r

Deliverable D1.5

5G-VINNI E2E Network Slice Implementation and Further Design Guidelines

Editor:	Wint Yi Poe, Huawei
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	30th September 2020
Actual delivery date:	5th October 2020
Suggested readers:	ICT-19 projects; industry verticals seeking to conduct Testing on 5G systems for Network Slicing; 5G-VINNI WPs dealing with platform implementation for Network Slicing Operation;
Version:	2.0 (of 5G-VINNI deliverable D1.2)
Total number of pages:	71
Keywords:	5G; Architecture; Network Slicing Design & Implementation; Control Mechanisms;

Abstract

Network slicing as a key feature of 5G is supported by 5G-VINNI end-to-end Facility to validate the performance of 5G services and use cases by operating trials required by Verticals. Network Slicing design and supporting systems for 5G-VINNI Facility Sites are mainly based on 3GPP specifications, but also taking into consideration work in other standardization bodies (SDOs) for vertical requirements and 5G evolutions. The previous release of this document contains Network Slicing architecture and supporting systems for 5G-VINNI Facility Sites to be able to implement 5G services with Network Slicing. This document contains the enhancements of 5G Network Slicing and slice operation learned during the implementation of 5G services with Network slicing to Verticals.

[End of abstract]

Disclaimer

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

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP1 Architecture and Design of 5G-VINNI End-to-End Platform
Number and title of task(s)	T1.2 Network Slicing design
Document title	5G-VINNI E2E Network Slice Implementation and Further Design Guidelines
Editor: Name, company	Wint Yi Poe, Huawei
Work-package leader: Name, company	Dan Warren, Samsung

Copyright notice

Executive summary

5G-VINNI aims at deploying large scale and standard compliant 5G systems at its main Facilities Sites, to serve as a platform enabling vertical stakeholders to carry out experimentation focusing on the integration of vertical industries within 5G communication systems. To this end, a common, standard compliant Facility Site reference architecture needs to be specified. The reference architecture must include the specification of the Network Slicing feature, which is seen as the key 5G enabler for verticals integration.

Building upon the design of infrastructure architecture and subsystems and documented in Deliverable D.1.1 [1] and D1.4 [2]) this document provides a second release of the 5G-VINNI Deliverable D1.2 [3] taking into account of the feedback and considerations that have resulted from Network Slicing design and implementation efforts in 5G-VINNI Facility Sites.

This document further defines design considerations and guidelines for end-to-end (E2E) network slicing to be adopted by 5G-VINNI project in Final Release, as well as the real implementation and considerations of Network Slicing Use Cases offered by 5G-VINNI Facility Sites. It specifies 3GPP-compliant network slicing features implemented and deployed in 5G-VINNI end-to-end facility, covering access, transport, and core networks.

Based on the design guidelines provided in Deliverable D1.2 [3], the document provides enhanced isolation and security aspects. Multi-domain and multi-operator scenarios are covered with enhanced network slice federation mechanisms considered in 5G-VINNI facility. The document also provides a top level reference architecture to support vertical stakeholders' communication services spanning across sites.

Research programs described in Deliverable D1.2 [3] are progressed by highlighting the key technologies and research directions for future mobile network architecture.

This deliverable takes as baseline 3GPP release 15 work items (WI) and release 16 study items (SI) specifications. However, as simplifications may be needed, due to implementation and deployment constraints at the Facility Sites, in some aspects 100% 3GPP compliance may be not achieved.

This deliverable is the cornerstone for the detailed design, implementation and deployment of Network Slicing for and across 5G-VINNI Facility Sites. This deliverable may be subject to updates upon feedback arising during 5G-VINNI Release Final implementation and deployment phases.

List of authors

Company	Author
UC3M	Adrián Gallego Sánchez
ICOM	Alexios Lekidis
Eurescom	Anastasius Gavras
Telenor	Andres Gonzalez
Fraunhofer	Arun Prakash
SES	Christos Politis
UoP	Christos Tranoris
AUEB	Costas Kalogiros
Samsung	Dan Warren
ICOM	Dimitrios Kritharidis
SRL	Foivos Michelinakis
AUEB	George Darzanos
Ericsson	Hrvoje Marinovic
Nokia	João António Rodrigues
ALB	Jorge Carapinha
TID	Jose Ordonez-Lucena
Telenor	Kashif Mahmood
SES	Konstantinos Liolis
ICOM	Kostas Chartsias
ICOM	Kostas Stamatis
Fraunhofer	Marius Corici
Telenor	Michael Stornes
Telenor	Min Xie
AUEB	Nausika Kokkini
Telenor	Pål Grønsund
BT	Paul Muschamp
Huawei	Ramin Khalili
Fraunhofer	Santosh Kumar Rajaguru
ICOM	Vasileios Theodorou
Huawei	Wint Yi Poe
Huawei	Xun Xiao

Table of Contents

List of figures

List of tables

Abbreviations

3GPP	3rd Generation Partnership Project
5G	Fifth Generation (mobile/cellular networks)
5G NSA	5G Non-Standalone
5G PPP	5G Infrastructure Public Private Partnership
5G SA	5G Standalone
5G-ACIA	5G Alliance for Connected Industries and Automation
5GC	5G Core
5GS	5G System
AF	Application Function
AI	Artificial intelligence
AMF	Access and Mobility Management Function
AR/VR	Augmented Reality/ Virtual Reality
AS	Access Stratum
AVT	Alternative Virtualization Techniques
BSS	Business Support System
CN	Core Network
CNI	Container Networking Interface
CP	Control Plane
C-RAN	Cloud RAN
CSC	Communication Service Customer
CSP	Communication Service Provider
DECOR	Dedicated Core
DNN	Data Network Name
E2E	End-to-end
eMBB	Enhanced Mobile Broadband
eMBBLLC	Enhanced Mobile Broadband and Low Latency Communication
eNB	enhanced NodeB
EPC	Evolved Packet Core
FlexE	Flexible Ethernet
gNB	next generation NodeB
GSMA	Global System for Mobile Communications
GST	Generic network Slice Template
HDDL	High Density Deep Learning

HPLMN	Home Public Land Mobile Network
IETF	Internet Engineering Task Force
IIoT	Industrial Internet of Things
IoT	Internet of Things
IPSec	IP security
IPUPS	Inter-PLMN User Plane Security
ISG	Industry Specification Group
K8s	Kubernetes
LTE	Long Term Evolution
MANO	Management and Network Orchestration
MEC	Multi-access Edge Computing
mIoT*	Massive Internet of Things
ML	Machine learning
MME	Mobility Management Entity
mMTC*	Massive machine type communication
MTN	Metro Transport Network
NAS	Non-Access Stratum
NEST	Network Slice Type
NF	Network Function
NFD	Node Feature Discovery
NFVI	Network Functions Virtualization Infrastructure
NG-RAN	Next-Generation Radio Access Network
NOP	Network Operator
NPN	Non-Public Network
NRF	Network Repository Function
NSaaS	Network Slice as a Service
NSD	Network Service Descriptor
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
OAI	Open Air Interface
OSM	Open Source MANO
OSS	Operations Support System
OVS	Open vSwitch
PCRF	Policy and Charging Rules Function
PDU	Packet Data Unit

PLMN	Public Land Mobile Network
PNF	Physical Network Function
PNI-NPN	Public Network Integrated NPN
PoP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource Block
RRC	Radio Resource Control
RRM	Radio Resource Management
SBA	Service Based Architecture
SD-WAN	Software Defined WAN
SEPP	Security Edge Protection Proxy
SLA	Service Level Agreement
SLO	Service Level Objectives
SMF	Session Management Function
S-NEST	Standardized NEST
SNPN	Stand-alone NPN
S-NSSAI	Single-NSSAI
SO	Service Orchestration
SPGW	Serving Gateway/PDN Gateway (SPGW)
SST	Slice/Service type
TA	Tracking Area
TMF	Telemanagement Forum
TN	Transport Network
TR	Technical Report
TS	Technical Specification
uCPE	Universal Customer Premise Equipment
UDM	Unified Data Management
UE	User Equipment
UP	User Plane
UPF	User Plane Function
URLLC	Ultra-Reliable and Low Latency Communication
VNFD	Virtual Network Function Descriptor
VPN	Virtual Private Network

WP	Work Package
ZSM	Zero-touch network and Service Management

* The terms mIoT and mMTC are used interchangeable in this document.

