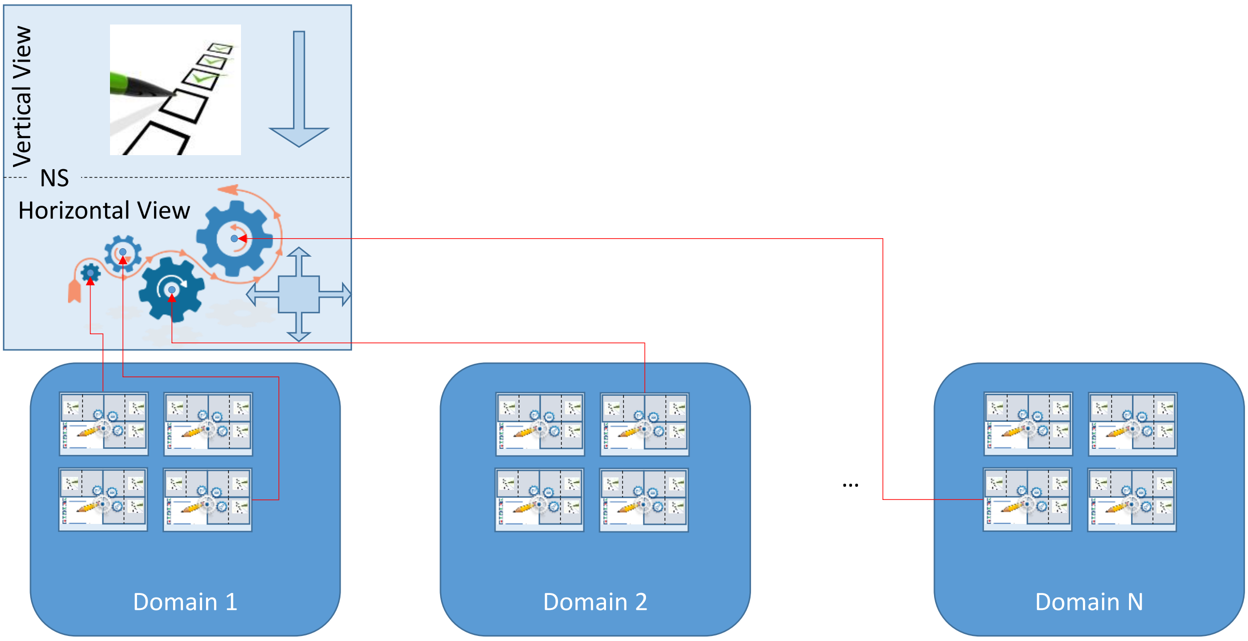

Figure 21: NSI Selection and Provisioning

6.4 Functional Architecture

This section provides a set of high-level workflows across the logical architecture components identified in section 6.2. The objective is not to provide an extensive list of workflows covering all the possible technical use-cases of the SLICENET architecture, but rather to exercise the main set of functional flows.

6.4.1 Preparation Phase Workflows

This section describes the workflows related with the Network Slice preparation phase, i.e., the phase in which the Network Slice is designed, onboarded to the system and the required network resources are deployed.

6.4.1.1 Single-Domain Network Slice Design & Onboard

Designing the Network Slice as a Service (NSaaS), including all its resource dependencies, configurations, produced KPIs, etc., and onboarding it to the architecture is the foundation workflow of the SLICENET architecture. This workflow is represented in Figure 22 and described in Table 33.

This workflow assumes, to avoid going into too much detail, that previous design and onboard workflows for the resources and network slices have already taken place. Therefore, resource and network slice templates are already onboarded to the architecture.

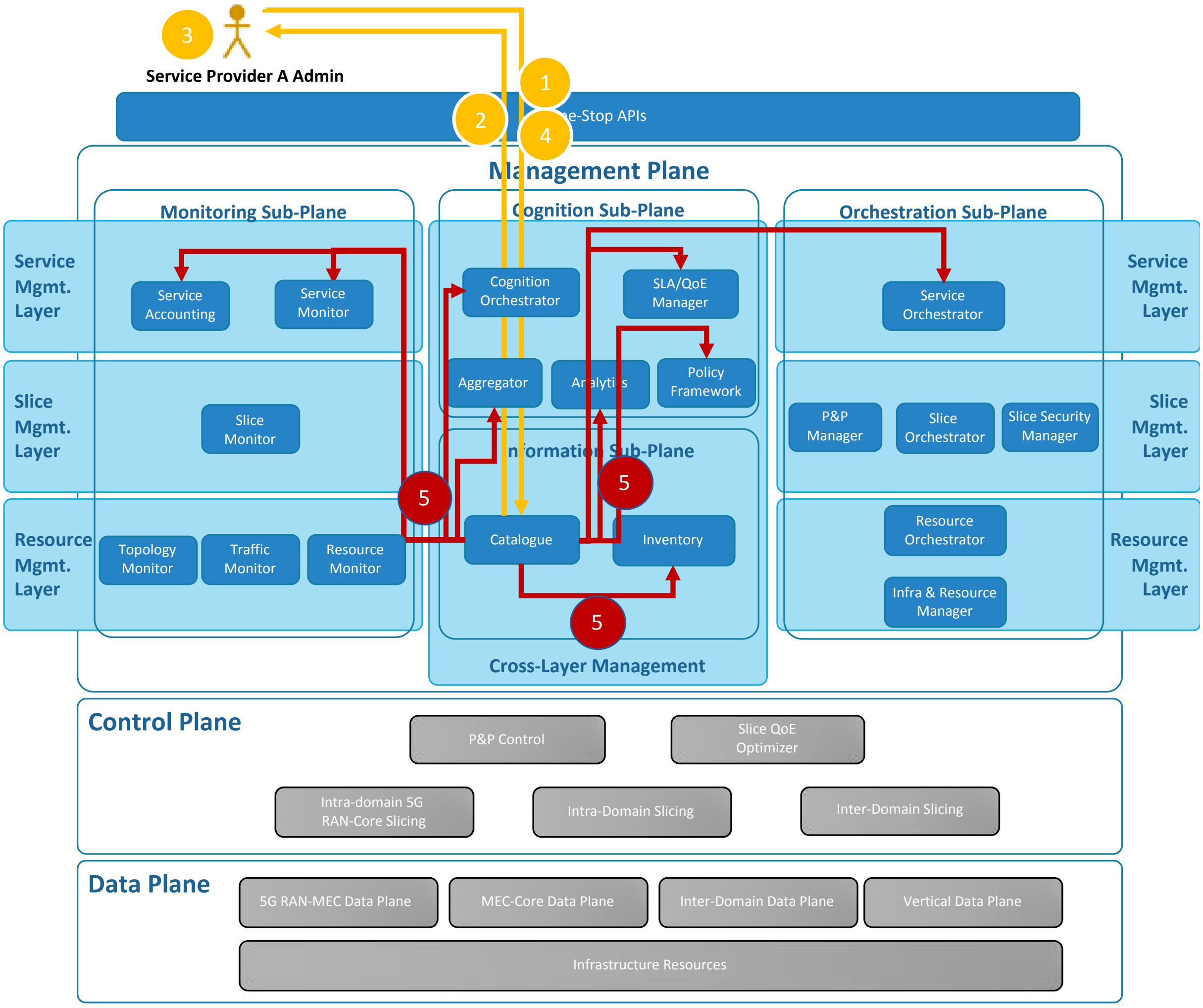

Figure 22: Single-Domain Network Slice Design & Onboard Workflow

Table 33: Single-Domain Network Slice Design & Onboard Steps

Step	Description
1	Service Provider (SP) Admin needs to design and onboard a new Network Slice as a Service (NSaaS) template descriptor (e.g. NSaaS X). The rationale behind the need to design the Network Slice as a Service template might be triggered by business strategies or due to internal requirements of network operation. To proceed with the NSaaS design, the SP Admin requests the Service & Slice Catalogue information about the onboarded resources and Network Slices templates.
2	Service & Slice Catalogue provides the requested information (resources and Network Slices templates onboarded) to the SP Admin.
3	Based on the received information, the SP Admin proceeds with the NSaaS design.
4	SP Admin uploads to the Service & Slice Catalogue the design NSaaS template.
5	Service & Slice Catalogue publishes the onboarded NSaaS (NSaaS X) information to the required architecture components. For example:

	- Service Monitor: information about the NSaaS logs, counters, events, etc. that must be collected, as well as how they should be collected; information about the NSaaS SLA KPIs that should be reported to the subscriber; - Service Accounting: information about the usage parameters that should be collected; - Service Orchestrator: information about the new NSaaS that the Service Orchestrator that the SO should be prepared to process; - Cognition Orchestrator: orchestrate ML tasks for new NSaaS (initial models, refinements); - Aggregator: define data queries/subscriptions for new NSaaS; aggregate data for NSaaS design; - Analytics: compute parameters/models for new NSaaS;

6.4.1.2 Multi-Domain Network Slice Advertisement

As soon as the NSaaS template (NSaaS X) is onboarded to the Service & Slice Catalogue, the later advertises it to the peer domains. Here it is assumed that previous partnerships were already made between the SPs and information about how to publish and/or subscribe the NSaaS templates is already provisioned in the architecture. This workflow is represented in Figure 23 and described in Table 34.

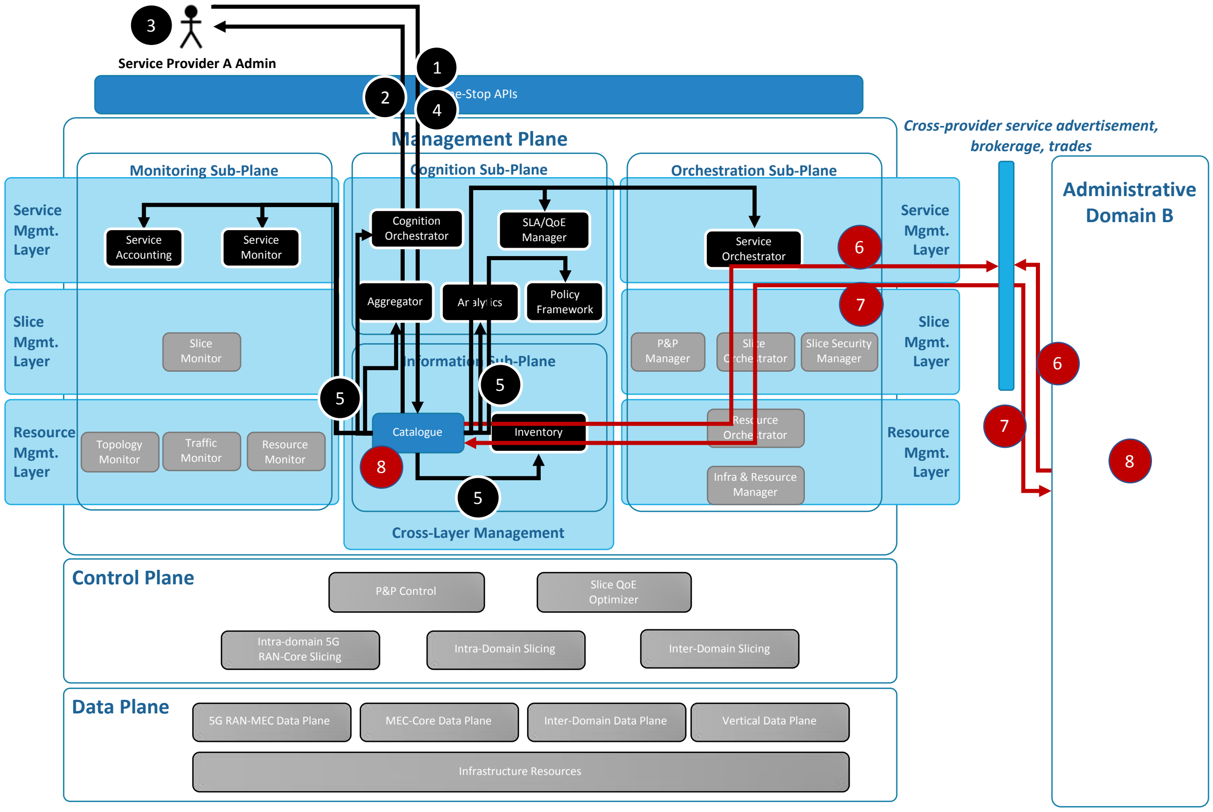

Figure 23: Multi-Domain Network Slice Advertisement Workflow

Table 34: Multi-Domain Network Slice Advertisement Steps

Step	Description
6	Due to the update of a new NSaaS template (NSaaS X) on the Service & Slice Catalogue, the latter advertises (e.g. through a Pub/Sub mechanism) it to the administrative domains that belong to the previously established partnership. The published NSaaS template contains all the required information that the peer domain needs to subscribe and monitor the performance of the NSaaS. The peer administrative domain does the same procedure to advertise its own NSaaS offer (e.g. NSaaS Y).

7	The SP A receives the NSaaS template (NSaaS Y) offer provided by administrative domain B and stores it on the Service & Slice Catalogue.
8	The collected NSaaS offer (NSaaS Y) is stored on the Service & Slice Catalogue to be used on a multi-domain NSaaS design and onboard process (section 6.4.1.3).

6.4.1.3 Multi-Domain Network Slice Design & Onboard

After receiving an NSaaS offer (NSaaS Y) from a peer administrative domain, the Service & Slice Catalogue on SP A has its own NSaaS offer (NSaaS X) and the peer domain NSaaS offer (NSaaS Y). Based on the updated information from the peer domains available on Service & Slice Catalogue, the SP A Admin decides to design and onboard a new, multi-domain, NSaaS offer (NSaaS XY) composed by the NSaaS offer from SP A (NSaaS X) and the NSaaS offer from SP B (NSaaS Y). The workflow steps for this case are exactly the same as the ones presented and described for the single-domain NSaaS offer (NSaaS X) depicted in section 6.4.1.1. After designing and onboarding the multi-domain NSaaS offer (NSaaS XY), the Vertical customer is able to request its instantiation.

6.4.2 Subscription Phase Workflows

6.4.2.1 Multi-Domain Network Slice Instantiation

After the preparation phase, the onboarded NSaaS templates are ready to be instantiated by the Vertical customers. Figure 24 illustrates the workflow and Table 35 describes the related steps that take place when a Vertical subscribes a NSaaS.

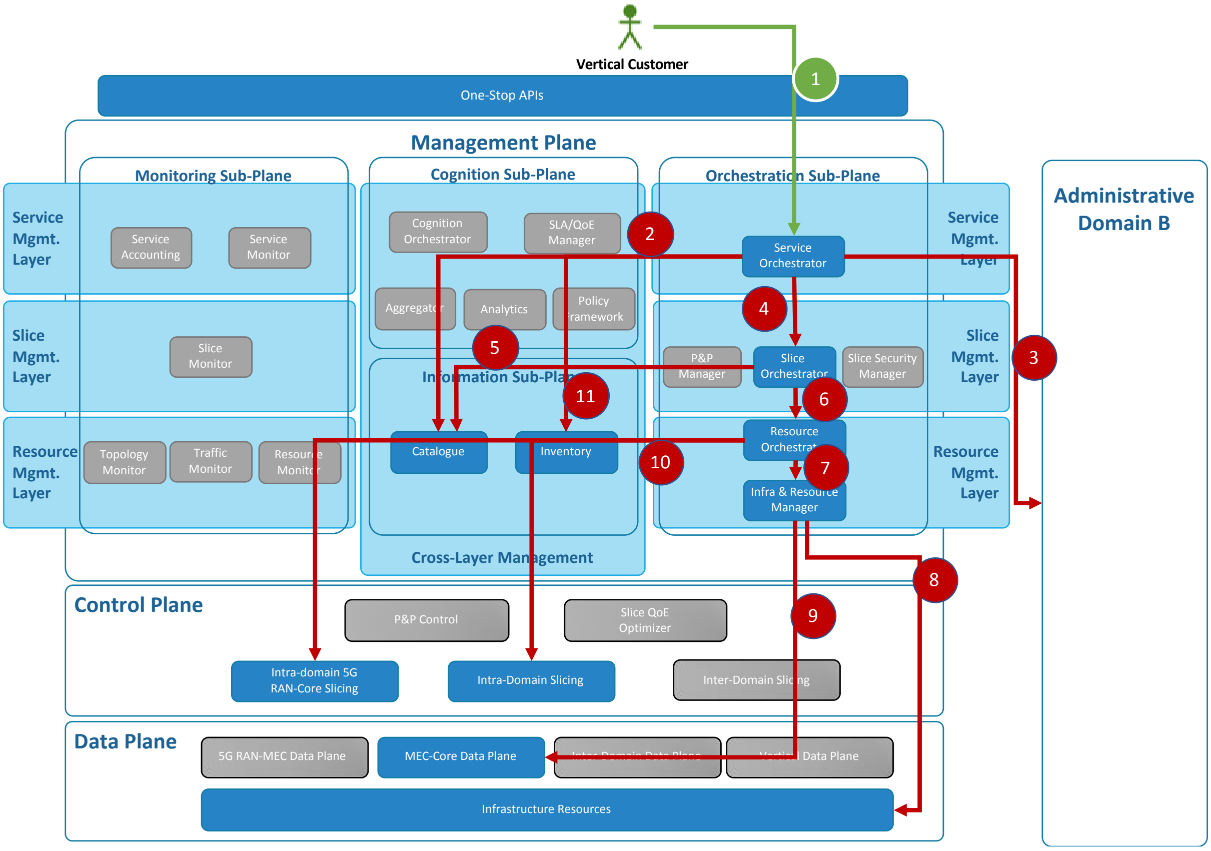

Figure 24: Multi-Domain Network Slice Instantiation Workflow

Table 35: Multi-Domain Network Slice Instantiation Steps

Step	Description
1	Vertical customer subscribes the multi-domain NSaaS offered by SP A (e.g. NSaaS XY)
2	Service Orchestrator decomposes the multi-domain NSaaS XY offer in the Service & Slice Catalogue to obtain the intra-domain NSaaS and NSI that compose the offer, as well as the inter-domain NSaaS offered by another administrative domain. For example: • NSaaS XY = NSaaS X + NSaaS Y; • NSaaS X @ SP A Domain = NSI X; • NSaaS Y @ SP B Domain;
3	Service Orchestrator requests the instantiation of NSaaS Y at SP B domain. SP B receives the instantiation request and proceeds with the required internal workflows to deploy the NSaaS Y.
4	Service Orchestration in SP A, in parallel or sequentially with respect to the NSaaS Y instantiation in SP B, requests the Slice Orchestrator the instantiation of the intra-domain NSI X.
5	Slice Orchestrator decomposes the NSI X in the Service & Slice Catalogue to obtain the NSSIs and resources that compose the NSI X. For example: • NSI X = NSSI X_1 + NSSI X_2; • NSSI X_1 = PNF A + VNF B; • NSSI X_2 = VNF A + SDN-Application A;
6	Slice Orchestrator requests the Resource Orchestrator the instantiation (if not yet deployed) and configuration of the required resources to fulfill the NSSI X_1 and NSSI X_2.
7	For the resources not yet deployed, the Resource Orchestrator requests their instantiation to the Infra & Resource Manager.
8	Infra & Resource Manager allocates the required virtual resources on the infrastructure.
9	Infra & Resource Manager deploys the VNFs on the allocated virtual resources.
10	Resource Orchestrator configures the resources that compose the NSSI X_1 and NSSI X_2. **NOTE**: the resources configuration might also be done directly from the Slice Orchestrator component (it is still under discussion and the final decision will be reported in subsequent deliverables).
11	Service Orchestrator registers all the information related with the NSaaS XY instantiation in the Service & Slice Inventory. **NOTE**: the registration of the slice instantiation might also be done by the Slice Orchestrator component (it is still under discussion and the final decision will be reported in subsequent deliverables).