1 Introduction

This document is a second release of the 5G-VINNI deliverable '*D1.2 – Design of network slicing and supporting systems v1*' [3].

The first release of the document [3] contains the principles that may guide the design and supporting systems for network slices in 5G-VINNI E2E Facility based on network slicing architecture of 3GPP release 15 work items (WI) and release 16 study items (SI) specifications.

This document addresses (i) *how* network slicing architecture can be designed and realized in 5G-VINNI E2E Facility; (ii) *which* control mechanisms are considered/introduced for inter-slice interconnection; (iii) *which* architecture enhancement of Network Slicing are considered for Private Networks and (iv) *which* security aspects are considered to support 5G communication services with Network Slicing in 5G-VINNI E2E Facility to support project trials as well as ICT-19 projects and industry verticals partners.

This document also provides technologies and research directions led by 5G-VINNI partners as research direction for the future mobile network architecture.

Furthermore, a number of Network Slicing use cases offered by 5G-VINNI (E2E) facility is provided with preliminary gap analysis and assessment of some specific Network Slicing use cases offered by 5G-VINNI Facility Sites.

1.1 Objective of this document

The specific activities of this document include:

- Design and implementation of network slicing architecture in 5G-VINNI facility sites taking into consideration the Network Slice types, roles and tenants in the 5G systems.
- Design and implementation of inter-slice interworking (i.e., Network Slice Federation) between 5G-VINNI facility sites.
- Design and implementation of security and isolation mechanisms for Network Slicing in 5G-VINNI facility sites.
- Identify research directions led by 5G-VINNI partners as research direction for the future mobile network architecture.
- Conduct a gap analysis of Network Slicing use cases offered by 5G-VINNI (E2E) facility.

1.2 Scope of this document

The document is structured as follows:

- Chapter 2 addresses an updated 5G-VINNI Network Slicing architecture with respect to the architecture reported in D1.2 [3] and provides the updates of 3GPP Rel. 16 features for further design guidelines in 5G-VINNI (E2E) facility.
- Chapter 3 identifies network slice design and control mechanisms implemented in 5G-VINNI (E2E) Facility taking into consideration the RAN, Core and Transport networks, as well as Network Slice federation design and mechanisms.
- Chapter 4 contains security aspects considered and implemented in 5G-VINNI E2E Facility to support 5G communication services with Network Slicing.
- Chapter 5 addresses some detailed key technologies and research directions led by 5G-VINNI partners as research direction for the future mobile network architecture.
- Annex A contains a wide range of slicing use cases to the vertical industries and their applications supported by 5G-VINNI E2E facility are provided, along with an analysis of the gaps and an early assessment of the gap type for the currently available use cases.

2 5G-VINNI E2E Network Slicing Architecture

2.1 5G-VINNI E2E Network Slicing Architecture Refinements

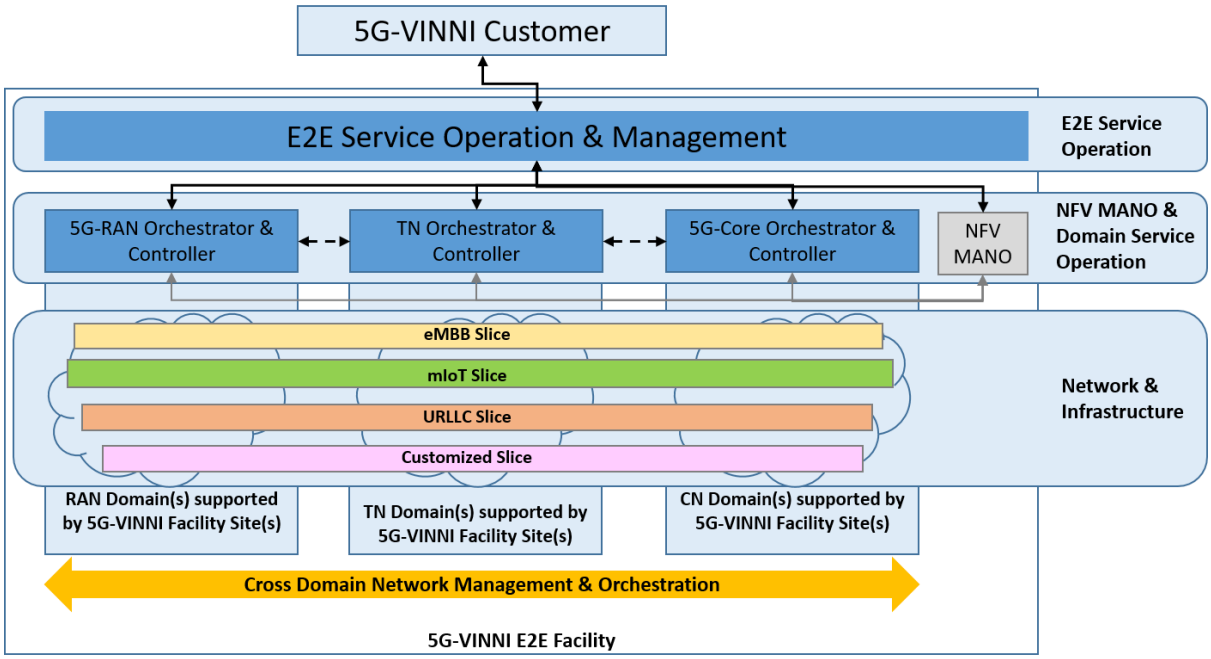


Figure 2-1: An updated 5G-VINNI E2E Network Slicing Architecture

In [3], the high-level architecture of 5G-VINNI E2E Network Slicing is designed based on three network slice type (e.g., eMBB, mIoT, URLLC).

In the updated architecture of 5G-VINNI E2E Network Slicing as presented in Figure 2-1, a customized network slice type is included in addition to eMBB, mIoT and URLLC. This customized network slice type, originally defined in [4], provides an execution environment for the delivery of communication/digital services that do not fall into a single 5G category. The design of a customized network slice responds to the needs of support for

- one service whose performance profile does not fit with any 3GPP 5G defined service category, e.g. IIoT use case for smart metering; or
- two or more services whose individual performance profiles correspond to different 3GPP 5G defined categories.

In the latter case, the slice shall support the establishment of independent PDU sessions, each corresponding to a different service type. For example, a typical case would be an eMBBLLC service, provisioned with two PDU sessions: PDU session #1 (carrying IP packets corresponding to an eMBB service) and PDU session #2 (carrying IP/Ethernet packets corresponding to URLLC service).

The need to differentiate standard slice types (i.e. eMBB, mIoT and URLLC slices) from non-standard slice types (i.e. customized slices) has already been discussed in industry organizations like GSMA. The GSMA Network Group, which is responsible for updating and maintaining Generic network Slice Template (GST) specification [4], has defined in this regard two categories of NEtwork Slice Types (NESTs)[1, 2]: Standardized NEST's (S-NEST's), used for describing the characteristics of network slices

[1] A Network Slice Type (NEST) describes the characteristics of a network slice by means of filling GST attributes with values based on vertical provided service requirements. In essence, a NEST is a filled-in version of GST. Different NESTs allow describing different types of network slices

based on 3GPP 5G service categories; and Private NEST (P-NEST's), aimed at serving industry specific use cases.

5G-VINNI facility leverages the above rationale and categorize the four slice types into two main groups:

- Standard network slice types (i.e. eMBB, mIoT and URLLC slices). 5G-VINNI facility shall have the ability to allow deploying instances of these slices across one or more main facility sites. Therefore, the specification of network slices from this group shall be agreed among main facility sites, for the sake of interoperability across them.
- Non-standard network slice types (i.e. customized slices). Designed for the exclusive usage of an industry vertical, their specification is up to each 5G-VINNI facility site. This means that, unlike the first group, the 5G-VINNI facility does not need to provide multi-site support for the provisioning of customized slice instances.

2.2 Network Slicing Design Guidelines Updates related to 3GPP Release 16

An extensive design guidelines of 3GPP Release 15 Network Slicing, and some consideration of Release 16 work was conducted in 5G-VINNI D1.2 [3].