6.4.2.2 Multi-Domain Network Slice Plug & Play Instantiation & Configuration

The Multi-Domain Slice Plug&Play instantiation and configuration takes place soon after the MD Slice is instantiated, therefore the execution of the Multi-Domain NSaaS Instantiation workflow has to be considered as a pre-requisite and starting point.

For each new Multi-Domain slice instance, a dedicated Multi-Domain Plug&Play instance has to be created and properly configured to activate specific vertical-tailored control functions and expose customized control and management primitives to the vertical customer.

Figure 25 illustrates the Multi-Domain Plug&Play instance activation and configuration workflow over the logical architecture and Table 36 describes the related steps.

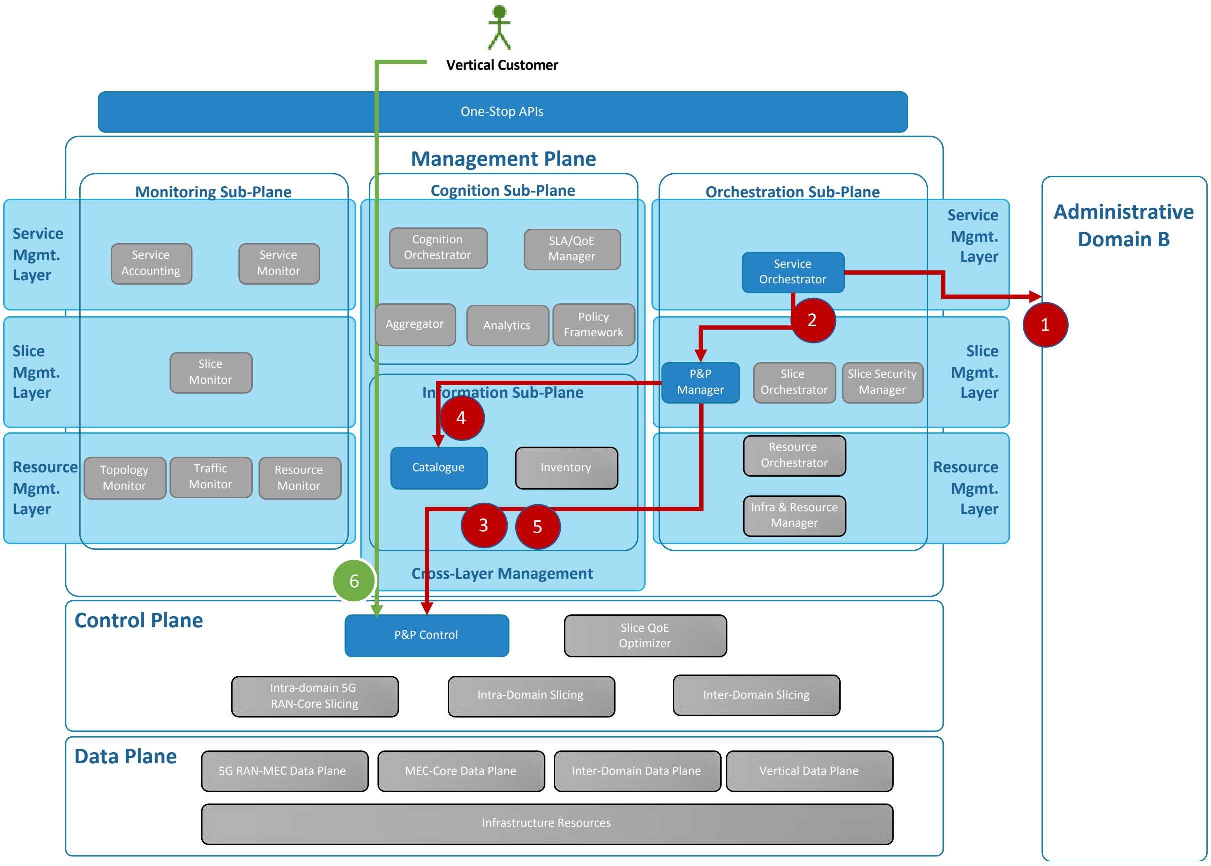

Figure 25: Multi-Domain Network Slice Plug & Play Instantiation & Configuration Workflow

Table 36: Multi-Domain Network Slice Plug & Play Instantiation & Configuration Steps

Step	Description
1	As soon as the Multi-Domain NSI is successfully created (i.e. after step 11 of Multi-Domain NSaaS Instantiation workflow – ref. section XYZ), the Service Orchestrator retrieves from each other administrative domain involved in the Multi-Domain slice information about Plug&Play primitives and capabilities exposed for the given slice instance. This translates into the collection of control and management APIs and related endpoints that each other administrative domains offers for the given NSSI part of the end-to-end Multi-Domain NSI
2	The Service Orchestrator invokes the Plug&Play Manager with the aim of activating a new Plug&Play instance for the end-to-end Multi-Domain NSI. The activation request includes the information collected at step 1 from other administrative domains, so that the Plug&Play manager is aware of the APIs and endpoints that the new Plug&Play instance will be allowed to access in other administrative domains. Moreover, the request may include additional information related to vertical requirements or customization needs expressed at the Multi-Domain NSaaS Instantiation request
3	The Plug&Play manager, as the entity responsible for the lifecycle management of Plug&Play control

	functionalities and instances within the SliceNet platform, create a new instance of Plug&Play control, in the form a generic instance without any customization or tailored control function
4	When the new Plug&Play instance (with generic configuration) is ready, the Plug&Play manager retrieves from the Service & Slice Catalogues the needed information to customize the Plug&Play control instance according to the Plug&Play capabilities expressed in the Service and Slice templates to which the given multi-domain slice instance refers. In practical terms, this operation allows the Plug&Play manager to select which control and management functions activate on the Plug&Play instance, enabling the dedicated drivers as well as configuring the vertical tailored Plug&Play abstraction features. This customization is also integrated and merged with those capabilities and APIs offered by other administrative domains (step 1) and with any additional requirement expressed by the vertical when requesting its slice (as considered in step 2)
5	The Plug&Play manager applies the vertical tailored configuration of the Plug&Play control instance taking two major actions: i) activation of the specific drivers to allow the Plug&Play control instance to have access to the SliceNet control and management functionalities (including those offered by other administrative domains) according to the information retrieved in step 4, ii) configuration of vertical specific abstraction features, iii) activation of control and management APIs and endpoints dedicated for the given end-to-end NSI and to be exposed to the vertical customer
6	Once the Plug&Play control instance is activated and configured, the vertical customer can access it to apply its own control logics to the NSI at runtime

6.4.2.3 Single-Domain Network Slice SLA/QoE Management

Figure 26 and Table 37 describe the network slice instantiation workflows and steps, respectively, with QoE guarantees.

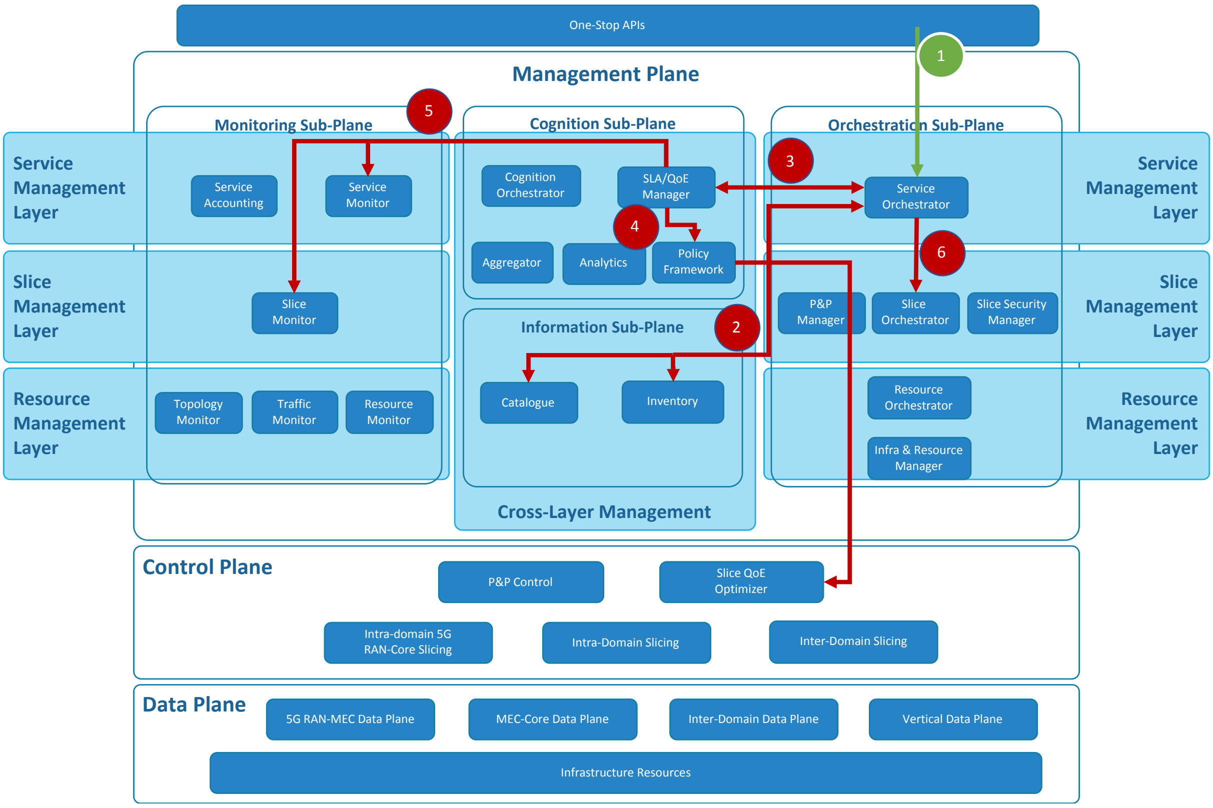

Figure 26: Single-Domain Network Slice Instantiation with QoE Guarantees Workflow

Table 37: Single-Domain Network Slice Instantiation with QoE Guarantees Workflow Steps

Step	Description
1	A Vertical requests a NSI with QoE guarantees.
2	The Service Orchestrator consults the catalogue and inventory to obtain the NSI template and NSSIs.
3	The Service Orchestrator interacts with the SLA/QoE manager to decompose the end-to-end QoE requirements into SLA and QoS configuration parameters for each NSSI.
4	The SLA/QoE Manager updates the Slice QoE optimizer via the Policy Framework with specific policies to support the per NSSI QoS and end-to-end QoE as detailed in the Single-Domain Network Slice Management Closed Loop flow.
5	The SLA/QoE Manager updates the Service and Slice Monitors to collect the necessary KPIs to enforce per-NSSI QoS and to analyze the service end-to-end QoE as detailed in the Single-Domain Network Slice Monitoring & Reporting flow
6	Slice instantiation continues as the non-QoE case, orchestrating the decomposed template as NSSIs

6.4.3 Run-time Phase Workflows

6.4.3.1 Single-Domain Network Slice Monitoring & Reporting

After the NSaaS is instantiated, it should be monitored and reporting information delivered to both the vertical and/or the SP. Figure 27 illustrates the workflow and Table 38 describes the related steps that take place during this flow.

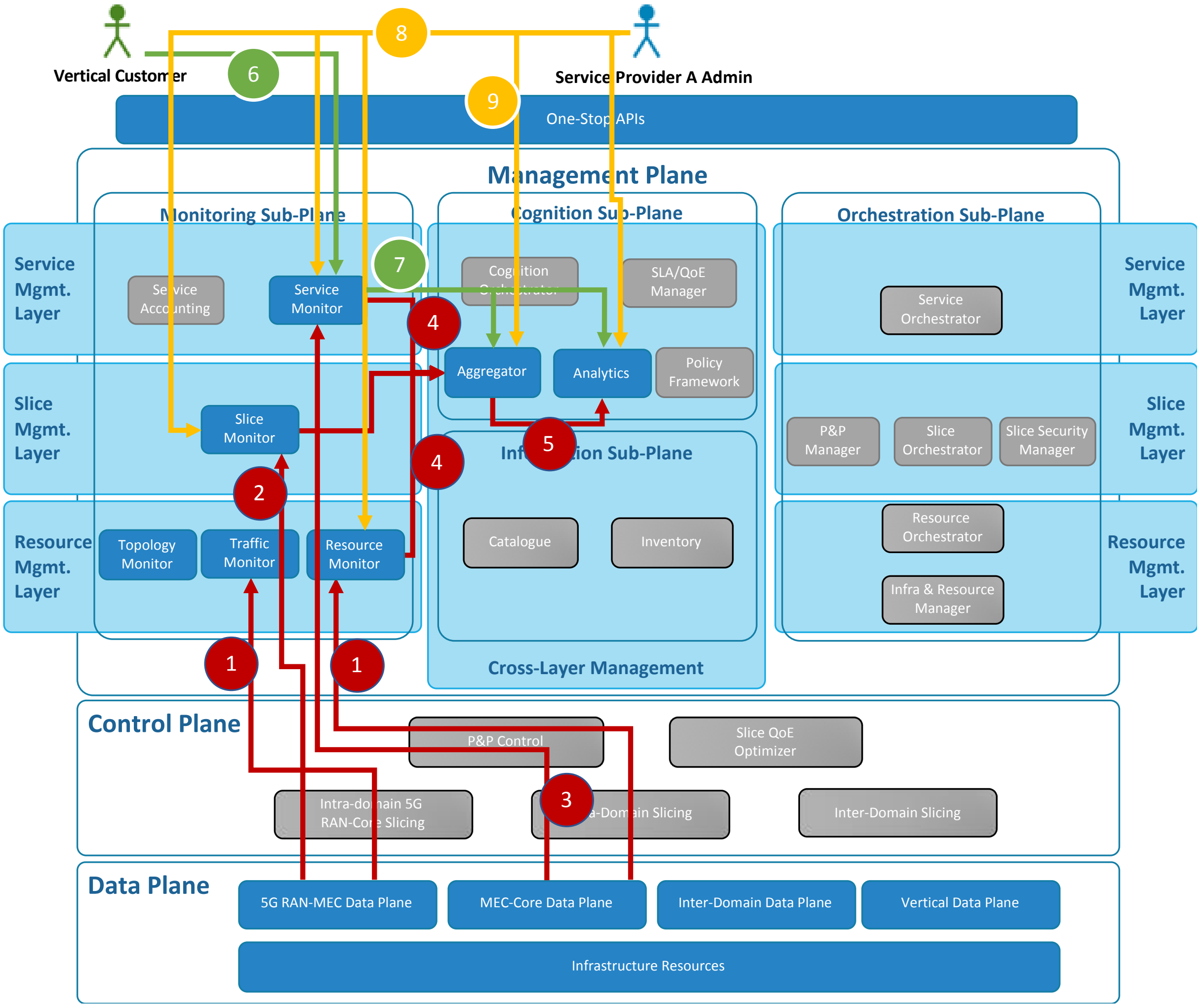

Figure 27: Single-Domain Network Slice Monitoring & Reporting Workflow

Table 38: Single-Domain Network Slice Monitoring & Reporting Steps

Step	Description
1	Resource Monitor collects and persists counters, flows, events and/or alarms from the network resources;
2	Slice Monitor collects and persists counters, events and/or alarms from the slice;
3	Service Monitor collects and persists counters, events and/or alarms from the service;
4	Based on configuration information during the NSaaS onboarding phase, the Aggregator processes and aggregates counters, events and/or alarms from the Resource Monitor, Slice Monitor and Service Monitor. In the end of the aggregation process, either in batch or in streaming, NSI, NSSI and NSaaS-level Key Performance Indicators (KPIs) will be produced indicating the performance of each one of these abstractions. Additionally, and also based on information provided during the NSaaS onboarding phase, a threshold engine is configured to trigger an alarm when the produced KPIs cross a specific value. The produced alarms can be consumed by several entities of the SLICENET architecture (e.g. Analytics).

5	Analytics processes aggregated information and provides advanced reports and dashboards – e.g., trends, top-*, predictions, etc. Another important example is to use analytics to create and report SLA information.
6	Vertical customer requests the Service Monitor information about the KPIs status that were agreed in the SLA contract; this information flow might also involve the P&P control component in order to retrieve and provide the monitoring information to the vertical.
7	Service Monitor retrieves the KPIs information from the Aggregator and/or Analytics (or from the persistence area in which the KPIs are stored), filters and presents them to the Vertical;
8	SP Admin requests the Resource, Slice and Service Monitor information about the counter, events and/or alarms at each one of these abstractions;
9	SP Admin requests the Aggregator and/or Analytics information about the produced KPIs. In this case, there is no filtering and full information is provided to this actor (SP Admin).

6.4.3.2 Single-Domain Network Slice Cognition

Figure 28 and Table 39 illustrate and describe a potential cognition workflow within the SLICENET system architecture.