The new features of 3GPP Release 16 Network Slicing architecture for the Release Final of 5G-VINNI are covered in Section 3.1 of 5G-VINNI D1.4 [2]. In particular, Section 3.1.2 of 5G-VINNI D1.4 [2] discussed the additional or enhanced features in 5G systems based on specifications of 3GPP Release 16 as further Network Slicing design guidelines for the 5G-VINNI Final Release.

[2] As specified in D3.1 [4], VINNI-SB leverages the GST/NEST definition.

3 Network Slice Design and Control Mechanisms implemented in 5G-VINNI (E2E) Facility

3.1 5G System Design: Architectural and Functional Considerations

This section addresses 5G System architectural and functional considerations on 5G Core, RAN and Transport Network (TN).

3.1.1 5G Core Network Functions Design Considerations

Through the development of 5G-VINNI several directions for the design of the 5G Core Network Functions has been considered. In this context, it has been identified at least the following steps for network slice design:

- **Slice Composition**, i.e. what network functions are used in the slice and which needs to be dedicated to the slice.
- **Network Slice Distribution**, i.e. what network functions must be distributed due to low latency or high availability requirements.
- **Network Slice Type Selection**, i.e. is it eMBB, mMTC, URLLC, or Customized Slice type.

In addition, for the case of Network Slice Composition the following deep considerations are taken. Since the introduction of 5G, network slicing has been a topic of interest. While partially implemented in 4G, many see 5G slicing as one of the main drivers for new revenue streams from verticals. When 5G network slicing was discussed early on, there was an understanding that separate and dedicated slices with a fully dedicated core would be made for every use-case and every customer, regardless of the complexity of the use case or the size of the customer.

Then the community started to realize that due to physical and financial constraints, dedicated slicing for every customer or use-case is not likely to be commonplace. The number of entire slices may become directly proportional to number of new verticals requesting services, which may not be scalable. Therefore, sharing some common functions, depending on the situation, appears as the best option. Each slice will cause some overhead, and as such increase the **resource consumption**. With a large number of slices, the increase in consumption could be substantial and require further **hardware investments**. Depending on contractual agreements with the vendors, each slice could also bring **additional licensing cost**. Practically, the **complexity of operation will also increase, and necessitate support from the OSS and BSS systems**.

In 5G-VINNI it has been observed that there are vast differences in requirements from different customers and use-cases, and as such differences in their need for a dedicated slice. So far, 5G-VINNI has mainly worked with slicing on an NSA Core using dedicated core (DECOR). Being based on NSA Core, a dedicated slice would mean dedicated serving gateway/PDN gateway (SPGW), mobility management entity (MME), policy and charging rules function (PCRF) etc. Some of the use-cases we have seen so far no doubt require a dedicated slice, while others could share some or all network functions that constitute the slice.

One of the use-cases that has a very special set of requirements and conditions in this setup is the Norwegian armed forces were a dedicated slice was necessary. While working with them, it became apparent that their use case was complex with special requirements on security and high availability. With their extensive list of requirements and severity of use, it was necessary to dedicate a full slice in our NSA setup. For this customer, we believe that a fully dedicated slice would be necessary also in 5G SA slicing.

In contrast, for instance, there is the case of a fish farm were sharing network functions is possible. While this vertical initially indicated that they wanted a dedicated slice, further research into their needs showed that such measures were not necessary. The main requirement they had was heavy

throughput requirements due to their extensive use of video transmission from the fish farming pens. Thus, they could require a dedicated SPGW potentially distributed to the Edge, while the other network functions could be shared. In a 5G SA setting, this would mean a slice with only UPF and SMF dedicated.

Operators will receive many requests for dedicated slices in the years to come. When there are requests from multiple verticals customers, it is not cost efficient to have a full dedicated slice for every customer. In certain cases, the network slice can share network functions with other network slices, whereas in other cases this is not possible and network functions must be dedicated to that slice. This is an important part of the *Network Slice design* process. The main input to this *Network Slice Design* process is the requirements such as performance, security, privacy and high availability.

5G-VINNI initially proposed a flow chart or decision tree design as a first step in answering these questions. This flow chart should first and foremost be considered as a tool in the network slice design process between the operator and the customer.

3.1.1.1 Flow chart guide

The complete flow chart is shown in Figure 3-1.

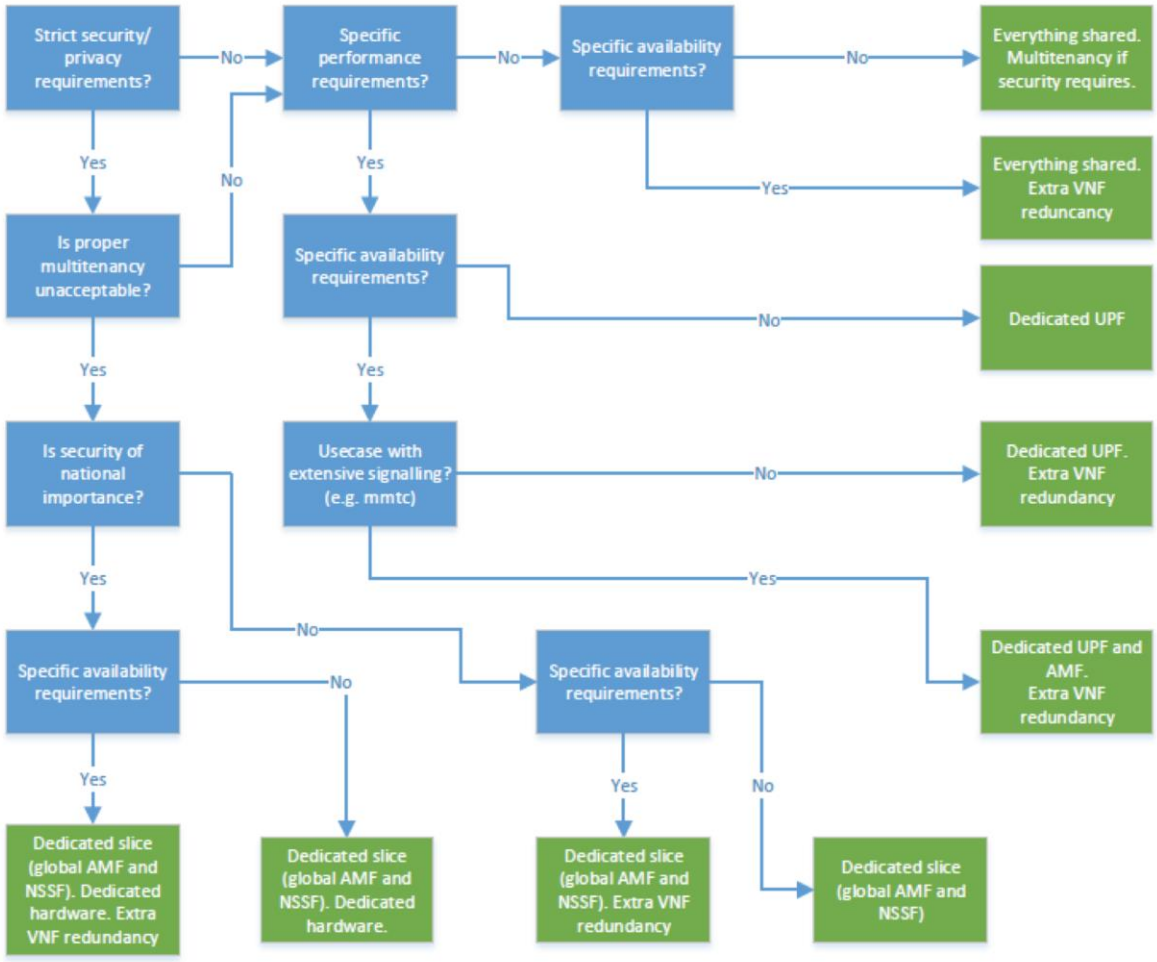


Figure 3-1: Flow Chart for Network Slice design

It is important to clarify that this approach:

1. works for 5G-VINNI Network Slice use cases as it is based on their specific design consideration;

2. is an initial proposal (first step) for illustration of how to decide which functions should or should not be shared in a specific Use Case slice.

The presented approach should be used in the following manner: The starting point is in the upper left corner. In this initial state, the assumption is that the customer will operate in a generic, shared slice. This is the baseline, possibly being an operator-wide use-case agnostic slice, similar to what operators offer on 4G, or a specific slice for eMMB (SST-1), URLLC (SST-2) or mMTC (SST-3) use-cases.