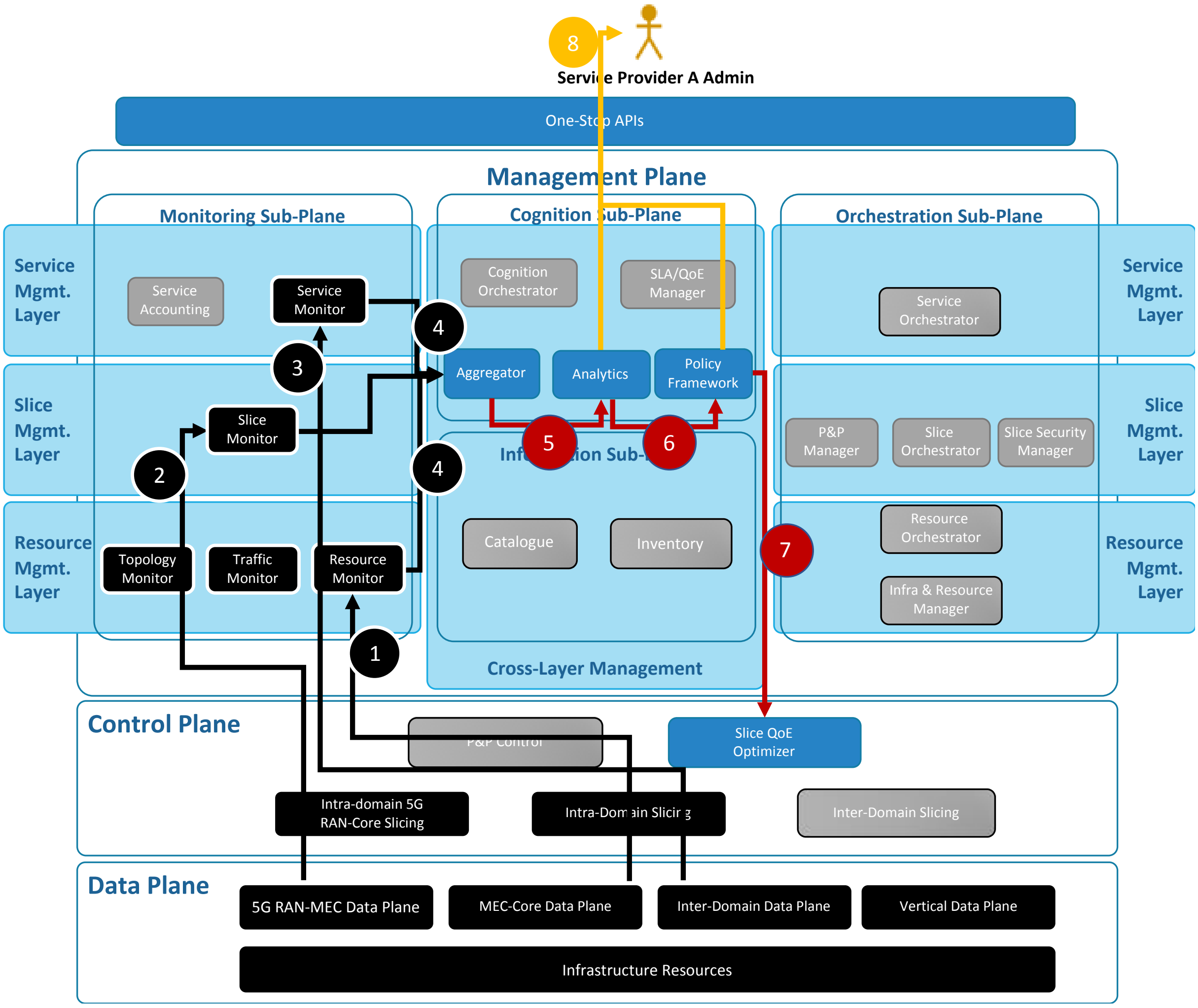

Figure 28: Single-Domain Network Slice Cognition Workflow

Table 39: Single-Domain Network Slice Cognition Steps

Step	Description
1 – 5	Single-Domain Network Slice Monitoring & Reporting workflow described in section 6.4.3.1
6	Based on aggregated information, the Analytics runs the intelligence/learning algorithms and outputs new/updated policies towards the Policy Framework.
7	If the new/updated policies impact control-level policies, these are delivered to the control-plane components that implement the control closed-loop (Slice QoE Optimizer).
8	SP Admin might request information about the intelligence procedures (from the Analytics) or about the new/updated policies (from the Policy Framework).

6.4.3.3 Single-Domain Network Slice Management Closed Loop

During run-time, policy-driven, management closed loops will be in place to optimize the NS. Figure 29 and Table 40 illustrate the management closed loop workflow and steps description, respectively.

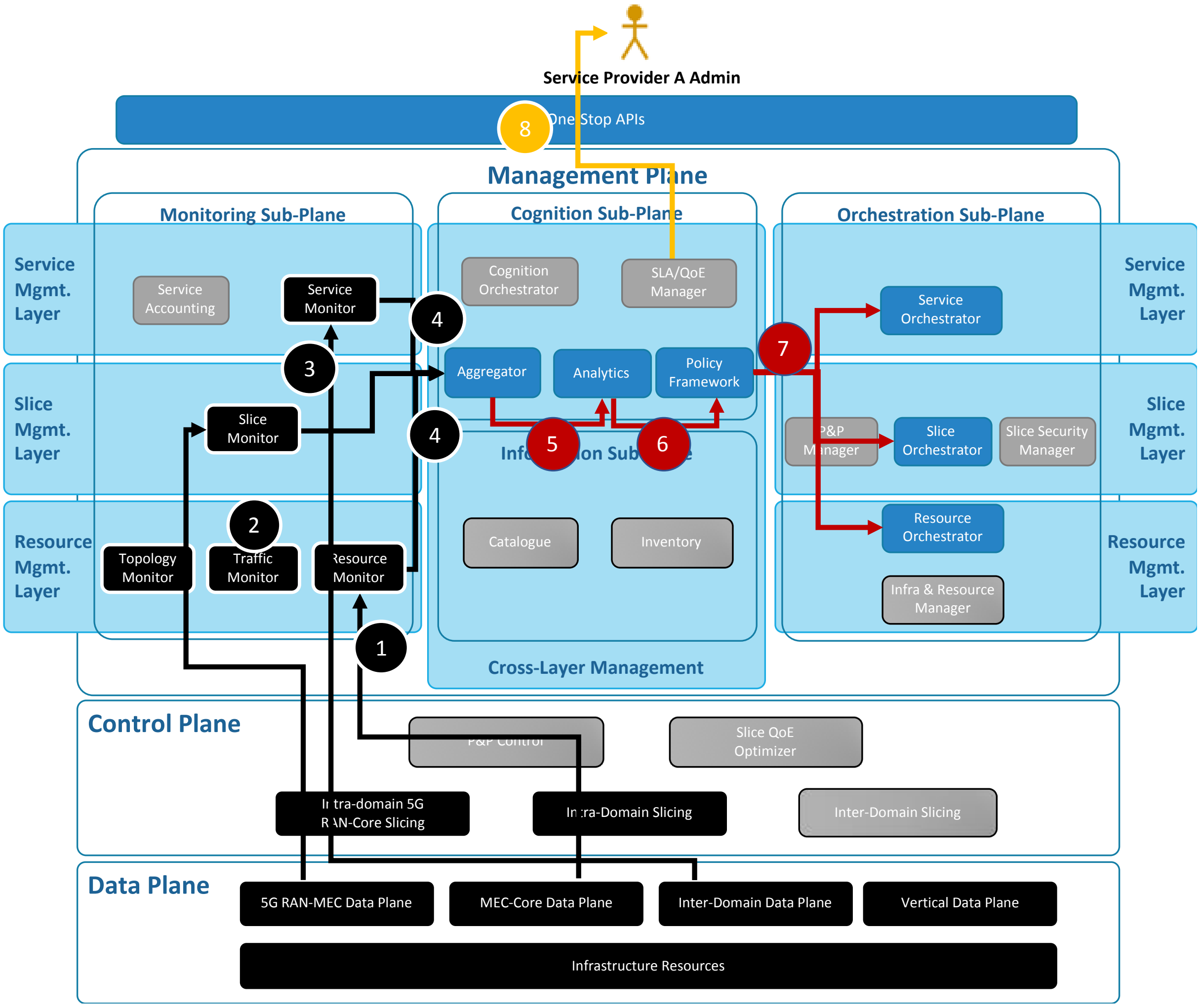

Figure 29: Single-Domain Network Slice Management Closed Loop Workflow

Table 40: Single-Domain Network Slice Management Closed Loop Steps

Step	Description
1 – 5	Single-Domain Network Slice Monitoring & Reporting workflow described in section 6.4.3.1.
6	Analytics output triggers a well-known policy within the Policy Framework engine.
7	As a result, the Policy Framework engine applies an action. The action might impact, among others, the Service Orchestrator, the Slice Orchestrator or the Resource Orchestrator.
8	SP Admin is updated when a specific policy is applied.

6.4.3.4 Single-Domain Network Slice Control Closed Loop

Besides the management closed loops described in section 6.4.3.3, policy-driven closed loops also take place at the control plane level. Figure 30 and Table 41 illustrate the control closed loop workflow and steps description, respectively.

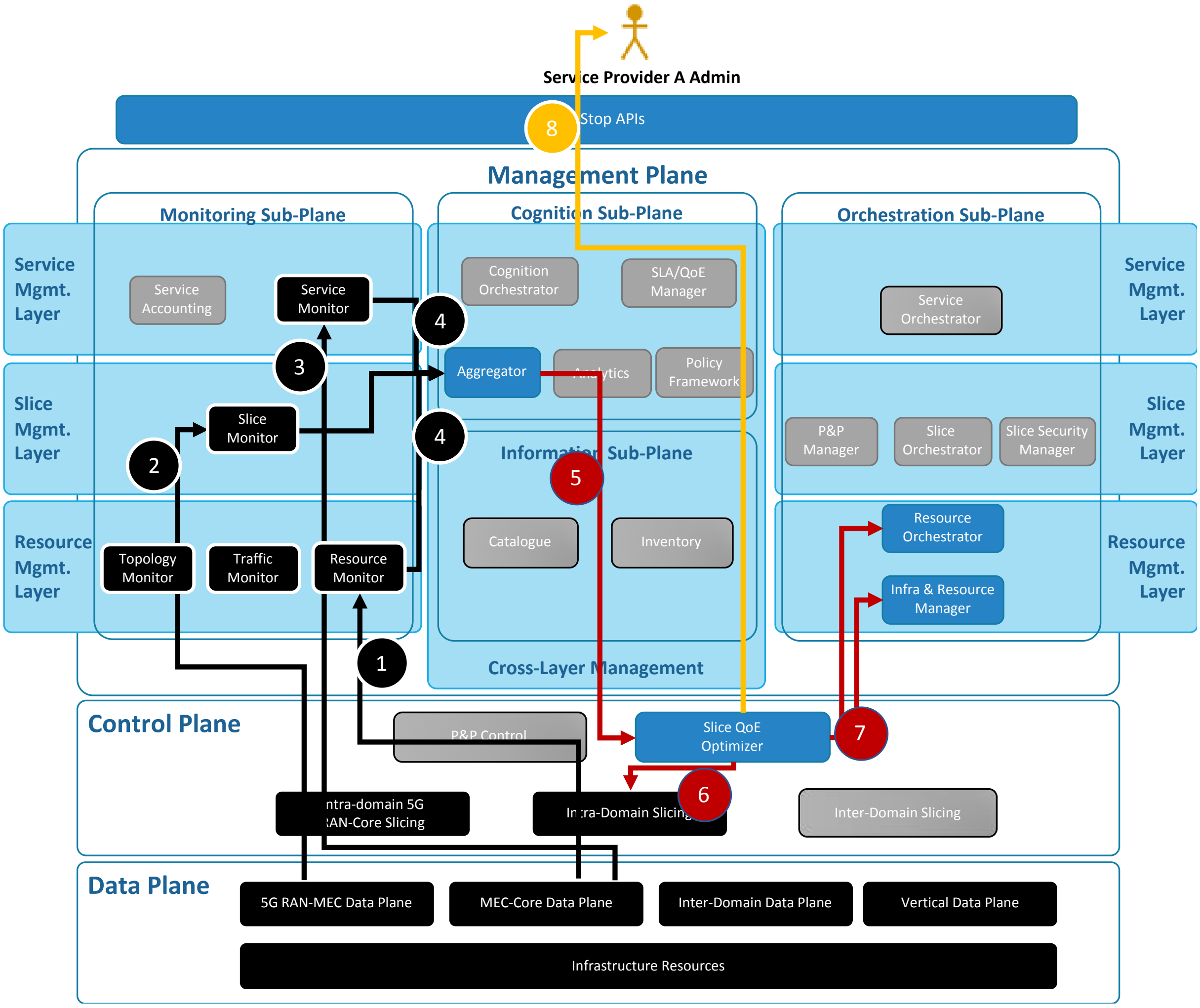

Figure 30: Single-Domain Network Slice Control Closed Loop Workflow

Table 41: Single-Domain Network Slice Control Closed Loop Steps

Step	Description

1 – 5	Single-Domain Network Slice Monitoring & Reporting workflow described in section 6.4.3.1.
6	Aggregator output triggers a well-known policy within the Slice QoE Optimizer.
7	As a result, the Slice QoE Optimizer applies an action, which might impact, among others, the Resource Orchestrator or functions at the control and/or data-plane level.
8	SP Admin is updated when a specific policy is applied within the Slice QoE Optimizer.