From the starting point, the reader should answer the questions presented with a “yes” or a “no”, each leading to a new question as indicated by arrows. The user of the approach must answer multiple questions, and each path of questions leads to a green box, constituting a complete core design. By answering “yes” to a question, a relevant feature is added to the core design, and the green box is as such a composite of the features added by answering "yes" to the questions. If a question is answered “no”, the feature relevant to that question is not added to the design.

The questions are based on the following requirements:

- Security and privacy – The confidentiality and data integrity requirements
- Performance – User-plane performance requirements
- Availability – The number of uptime 9s required
- Signalling – Requirements for signalling

These requirements are also summarized from a wider perspective in Table 3-1, which at the same time is a complement for Figure 3-1. A match between the question and the functionality added can be seen in Table 3-1, below. Keep in mind that the questions use the functionality of the shared generic slice as a baseline. This means that by answering “no” to the question about a requirement, e.g. security, it is implied that the generic slice is compliant with the present security requirements for the use-case. Note that the questions stated are high level and usually there are details that must be considered for each of these questions depending on the service to be delivered (e.g. what is the definition of strict security requirements?).

Table 3-1: Questions for Slice Design process using the Flow Chart method

Does the use case …		If yes …	Comment
Have strict security or privacy requirements?	→	Dedicated slice (AMF and NSSF is global)	A wholly dedicated slice for the use-case. However, the NSSF is global. Additionally, while the slice has a dedicated AMF, it is also necessary to initially be visible to the global AMF.
UNLESS proper multitenancy is acceptable, then -		Shared slice with multitenancy	If multitenancy is acceptable, and available, a dedicated slice is not necessary
Have security requirements of national importance?	→	Dedicated hardware for the slice	Due to the extreme security requirements, full isolation - with a dedicated slice on dedicated hardware. Potentially hardware shared with other use-cases of similar importance
Have specific performance requirement requirements?	→	UPF is dedicated	Dedicate UPF to ensure good UP capacity. Possibly extend flowchart with separate discussion on edge
Have specific availability requirements?	→	Extra VNF redundancy	Stricter requirements on uptime necessitates better redundancy
Rely on extensive signalling (e.g. mMTC)?	→	AMF is dedicated	Dedicate AMF to ensure good signalling capacity

3.1.2 5G RAN Slicing Design Considerations

5G RAN slicing is a promising new concept, currently under investigation by the research community. As such, 3GPP has not yet released specifications documenting possible implementations or defining interoperability requirements, therefore what is presented in early draft 3GPP specification versions and Releases might be different to what will eventually be standardized. This section serves as a brief literature review of the academic state of the art on this topic.

The basic premise behind RAN slicing is to offer dynamic resource management down to the Resource Block (RB) level to external slice owners. Each RAN slice has its own scheduler, which has a limited view of the gNB resources in order to achieve isolation and, to a limited extent, security. This isolation also guarantees the respect of the slices' SLA, regardless of the present coexisting slices. Each RAN slice may have its own interference management algorithm and handover mechanism, thus each RAN slice may act as an independent virtual base station. The allocation of RBs to the RAN slices may change with a granularity of 1 Transmission Time Interval (TTI). It is possible to provision for instances where meeting a strict SLA may require a slice to use resources of another slice. For example, a URLLC slice may use, for a brief period, RBs allocated to an eMBB slice. The loss of eMBB resources is covered by retransmissions.

There may be several RAN slice instances per gNB, managed by a RAN Real Time Control element. The configuration of RAN slicing is controlled by a RAN Near-Real Time Control element, which may reside at a different location. Each slice instance may be associated with one or more network slices. Since the UE may be served by up to eight network slices it is possible to be served by several RAN slice instances simultaneously. For example, an operator may offer an eMBB and a URLLC slice. Each of these slices would be served by a different RAN slice instance. An automotive UE, may be served by both of these RAN slice instances, where the eMBB slice may serve the on-board entertainment systems and the URLLC slice may offer safety messages services.

The most notable implementation of RAN slicing that has gathered the interest of the research community is Orion [5]. Orion is based on FlexRAN project [6] of the MOSAIC-5G ecosystem [7]. It allows independent and fully customizable control planes for each slice. It supports full isolation of radio resources and control functions. It is now integrated into the open source project Open Air Interface (OAI) [8]. This project currently supports LTE only. The most likely candidate of RAN slicing implementation within a 5G-VINNI research item is this OAI-based implementation, since it is an open project.

Services and UEs in NR SA deployments are always associated with E2E network slices in the 5GS. Network slicing is not an optional feature that could be switched on or off on demand and thus, even running only a single service like MBB over NR SA requires the provisioning of a corresponding E2E network slice. Figure 3-2 shows the provisioning concept for network slices and the aforementioned slice identifiers in a mobile network. The provisioning is done by means of:

- Device or node configuration – for example, the Configured NSSAI and Default Configured NSSAI are provisioned in the Network Slice Selection Function (NSSF) and Unified Data Management (UDM) nodes respectively. As mentioned before, the Default Configured NSSAI can also be pre-provisioned on the UE (SIM card).
- Non-Access Stratum (NAS) signalling – for example, the UE receives the Configured and Allowed NSSAIs (and Rejected NSSAI) via the AMF of the serving PLMN and the UE sends the Requested NSSAI when it registers with the mobile network.

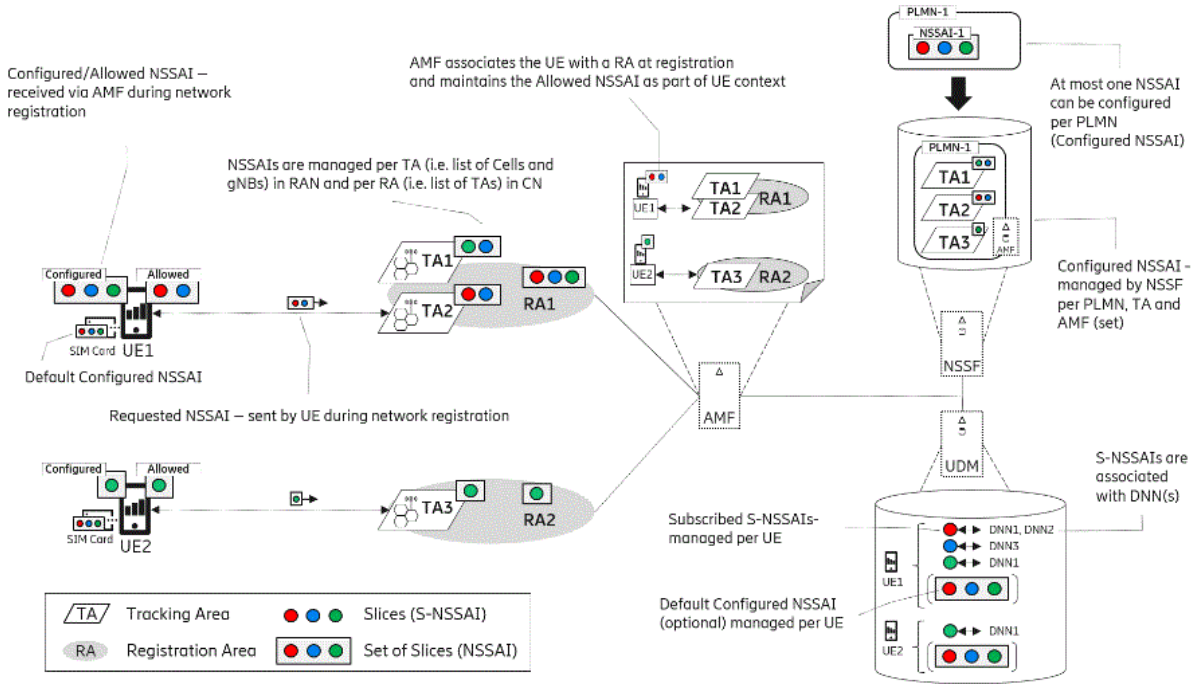


Figure 3-2: NSSAI and S-NSSAI provisioning overview

When a UE registers to the 5GS, it provides either the 5G Temporary Mobile Subscriber Identity (5G-S-TMSI), Registered AMF or Requested NSSAI (s-NSSAI-List) information to the gNB in the RRCSetupComplete message. The gNB selects the AMF using this information in following order: 5G-S-TMSI if present, otherwise Registered AMF if present, otherwise s-NSSAI-List.