7 Conclusions

The main objective of this work is to present the SLICENET overall architecture and define the main interfaces by using a structured approach where related academia and industry work is taken into consideration. Regarding the latter, three main areas were investigated: 5G-PPP projects, open-source projects and standardization activities. On 5G-PPP, phase I and phase II projects were considered and compared to SLICENET, identifying possible synergies to take advantage of current and future outcomes of these exploration and innovative projects. Also, taking into account the industry increased adoption of open-source technologies in production and pre-production environments, several projects were investigated regarding their functionality and future use on SLICENET. Concluding the state-of-the-art chapter on this deliverable, standardization work was also addressed by looking into relevant work in SDOs. The latter is of special relevance given the importance of standardization on the telecommunications industry where SLICENET intends to provide meaningful contributions.

The business roles and stakeholders in SLICENET are also defined in this deliverable, which are based on the work provided by ITU on the subject and within the scope of network slicing based services. Moreover, the proposed roles and stakeholders model was also validated using the SLICENET use-cases, previously presented in Deliverable D2.1. This validation analysis provides an operational context for the SLICENET framework, which with the inclusion of architecture considerations and requirements from the technical use-cases point-of-view, serve as foundational aspects for the architecture design work. An additional note regarding the architecture considerations chapter, here innovative cornerstone aspects from the SLICENET Framework are described with the intent of shedding some light on the approach taken by the SLICENET project for the management of services based on network slices.

Finally, chapter 6 provides the main outcome of this work by providing the SLICENET logical and functional architecture, including the main reference points. The SLICENET logical architecture identifies logical components and general functionalities that need to be addressed by the project to achieve its objectives. Moreover, a high level functional architecture is also provided, which materializes the logical architecture previously mentioned. This chapter also shows how the logical components are mapped into functional ones, since some of the former are the result of the interwork between two or more functional components.

References

[1] 5GEx Project, http://www.5gex.eu/

[2] COHERENT Project, http://www.ict-coherent.eu/

[3] SELFNET Project, https://selfnet-5g.eu/

[4] COGNET Project, http://www.cognet.5g-ppp.eu/

[5] 5GNORMA Project, https://5gnorma.5g-ppp.eu/

[6] Open Network Automation Platform (ONAP), https://www.onap.org

[7] OpenBaton, http://openbaton.github.io/

[8] OpenSourceMano (OSM), https://osm.etsi.org/

[9] OPEN-O, https://www.open-o.org/

[10] Open Network Automation Platform (ONAP), https://www.onap.org

[11] OpenBaton, http://openbaton.github.io/

[12] Open Source Mano (OSM), https://osm.etsi.org/

[13] OPEN-O, https://www.open-o.org/

[14] OpenMano, https://github.com/nfvlabs/openmano

[15] OpenStack, https://www.openstack.org/

[16] OPNFV, https://www.opnfv.org/

[17] Tacker, https://wiki.openstack.org/wiki/Tacker

[18] TeNOR (T-NOVA), http://www.t-nova.eu/

[19] Docker, https://www.docker.com/

[20] Juju, https://jujucharms.com/

[21] Kubernetes, https://kubernetes.io

[22] ONOS, http://onosproject.org/

[23] OpenDaylight, https://www.opendaylight.org

[24] Ryu, https://osrg.github.io/ryu/

[25] OpenContrail, http://www.opencontrail.org/

[26] Nagios Core, https://www.nagios.org/projects/nagios-core/

[27] TensorFlow, https://www.tensorflow.org/

[28] Zeppelin, https://zeppelin.apache.org/

[29] Spark, https://spark.apache.org/

[30] sFlow-RT, http://www.sflow-rt.com/index.php

[31] SkyDive, http://skydive.network

[32] Jupyter, https://jupyter.org/

[33] Ceilometer, https://github.com/openstack/ceilometer

[34] Monasca, http://monasca.io/

[35] ClickOS, http://cnp.neclab.eu/clickos/

[36] DPDK, http://dpdk.org/

[37] IO Visor Project, https://www.iovisor.org

[38] ONIE, http://onie.org/

[39] OpenContrail vRouter, http://www.opencontrail.org/

[40] OpenFastPath, http://www.openfastpath.org

[41] P4FPGA, http://p4fpga.github.io/

[42] SONiC, https://github.com/Azure/SONiC

[43] OpenvSwitch (OVS), http://openvswitch.org/

[44] Mosaic-5G, https://gitlab.eurecom.fr/mosaic5g/mosaic5g

[45] OpenAirInterface, http://www.openairinterface.org

[46] 3GPP, http://www.3gpp.org/

[47] 3GPP SA1 - Services, http://www.3gpp.org/specifications-groups/sa-plenary/sa1-services/home

[48] 3GPP SA2 - Architecture, http://www.3gpp.org/specifications-groups/sa-plenary/sa2-architecture

[49] 3GPP SA3 - Security, http://www.3gpp.org/specifications-groups/sa-plenary/sa3-security/home

[50] 3GPP SA5 - Telecom Management, http://www.3gpp.org/specifications-groups/sa-plenary/sa5-telecom-management/home

[51] ETSI NFV, http://www.etsi.org/technologies-clusters/technologies/nfv

[52] ETSI MEC, http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

[53] ETSI ENI, http://www.etsi.org/news-events/news/1171-2017-02-new-etsi-group-on-improving-operator-experience-using-artificial-intelligence

[54] IETF/IRTF, https://www.ietf.org/ https://irtf.org/

[55] IETF SFC, https://datatracker.ietf.org/wg/sfc/documents/

[56] IRTF NFVRG, https://irtf.org/nfvrg

[57] IRTF NMRG, https://irtf.org/nmrg

[58] MEF, http://www.mef.net/

[59] IEEE, https://www.comsoc.org/conferences/ieee-nfv-sdn

[60] ITU, https://www.itu.int/en/ITU-T/sdn/Pages/default.aspx

[61] NGMN, https://www.ngmn.org/home.html

[62] TM Forum, https://www.tmforum.org/

[63] ONF, https://www.opennetworking.org/

[64] [ODL] OpenDayLight Project, https://www.opendaylight.org/

[65] [ODL-ARCH] OpenDayLight architecture, https://wiki.opendaylight.org/view/OpenDaylight_Controller:Architectural_Framework

[66] [OPENSTACK] OpenStack Project, https://www.openstack.org/

[67] [OPENSTACK-PN] OpenStack Project Navigator, https://www.openstack.org/software/project-navigator

[68] [ONAP] ONAP Project, https://www.onap.org/

[69] [ONAP-ARCH] ONAP architecture, https://wiki.onap.org/display/DW/Architecture

[70] [OPEN-O] Open-O Project, https://wiki.open-o.org/

[71] [OPEN-O-ARCH] Open-O SUN architecture, https://wiki.open-o.org/display/AR/Architecture+Home

[72] [OPNFV] OPNFV Project, https://www.opnfv.org/

[73] [ECOMP] ECOMP Project white paper, http://policyforum.att.com/wp-content/uploads/2017/03/ecomp-architecture-whitepaper-att.pdf

[74] [OVS] OVS Project, http://openvswitch.org/

[75] [DPDK] DPDK Project, http://dpdk.org/

[76] [MOSAIC] Mosaic-5G Project presentation, https://www.openairinterface.org/docs/workshop/4_OAI_Workshop_20171107/Talks/NIKAEIN_mosaic5g_OAI_WS.pdf

[77] [OAI] OpenAirInterface Project, http://www.openairinterface.org/

[78] [RUNTIME], Technical Report: “R AN slicing runtime system for flexible and dynamic service execution environment” http://www.eurecom.fr/fr/publication/5351/detail/ran-slicing-runtime-system-for-flexible-and-dynamic-service-execution-environment?popup=1

[79] [NGMN] NGMN, Description of Network Slicing Concept, v1.0/1.0.8, Jan/Sept 2016

[80] [3GPP] 3GPP TR 28.801 V1.1.0 (2017-03), Study on management and orchestration of network slicing for next generation network (Release 14)

[81] [IMT2020] ITU IMT-O-047, Draft Recommendation: Network Management Framework for IMT-2020, Dec 2016