If none of this information is provided by the UE, the gNB either selects a default AMF, if one was configured on the node, or otherwise, performs a weighted selection amongst all valid AMFs supporting the PLMN.

The AMF provides the UE with the Allowed, Configured and Rejected NSSAI information as well as the NSSAI inclusion mode in the NAS Registration Accept message.

For the UE registration process, it is important to note the following:

- For the initial registration, i.e. when the UE connects to the 5GS for the first time, the UE does not provide any of the 5G-S-TMSI, Registered AMF, or Requested NSSAI (s-NSSAI-List) information to the gNB over the Access Stratum (AS) layer in RRC messages. The handling of the Requested NSSAI in RRC messages is further explained below
- The CN controls whether and how the UE should provide the Requested NSSAI (s-NSSAI-List) to the gNB in RRC messages using the NSSAI inclusion mode (see 3GPP TS 23.501 [9]). The UE receives the NSSAI inclusion mode from the AMF in the NAS Registration Accept message. For the initial registration, 3GPP currently mandates the UE to operate per default according to NSSAI inclusion mode “D”. In this mode the UE does not provide the Requested NSSAI even though NSSAI information (Default Configured NSSAI) might have been pre-provisioned on the UE. The Requested NSSAI is thus only provided over the NAS layer to the AMF. Starting with 3GPP Rel-16, UEs can be pre-configured to operate according to NSSAI inclusion mode “C” in the HPLMN; the UE then includes the Requested NSSAI in RRC messages during the initial registration and mobility registration updates.
- For subsequent registration update procedures, e.g. mobility registration updates in case the UE moves to a Tracking Area (TA) outside the current Registration Area (RA) or periodic registration updates in case the UE needs to stay registered during inactivity, the UE can include the 5G-S-TMSI, Registered AMF, and Requested S-NSSAI (dependent on the NSSAI inclusion mode) information in RRC messages. Note: Dependent on the UE implementation

the Allowed NSSAI information that was once received during registration might be permanently stored on the device and be used to compose the Requested NSSAI in any following registration procedures including the initial registration after a UE power off/on cycle.

If 5G-S-TMSI, Registered AMF, and Requested NSSAI information is not presented by the UE during the initial registration, the gNB might select an AMF that cannot support the network slices that a UE requests over the NAS layer. In that case the AMF can re-route the registration request either indirectly via the RAN or directly to the target AMF dependent on local policy and subscription information (see 3GPP 23.502 [10] for details).

If such AMF re-routing methods are not supported or possible (due to AMF isolation requirements), the gNB must be configured with a default AMF that supports all the network slices deployed in the TAs served by the gNB.

Current 3GPP Rel-16 specifications do not natively support slicing in Next-Generation Radio Access Network NG-RAN architecture. This prevents having gNBs with slicing mechanisms in-built, and thus 5G-VINNI facility from providing differentiated behaviours on the RAN side. To provide 5G-VINNI with the ability to offer true 5G RAN slicing, two extended capabilities on individual gNBs need to be made available. First, the ability of a gNB to keep different Radio Resource Management (RRM) policies, each featuring a service-tailored resource scheduling. Secondly, the ability of a gNB to assign these policies to corresponding S-NSSAIs, each identifying a particular CN network slice.

In the absence of a Standardised method to provide these capabilities, RAN Slicing is considered as an optional extension for support within 5G-VINNI Facility Sites. Any implementation of RAN Slicing would be based either on extensions that cannot guarantee interoperability with core network Slicing, or with any UE capabilities that may be required to support RAN Slicing in the way it has been implemented. This does not preclude 5G-VINNI Facility Sites from including RAN Slicing, but if RAN Slicing is implemented, it is recognised that it is done so as a Research task that extends the network's capabilities beyond those defined in 3GPP R16.

3.1.3 5G TN Slicing Design Considerations

A transport slice is a logical network topology connecting a number of endpoints and a set of shared or dedicated network resources, which are used to satisfy specific Service Level Objectives (SLO) [11]. These objectives define a set of network resource parameters or values necessary to provide a service as requested for a given transport slice. A transport slice can have one or more SLOs associated with it; all SLOs combined to form an SLA. Some of the SLO categories are as follows: Guaranteed Minimum Bandwidth, Guaranteed Maximum Latency, Maximum permissible delay variation, Maximum permissible packet loss rate, Availability.

The isolation on transport networks between RANs and CNs can be hard or soft isolation, depending on slice requirements. The isolation technology can be FlexE/MTN interface isolation, MTN cross-connection isolation, or VPN+QoS isolation depending on isolation, latency and availability requirements [12].

- Hard Isolation Technology
    - FlexE/MTN Interface Isolation: With FlexE/MTN interface, multiple elastic Ethernet hard pipes can be created on one physical port. Services can achieve time slot isolation at the interfaces and achieve statistical multiplexing within the devices.
    - MTN Cross-Connection Isolation: Based on the Ethernet 64/66B code block-based cross-connection technology, TDM (Time Division Multiplexing) time slot isolation is achieved, thereby achieving extremely low forwarding delay and isolation effects. The forwarding delay of a single hop ranges from 5μs to 10μs, which is much lower than that of traditional packet switching devices.

The FlexE/MTN interface isolation technology can be used together with the MTN cross-connection isolation technology or packet forwarding technology for packet transmission

- Soft Isolation Technology
  - VPN+QoS isolation: Services on a physical network can be isolated using VPNs. However, the VPN+QoS software isolation cannot achieve timeslot-level isolation as in physical isolation.

The QoS technology includes different tools such as packet classification, policing and shaping, congestion avoidance and congestion management. The packet classification distinguishes one type of traffic from another based upon different packet fields such as 802.1P priority of the L2 VLAN tag or IP DSCP. Policing / Shaping are mechanisms to enforce the throughput metric of an SLA. The congestion avoidance techniques monitor network traffic load in an effort to anticipate and avoid network congestion. This is achieved through packet dropping and by employing mechanisms such as Random Early Detection (RED) and Weighted Random Early Detection (WRED). Congestion management features control congestion after it occurs. Queuing algorithms are used to sort the traffic and then determine some method of prioritizing it onto an output link. Congestion-management algorithms include Weighted Fair Queuing (WFQ), Strict Priority (SP), Weighted Round-Robin (WRR) as well as hybrid methods as a combination of the aforementioned.

However, traditional single-level QoS is not granular enough to fulfil the 5G slicing requirements for the transport network. In 5G, innovative business models [13] have been identified, that can be established when network slicing is used for the provisioning of communication services. These models assume the existence of innovative provider-client relationships between actors (network operators, communication service providers and communication service customers) involved in the service value chain, with these actors taking different roles, according to the position in the service value chain [14] . To this end, to accommodate the different roles that have been defined for the service ecosystem in traffic differentiation policies, hierarchical QoS (H-QOS) must be employed. Hierarchical QoS enables transport network slicing [15] by applying a hierarchy of schedulers and shapers as shown in Figure 3-3 [16].

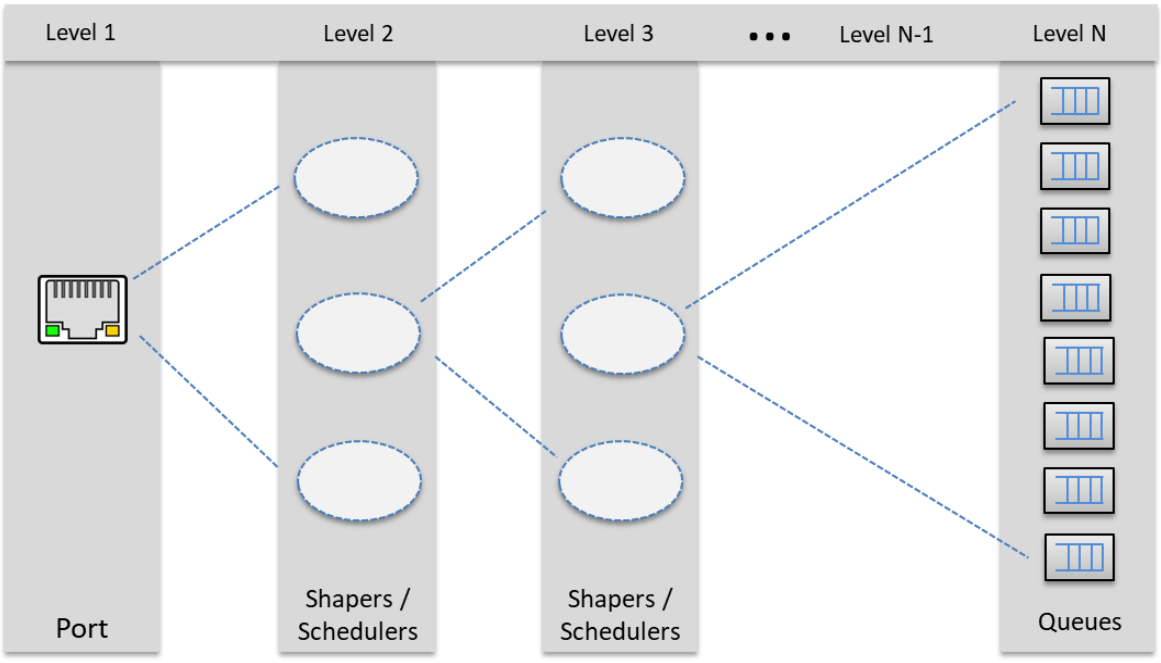


Figure 3-3: H-QoS architecture for transport network slicing

Each level of the scheduler hierarchy can be used to shape traffic based on different criteria. The aforementioned criteria can be different Ethernet packet fields that correspond to different roles of the 5G service ecosystem. Packet field marking is done at the egress ports of the related radio and core nodes. E2E QoS design uses a blend of the 3GPP and IETF QoS architectures and mapping from DiffServ to Multi-protocol label switching (MPLS) or p-bits, or the other way around, is a function of IETF QoS principles and not 3GGP (Figure 3-4) [17].

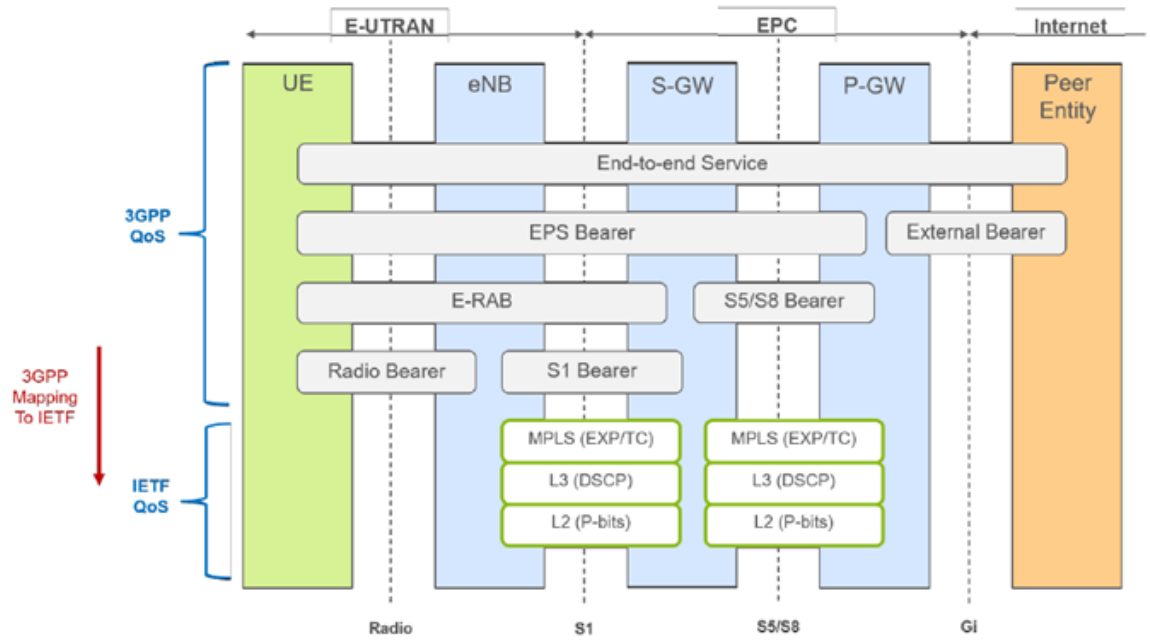


Figure 3-4: E2E QoS design uses a blend of the 3GPP and IETF QoS architectures [17]

3.2 5G System Design: Architectural Considerations of MEC in 5G-VINNI Facility

As stated in D1.1 section 6.1 [1] and D1.4 section 3.7 [2], the ETSI-Multi-access Edge Computing (MEC) framework (see ETSI GS MEC 003 [18]) allows to implement MEC applications as software-only entities, namely, Software as a Service (SaaS) applications that run on top of a virtualization infrastructure located close to the network/core edge. In the 5G-VINNI Project each Facility Site has developed a 5G NFV-based environment with several virtualization technologies that have led to ecosystems where both technologies co-exist. This kind of scenario is contemplated by the ETSI MEC Industry Specification Group (ISG) in ETSI GR MEC 017 [19] and in order to reach it, some conditions SHOULD be fulfilled (see D1.4 section 4.10.3 [2]); however, other considerations need to be taken into account as per deploying a MEC solution at any 5G-VINNI facility Site. Firstly, it is important to remark that a paramount number of virtualization technologies are being used over the 5G-VINNI Facility Sites and, therefore, the final architecture of the deployed solution MUST take into account that wide range of technologies. The ETSI GR MEC 027 [20] analyses the impact and support of the ETSI MEC framework for alternative virtualization technologies such as: OS containers, Higher-level containers, nesting and mixing of virtualization techniques, Alternative Virtualization Techniques (AVTs), etc. Nesting these technologies implies that the virtualization layer within each 5G-VINNI Facility Site Network Function Virtualization Infrastructure (NFVI) MAY be comprised of multiple virtualization technologies nested into sub-layers. Moreover, a major aspect of the MEC environments is the potential need to set-up MEC Applications that belong to different 3rd parties., One solution to this kind of scenario is to use the nesting of virtualization technology techniques in order to allocate a VNF to each different 3rd party application owner, allowing them to allocate the resources assigned to its VNF to the multiple applications it runs and, furthermore, to do so based on

its own internal criteria (see Figure 3-5 – left). Additionally, this kind of environment tends to mix multiple virtualization techniques, as shown in Figure 3-5 - right, with the benefit of the particular characteristics of the virtualization technology of choice.

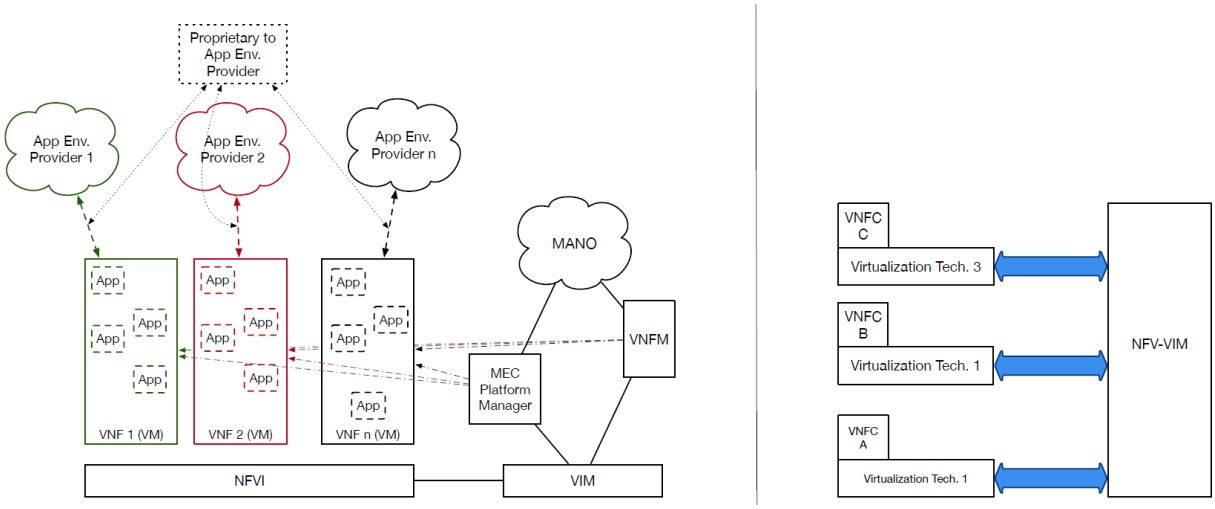


Figure 3-5: Example of nesting (left figure) and mixing (right figure) Virtualization Technologies key use cases

Taking into account the aforementioned considerations and potential use cases, 5G-VINNI facility sites could plan deployment of different MEC solutions within it virtualization infrastructure.