[82] [IETF] https://tools.ietf.org/html/draft-flinck-slicing-management-00

Annex A Technical Requirements

A.1 Data Plane requirements

Requirement Id	*Technical Use Cases Requirement tags*	*Slogan*
REQ-DP01	General	The data plane should provide support for resource inventory along with topology and availability information
REQ-DP02	CRTL_MGMT_EXP_NSI_1_DP-01	The data plane should provide support for information modelling of (V)NEs, (V)IT and (V)NFs to expose capabilities to upper layers
REQ-DP03	ORCH_NSI_DP-01 ORCH_NSI_DP-02 CHC_COG_NSI_DP-01 CHC_COG_NSI_DP-02 AOPT_NSI_DP-01 AOPT_NSI_DP-02	The data plane should be able to be configured according to management and control layer guidelines
REQ-DP04	CRT_NSI_DP-01 ORCH_NSI_DP-01 ORCH_NSI_DP-02 CHC_OSA_NSI_DP-02 CHC_COG_NSI_DP-02 ACT_OSA_NSI_DP-02 AOPT_NSI_DP-01	Creation, activation, modification and deletion of physical/virtual data paths should be allowed across the multiple segments of the data plane
REQ-DP05	CRT_NSI_DP-01 ORCH_NSI_DP-01 CHC_OSA_NSI_DP-01 CHC_COG_NSI_DP-01 ACT_OSA_NSI_DP-01 CM_CN_OSA_NSI_DP-01 AOPT_NSI_DP-02	Creation, activation, modification and deletion of physical/virtual IT resources should be supported by the data plane
REQ-DP06	CRT_NSI_DP-01 ORCH_NSI_DP-01 CHC_OSA_NSI_DP-01 CHC_COG_NSI_DP-01 ACT_OSA_NSI_DP-01 CM_CN_OSA_NSI_DP-01 AOPT_NSI_DP-02	Creation, activation, modification and deletion of physical/virtual NFs should be supported by the data plane
REQ-DP07	SM_NSI_DP-01 SM_NSI_DP-02	The data plane should support the deployment of dedicated functions to monitor and react against security threats
REQ-DP08	General	The data plane should support physical or virtual

		isolation of resources assigned to a NSI to allow coexistence of multiple concurrent NSIs
REQ-DP09	AHEA_NSI_DP-01 AHEA_NSI_DP-02 AHEA_NSI_DP-03	The data plane should support protection mechanisms for automatic failure recovery at the data layer
REQ-DP10	AHEA_NSI_DP-01 AHEA_NSI_DP-02 AHEA_NSI_DP-03	The data plane should provide the necessary support for cross-layer failure recovery if control and/or management layers involvement is necessary
REQ-DP11	FM_NSI_DP-01 PFM-ML_DP-01	The data plane should support the generation of alarms related to the health of physical/virtual elements and functions
REQ-DP12	AM_NSI_DP-01 PM_NSI_DP-01 DCC_DP-01 QoE-CML_DP-01	The data plane should support the generation of statistics to account for the utilization and performance of physical/virtual elements and functions
REQ-DP13	CRT_NSI_MO-DP-01 CRT_NSI_MO-DP-02	The data plane should support the interconnection of NSIs/NSSIs across multiple administrative domains
REQ-DP14	CRT_NSI_MO-DP-03	The data plane should support monitoring/probing mechanisms to account for failure and performance metrics across multi-operator NSIs/NSSIs
REQ-DP15	CRTL_MGMT_EXP_NSI_1_DP-02	The data plane should allow for exposure of configuration APIs to upper layers
REQ-DP16	CM_RAN_OSA_NSI_DP-01	The data plane should support (re-)configuration of RAN parameters and NFs, either physical or virtual
REQ-DP17	RM_NSI_MO_DP-01	The data plane should support resource and NF (re-) configuration for data/function mobility across multiple operators with a roaming agreement
REQ-DP18	CTRL_MGMT_EXP_NSI_2_DP-01	The P&P control instance is given access to programmable NF APIs for configuration purposes

A.2 Control Plane requirements

Requirement Id	*Technical Use Cases Requirement tags*	*Slogan*
REQ-CP01	CRT_NSI_CP-01 ACT_OSA_NSI_CP-01	For each of the subnets constituting a given Network Slice being created, activated, modified and deleted, • the Control Plane shall use the Network Slice Subnet (NSS) descriptors, as received by the cross-plane orchestrator, to trigger the Network Slice Subnet Instance (NSSI) creation, activation, modification and deletion towards the underneath Subnet northbound interfaces.
REQ-CP02	CRT_NSI_CP-02 ACT_OSA_NSI_CP-02	The Control Plane network slice subnet functions shall provide means for the creation, activation, modification and deletion of a Network Service as translated from the received NSS descriptor.
REQ-CP03	CRT_NSI_CP-02	The Control Plane network slice subnet functions shall provide means for the partitioning of Network Functions (NF) resources and their chaining into a slice subnet.
REQ-CP04	CRT_NSI_CP-01 ORCH_NSI_CP-03	The Control Plane network slice subnet functions shall support the creation of Transport Nodes by interfacing the related SDN Controller.
REQ-CP05	CHC_OSA_NSI_CP-01	The Control Plane shall support orders over the One-Stop-API (OSA) for the change of capacity per a given Network Slice and this will be executed to take into account the customized policy for the ad-hoc rules defined in Control Plug&Play module.
REQ-CP06	CHC_COG_NSI_CP-01 AOPT_NSI_CP-04, AHEA_NSI_CP-01, ORCH_NSI_CP-02, ORCH_NSI_CP-03	The Control Plane shall trigger the updating of the NSSIs by the underneath Subnet northbound interfaces upon ordered logics coming from Cognitive through cross-plane orchestration. This will be executed also by taking into account the customized policy for the ad-hoc rules defined in Control Plug&Play module.
REQ-CP07	CHC_OSA_NSI_CP-02, CHC_COG_NSI_CP-02, AHEA_NSI_CP-02	For a given NSSI, the Control Plane network slice subnet functions shall orchestrate the configuration change action by identifying NFs to either re-configure or add/remove.
REQ-CP08	CHC_OSA_NSI_CP-03, CHC_COG_NSI_CP-03	In case of shared NF among NSIs, the Change Capacity order shall be refused if there is a negative impact on other NSIs.

REQ-CP09	CTRL_MGMT_EXP_NSI_1_CP-01	a correspondent set of control primitives for NFs and slice KPIs monitoring is activated and exposed to slice customer
REQ-CP10	CTRL_MGMT_EXP_NSI_2_CP-01, ORCH_NSI_CP-01	the P&P control instance is given access to a limited set of SDN APIs for customized slice run-time operation
REQ-CP11	CTRL_MGMT_EXP_NSI_2_CP-02, ORCH_NSI_CP-01	a correspondent set of control primitives for programmable NFs configuration, SDN logics, and limited NFV management is activated and exposed to slice customer
REQ-CP12	CTRL_MGMT_EXP_NSI_3_CP-01, ORCH_NSI_CP-01	a correspondent set of control primitives for NFV and slice management/orchestration is activated and exposed to slice customer
REQ-CP13	AOPT_NSI_CP-01, AOPT_NSI_CP-03	The Control Plane shall support orders either over the One-Stop-API (OSA) or from Cognitive plane for the QoE optimization that can lead to modify a deployed NSI to maintain QoE levels.
REQ-CP14	QoE-CML_CP-01	Report SLA control adjustments and known violations
REQ-CP15	QoE-CML_CP-02	Apply model to infer QoE for current KPIs
REQ-CP16	QoE-CML_CP-03	Adjust configuration to achieve QoE
REQ-CP17	QoE_CML-CP-01	CP should report on SLA-related adjustment, observed violations and faults
REQ-CP18	QoE_CML-CP-02	CP should apply classification model (generated as policy by CogI) to infer QoE based on current KPIs
REQ-CP19	QoE_CML-CP-03	CP should set per-QoE configuration (based on classified QoE) and apply the necessary adjustments (e.g., set new KPI thresholds)
REQ-CP20	PFM_ML-CP-01	CP should execute corrective actions based on slice-specific policy. This may require applying a ML model provided as a policy by CogI

A.3 Management Plane requirements

Requirement Id	*Technical Use Cases Requirement tags*	*Slogan*
REQ-MP01	ACT_OSA_NSI_MP-01, ORCH_NSI_MP-01, ORCH_NSI_MP-02	Management plane should support NSCRT_NSI_MOI creation, activation, modification and deletion requests including any of CN or RAN part
REQ-MP02	CRT_NSI_MP-05, ACT_OSA_NSI_MP-01, ORCH_NSI_MP-01, ORCH_NSI_MP-02	Management plane should support NSI creation, activation, modification and deletion requests with sharing options
REQ-MP03	AOPT_NSI_MP-01, AOPT_NSI_MP-02, AHEA_NSI_MP-01, ORCH_NSI_MP-03	Management plane should support inventory of NSI resources along with sharing and reuse options
REQ-MP04	CRT_NSI_MP-06, CHC_OSA_NSI_MP-04, CHC_COG_NSI_MP-04, ACT_OSA_NSI_MP-04, AOPT_NSI_MP-03	Management plane should be able to manipulate CN and RAN resources according to requested LifeCycle actions
REQ-MP05	CRT_NSI_MP-03	Management plane should be able to determine location of CN and RAN resources to be included in NSI
REQ-MP06	CRT_NSI_MP-01, ORCH_NSI_MP-04	MP (OSApi) should allow listing of E2E Service templates to be selected by verticals
REQ-MP07	CM_RAN_OSA_NSI, CM_CN_OSA_NSI, CRT_NSI_MO, MGT_EXP_NSI, CTRL_MGMT_EXP_NSI_1,2,3	MP (OSApi) should allow the creation of NSS templates based on NFs
REQ-MP08	CHC_OSA_NSI_MP-01, CHC_OSA_NSI_MP-02	MP (OSApi) should allow the tuning of an NS template per NSI with respect to the parameters exposed by the NS template
REQ-MP09	CRT_NSI_MP-02, CRT_NSI_MP-03, CRT_NSI_MP-04, CHC_OSA_NSI_MP-03, CHC_COG_NSI_MP-01, CHC_COG_NSI_MP-02, ACT_OSA_NSI_MP-03, AOPT_NSI_MP-01	MP should process NS template details either with the default options or with any modifications and resolve the required NSS to be instantiated, activated, modified and deleted
REQ-MP10	CRT_NSI_MP-03	MP should support location details per NSSI