Three different edge deployment scenarios (Figure 3-6, Figure 3-7, and Figure 3-8) in 5G environments have been contemplated by 5G-VINNI, in all of them the UPF has a dedicated N6 interface associated with each edge node hosting multiple applications. In some cases, the UPF MAY have multiple logical N6 interfaces (one for each application) associated with the edge node.

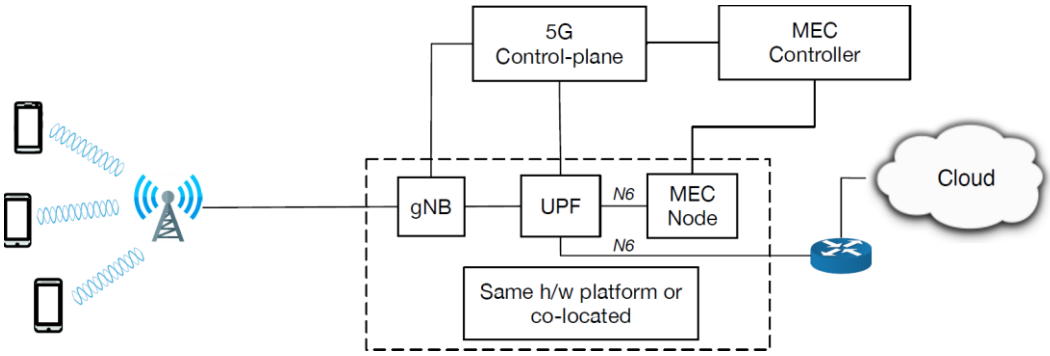


Figure 3-6: Scenario 1 - MEC node hosts MEC Apps and is co-located with Base Station and UPF

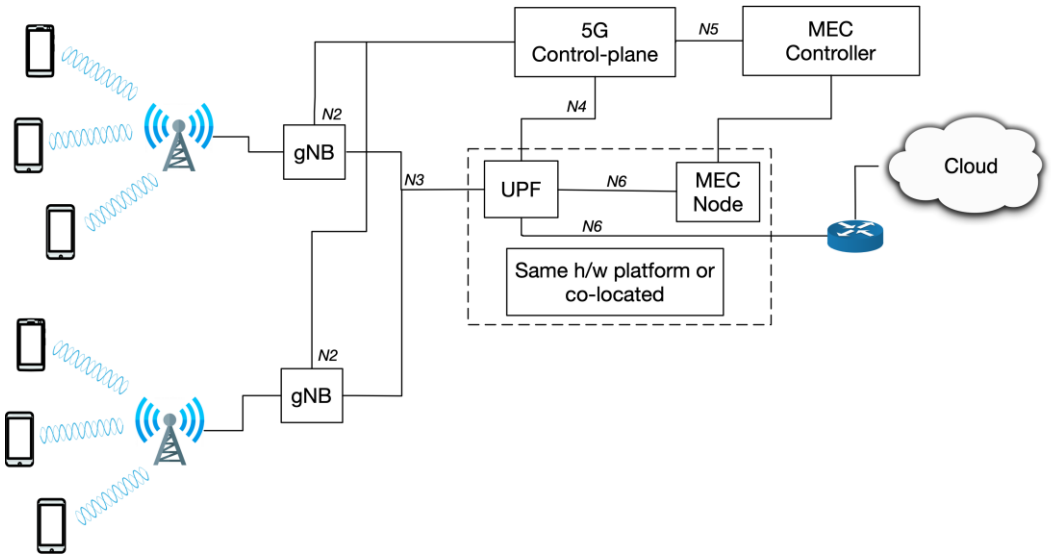


Figure 3-7: Scenario 2 - MEC node hosts MEC Apps and is co-located with local UPF

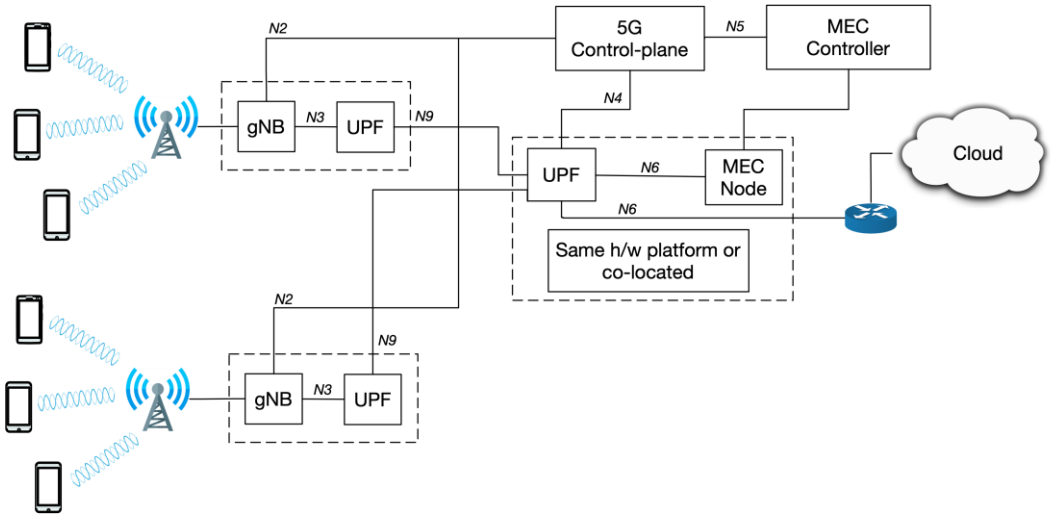


Figure 3-8: Scenario 3 - MEC node hosts MEC Apps and is co-located at Regional-Office

3.3 5G Control Mechanisms: Network Slice Federation

One of the advantages for a vertical, of using 5G-VINNI, is the possibility to deploy a service (Network Slice) across different facilities and domains. Such a service deployment requires data plane connectivity between the involved facilities, and interoperability among their orchestration systems. For this interworking, two approaches can be followed:

- hierarchical orchestration;
- peer-to-peer orchestration.

The first federation approach assumes the definition of a parent orchestrator, sitting on top of multiple child orchestrators, coordinating their workflows and providing translation of their information/data models. This introduces significant burdens in management scalability, as the number of facilities connected to this master orchestrator increases. Additionally, the scenario of having a network operator taking the broker role is unrealistic for upcoming operational networks, as it would raise concerns with the rest of operators in terms of privacy, auditability and non-repudiation. For this reason, the peering approach is preferred for federating domains.

In the following subsections, the use cases supported by 5G-VINNI are summarised and the functional architecture designed for these use cases is described in detail.

3.3.1 Network Slice Federation Use Cases in 5G-VINNI E2E Facility

In this section, an example of how federation enables the deployment and operation of an E2E slice instance across two facility sites upon vertical request, is provided. In this process, three phases can be envisioned: *slice ordering*, *slice fulfilment* and *slice operation*. This example may apply to any facility from 5G-VINNI, though, for this example we will consider two (2) 5G-VINNI Facility Site (X & Y), as shown in Figure 3-9. There will be a reference to TMForum Open APIs used, though, those will be detailed in the forthcoming 5G-VINNI deliverables.