REQ-MP11	CRT_NSI_MP-03	MP should consider management domain in location resolution
REQ-MP12	CRT_NSI_MP-05	MP should consider sharing options both for identified NSSIs but also NS sharing scheme
REQ-MP13	CRT_NSI_MP-04	MP should resolve conflicts at the NSSI level by the creation of additional NSSI to accommodate NSI request
REQ-MP14	ACT_OSA_NSI_MP-02	NS template should support NSSI activation order
REQ-MP15	FM_NSI_MP-01, FM_NSI_MP-02, FM_NSI_MP-05, PM_NSI_MP-01	MP should support fault and performance management at least as foreseen by the NS template. E.g. collect specific metrics and alarms indicated in the template and compare them to thresholds
REQ-MP16	PM_NSI_MP-02, AHEA_NSI_MP-03, AHEA_NSI_MP-04	MP should be able to identify the fault and performance management metric sources and orchestrate appropriate configuration or resource provisioning if the existing does not suffice for producing the expected metrics
REQ-MP17	FM_NSI_MP-03, PM_NSI_MP-05, PM_NSI_MP-03	MP should maintain an inventory with all instances and related configurations and also with all fault and performance changes
REQ-MP18	CRT_NSI_MP-05	MP should mark configurations according to sharing options and maintain them until there are no instances referring to them
REQ-MP19	FM_NSI_MP-04, PM_NSI_MP-04	MP should allow fault/performance management based on rule based policies
REQ-MP20	SM_NSI_MP-02	MP should be able to map high level NS security constraints onto underlying NSS and NF specific ones
REQ-MP21	General	MP should be able to provide control intents for several control plane domains (RAN, edge, core, vim)
REQ-MP22	DCC-MP-01	MP should enable configuration of monitoring policy, alerting rules and data security policy
REQ-MP23	AM_NSI_MP-01, AM_NSI_MP-02, AM_NSI_MP-03,	The management plane should support the generation of LifeCycle Management events to account for the utilization

	AM_NSI_MP-04	and performance of physical/virtual elements and functions
REQ-MP24	CTRL_MGMT_EXP_NSI_1-MP-01	Once the NSI is provisioned, the dedicated P&P control instance is activated through the related management plane and orchestration functions
REQ-MP25	CTRL_MGMT_EXP_NSI_1-MP-02, PM_NSI_MP-06	The P&P control instance is given access to APIs for collection of KPIs related to NFs performance
REQ-MP26	CTRL_MGMT_EXP_NSI_1-MP-03, PM_NSI_MP-06	The P&P control instance is given access to APIs for collection of KPIs related to slice performance
REQ-MP27	CTRL_MGMT_EXP_NSI_2-MP-01	The P&P control instance is given access to a limited set of NFV APIs for managing a restricted set of lifecycle aspects of VNFs
REQ-MP28	CTRL_MGMT_EXP_NSI_3-MP-01	The P&P control instance is given full access to NFV APIs for VNF lifecycle management
REQ-MP29	CTRL_MGMT_EXP_NSI_3-MP-02	The P&P control instance is given access to slice management and orchestration APIs
REQ-MP30	SM_NSI_MP-03	The management plane should support physical or virtual isolation of resources assigned to a NSI to allow coexistence of multiple concurrent NSIs
REQ-MP31	SM_NSI_MP-04, SM_NSI_MP-01	The management plane should provide the support for storing access and authentication credentials for security management and for exposing them to upper layers per each NSI/NSSI.
REQ-MP32	SM_NSI_MP-05, SM_NSI_MP-02	The management plane shall be able to notify about security threats in NFs and/or infrastructure resources and it shall propagate to upper layers for the affected NSIs/NSSIs.
REQ-MP33	QoE-CML_MP-01	Define desired QoE via One-Stop-API
REQ-MP34	QoE-CML_MP-02, QoE-CML_MP-03	Define slice SLAs and monitored KPIs, configure monitoring policy
REQ-MP35	QoE-CML_MP-04, QoE-CML_MP-05	Provide QoE labels via One-Stop-API – level of perceived QoE (either from DSP or CSP)
REQ-MP36	QoE-CML_MP-06	decompose, compile, and dispatch labels

REQ-MP37	CM_RAN_OSA_NSI-MP-01, CM_CN_OSA_NSI_MP-01	RAN or NF configuration order from One Stop API for NSI
REQ-MP38	CM_RAN_OSA_NSI-MP-02, CM_CN_OSA_NSI_MP-02	Identification of RAN or CN NSSIs to reconfigure
REQ-MP39	CM_RAN_OSA_NSI-MP-03, CM_CN_OSA_NSI_MP-03	Configuration module triggers RAN or CN NSSI specific configuration actions
REQ-MP40	CM_RAN_OSA_NSI-MP-04	RAN NSSI Configuration Management identifies what to configure: radio interface parameters or RAN virtualized component to deploy
REQ-MP41	CM_CN_OSA_NSI_MP-04	CN NSSI Configuration Management identifies the concerned NFs to configure
REQ-MP42	CM_RAN_OSA_NSI-MP-05	Management of identified configurations
REQ-MP43	CM_CN_OSA_NSI_MP-05	Management of identified configurations by EMSs
REQ-MP44	AHEA_NSI_MP-02, AHEA_NSI_MP-05	The management plane should support protection mechanisms for automatic failure recovery in the data plane.
REQ-MP45	QoE_CML-MP-01	MP One-Stop-API should include defining desired QoE
REQ-MP46	PFM_ML-MP-01	MP One-Stop-API should include defining fault data collection policy and learning goals
REQ-MP47	QoE_CML-MP-02	MP should be able to define slice SLAs and KPIs to be monitored and to set monitoring policy. This may require applying a ML model provided as a policy by CogI
REQ-MP48	QoE_CML-MP-03	MP should provide way to collect QoE labels and app context via One-stop-API
REQ-MP49	QoE_CML-MP-04	MP should decompose received service-level QoE labels to the relevant slices and domains and dispatch the labeling data
REQ-MP50	PFM_ML-MP-02	MP should install proactive fault management goals and actions (policy) for each new instance of slice. This may require applying a ML model provided as a policy by CogI

REQ-MP51	PFM_ML-MP-03	MP should be able to react to fault notification and manage fault mitigation actions (that cannot be remedied by CP), including cross-domain actions
REQ-MP52	CTRL_MGMT_EXP_NSI_3-MP-03	The P&P control instance is given access to slice orchestration APIs

A.4 Intelligence Sub-Plane Requirements

Requirement Id	***Technical Use Cases Requirement tags***	***Slogan***
REQ-COG01	DCC-CogM-01, DCC-CogM-02, DCC-CogM-03, DCC-CogM-04, DCC-CogI-01, AOPT_NSI_CO-01	CogP should provide dynamic on-line monitoring re-configuration: define events to be detected, types of data to be collected, data capturing rules and set frequency in which different KPIs will be measured.
REQ-COG02	DCC-CogI-03	CogP should provide configuration of data processing and filtering methods
REQ-COG03	DCC-CogI-03	CogP should provide data processing, data filtering, data structuring and transformation based on predefined rules
REQ-COG04	DCC-CogM-01	CogP should provide collection of resource metrics in a structured way
REQ-COG05	DCC-CogM-01	CogP should detect events related to state of resources and signal on them based on alerting rules
REQ-COG06	DCC-CogP-02	CogP should auto-detect topology configuration and generate metadata describing the topology
REQ-COG07	DCC-CogP-03	CogP should provide streaming of service-level structured logs
REQ-COG08	DCC-CogP-04	CogP should provide streaming of network flows collected based on pre-defined filtering and frequency policies
REQ-COG09	DCC-CogI-02	CogP should enable receiving data monitored by CogM and other Cognition Aggregators
REQ-COG10	DCC-CogI-02, DCC-CogI-05, DCC-CogI-06,	CogP should enforce security policies

	DCC-CogI-07	
REQ-COG11	DCC-CogI-02	CogP should provide scalable transport pipe
REQ-COG12	DCC-CogI-04	CogP should enable storage of both structured and unstructured data
REQ-COG13	DCC-CogI-04	CogP should provide time-based information storage (history)
REQ-COG14	DCC-CogI-05, FM_NSI_CO-01	CogP should enable to subscribe for alerts and notifications based on subjects of interest
REQ-COG15	DCC-CogI-05	CogP should provide forwarding information to all the interested subscribers
REQ-COG16	DCC-CogI-06	CogP should be able to obtain data through queries from other data aggregators
REQ-COG17	DCC-CogI-07	CogP should provide querying mechanism to provide needed data/query results to different cognition loops, MP and cross domain observers
REQ-COG18	DCC-CogI-05, DCC-CogI-06, DCC-CogI-07	CogP should provide mechanism to enable data transfer cross domain based on predefined permission rules - for example enable removing certain fields or enable data anonymization
REQ-COG19	QoE-CML-CogM-01	Collect and report resource metrics/KPIs
REQ-COG20	QoE-CML-CogI-01	Collect historical KPI data
REQ-COG21	QoE-CML-CogI-02	Label historical data
REQ-COG22	QoE-CML-CogI-03	Learn KPI classification by QoE labels. Learn KPI to QoE correlation
REQ-COG23	QoE-CML-CogI-04	Install classification model (via policy framework)
REQ-COG24	QoE-CML-CogI-05	Refine and update model with new data and labels
REQ-COG25	QoE_CML-CogM-01, PFM_ML-CogM-02	CogM should collect and report all the predefined metrics, KPIs, fault stats and alerts
REQ-COG26	QoE_CML-CogI-01, QoE_CML-CogI-02, PFM_ML-CogI-01	CogI should be able to collect historical data on KPIs, fault stats and labeling

REQ-COG27	QoE_CML-CogI-03	CogI should learn KPI classification by QoE labels and KPI to QoE correlation
REQ-COG28	PFM_ML-CogI-03	CogI should learn minimal set of KPIs required to monitor and control fault rates with acceptable level of probability and create policies and models
REQ-COG29	QoE_CML-CogI-04	CogI should be able to install and continuously refine (based on new data) models for classification of KPIs by QoE labels
REQ-COG30	PFM_ML-CogI-02	CogI should compute and persist model (as policies) of fault probabilities per measured KPIs
REQ-COG31	PFM_ML-CogI-04	CogI should compute KPI to action rules per slice parameters and update per-slice policy
REQ-COG32	PFM_ML-CogI-05	Continuously refine and update model of fault probabilities based on newly received data