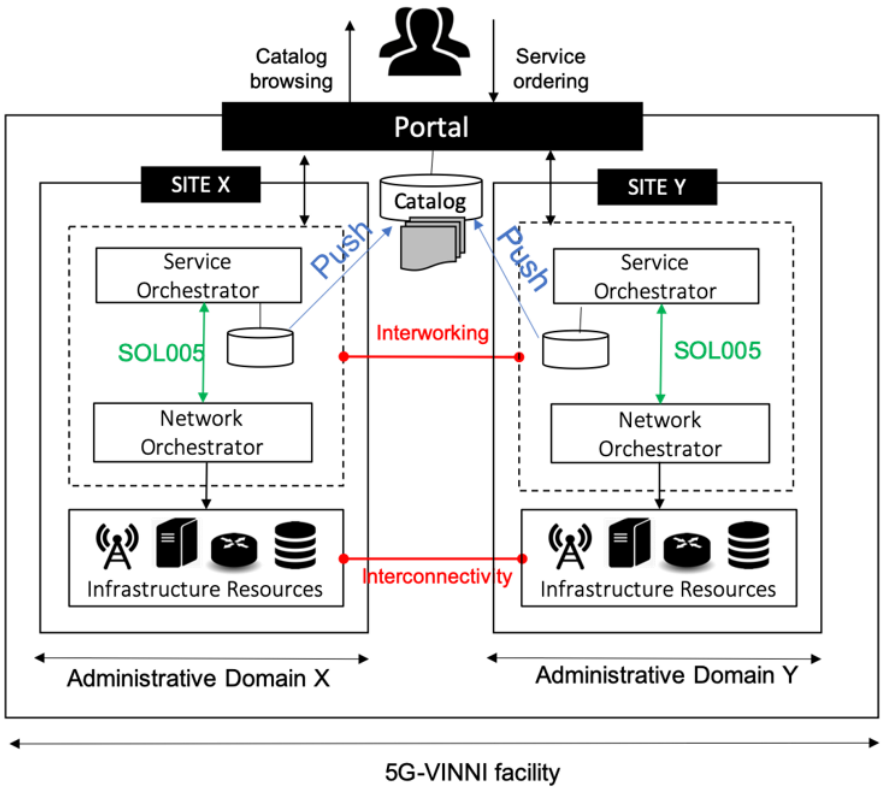


Figure 3-9: 5G-VINNI facility architecture

3.3.1.1 Slice ordering

In the first phase, the vertical gains access to the 5G-VINNI facility through the portal, browses the centralized service catalogue, selects one VINNI-SB and issues the corresponding service order (for details of VINNI-SB see D3.1 [14]). In this service order, the vertical provides a completed specification of the slice instance needed, including information on slice topology (possible extended with 3rd party VNFs), slice attributes (filled in with values fitting use case requirements) and slice location. We assume the following: *i)* the vertical wants the slice deployed across two Facility Sites, each having a different orchestration solution; and *ii)* the selected VINNI-SB was retrieved from the local catalogue of one of these Facility Sites. By way of example, consider that the vertical orders the provisioning of an eMBB slice instance across UK and Spain, by selecting a VINNI-SB with SST=1 from the 5G-VINNI service catalogue, retrieved from Spain's OSM catalogue.

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

3.3.1.2 Slice fulfilment

In the second phase, upon receiving the service order, Facility X checks it, realizing that part of the ordered slice needs to be deployed at Facility Y site. This means that federation between the Service Orchestrators (SO) of both sites (for this example we consider the ones in use in 5G-VINNI: OSM and Nokia's FlowOne) is needed. From this point, the event workflow is as follows. First, OSM on-boards the VINNI-SB into FlowOne's catalogue, by retrieving it through the **Service Catalog API**. Then, OSM decomposes the service order received from the portal, identifying the subnets that will be deployed on Facilty X and Facility Y. Finally, it triggers a service order towards FlowOne, and respective service instantiation through **Service Ordering API**. With this order, OSM informs FlowOne about the topology and attributes of the slice (subnet) instance to be deployed on Facility Y.

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

3.3.1.3 Slice operation

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

3.3.2 Inter-Slice Interworking Mechanisms and Functional considerations implemented in 5G-VINNI E2E Facility

5G-VINNI leverages the latest 5G technologies to assemble a test and validation facility that provides industry verticals with isolated service experimentation platforms, in the form of E2E slices. These slices, accessed and used by verticals to set up innovative use case trials, are provisioned under the Network Slice as a Service (NSaaS) model.

As shown in Figure 3-9, this facility consists of several interworking sites, each offering one or more slice types. The service offering from the different facility sites are registered into a single service catalogue, which is made available to verticals through a common portal.

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

Cross-domain slice deployments brings several challenges in 5G-VINNI facility, as they require data plane connectivity between the involved facility sites, as well as interworking between their orchestration systems. In the following, we will discuss different federation options (Section 3.3.2.1).

3.3.2.1 Federation options

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

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

All the above options are technically feasible when the federated sites rely on the same orchestration solution. In such a case, the use of proprietary interfaces is enough to enforce the required communication and capability exposure across domains. However, this scenario is rather unrealistic, as is unlikely to be found in commercial networks, where federation may involve multiple sites from different network operators, each making usage of a different orchestration solution.

In 5G-VINNI, though, there are multiple facilities each making usage of a different orchestration solution, and interoperability can only be achieved by means of standard interfaces. Table 3-2 gives an insight into the three federation options, specifying their main features and the standard interfaces that can be used to fulfil these features. As seen, there exists at least one interface to implement every federation option. For example, SOL011 and SOL005, which define RESTful APIs for the implementation of Or-Or and Os-ma-nfvo interfaces, have become the standard solutions for the second and third federation options.

According to the above reasoning, from a technical viewpoint, the three options are equally valid and feasible for the intended federation. However, from an industry viewpoint, some options are less appealing than others, as happens with NFVO-NFVO and SO-NFVO. The main problem with these options is the difficulty to bring them to the market, because of the reluctance of an operator to expose his NFVO beyond the boundaries of his administrative domains. Many reasons explain this reluctance. First, the need for the operator to expose on-boarded NSDs-VNFDs to other operators, especially considering that descriptor design is recognized as one of the main key enablers for revenue increase (and service differentiation) between operators. Secondly, the need for the operator to allow connection between his NFVO to an external NFVO-SO. In the case of SO-NFVO approach, this is even worse, since having two (or more) SOs connected to the same NFVO increases the risk of generating conflicting policies and inconsistencies in the status of that NFVO. Thirdly, the lack of in-built auditability in SOL011 and SOL005, which makes the corresponding NFVO exposed interfaces sensible points in terms of security and autonomy.

Considering the abovementioned cons, **SO-SO is considered the most realistic solution for future commercial networks, and thus it is the one explored in 5G-VINNI project**.

Table 3-2: Federation Options

Option	Main Features	Standard interfaces
SO-SO	Information exchanged with external SO: list of on-boarded VINNI-SBs, selected configuration of deployed slice (subnet) instances. Operations exposed for external SO invocation: slice (subnet) provisioning; slice (subnet) performance assurance; slice (subnet) fault supervision; network functions application layer conf & mgmt.	MEF LSO Interlude [21]
NFVO-NFVO	Information exchanged with external NFVO: list of on-boarded NSDs-VNFDs; records of deployed network service/VNF instances, with information on their resources. Operations exposed for external SO invocation: network service/VNF lifecycle mgmt; network service/VNF monitoring; network service/VNF resources mgmt.	Or-Or [5]
SO-NFVO	Information exchanged with external SO: the same as for NFVO-NFVO,	Os-Ma-nfvo

	but without information on instances resources. Operations exposed for external SO invocation: the same as for NFVO-NFVO, but without resources mgmt. Information exchanged with external NFVO: slice (subnet) – network service mapping.	[12]

3.3.3 Network Slice Federation Actors, Roles Model

A brief discussion of existing 5G/network slicing actor role models (5GPPP, TM Forum, 3GPP) was included in [3], along with an analysis on how the 3GPP model can be mapped to different 5G-VINNI scenarios. In particular, the 5G-VINNI model, discussed in [3], adopts the roles of: (i) Communication Service Customer (CSC), to be played by 5G-VINNI customers such as ICT-19 vertical industries (ii) Communication Service Provider (CSP), to be played by 5G-VINNI Facility Sites and (iii) Network Operator (NOP), to be played by 5G-VINNI Facility sites. Furthermore, two types of CSC were defined, the *basic CSC* that do **not** have management capabilities over the network slice/service consumed, and the *advanced CSC* that has management capabilities over the service consumed through the Communication Service Management Function (CSMF). Finally, the service provided by 5G-VINNI Facility Sites is categorized into single site and multi-site services. The latter case is depicted in Figure 3-10 and it is the scenario that is relevant when it comes to *network slice federation*. Note that this model focuses mostly on the functional roles related to network slicing rather than on business roles that were extensively discussed in [22].

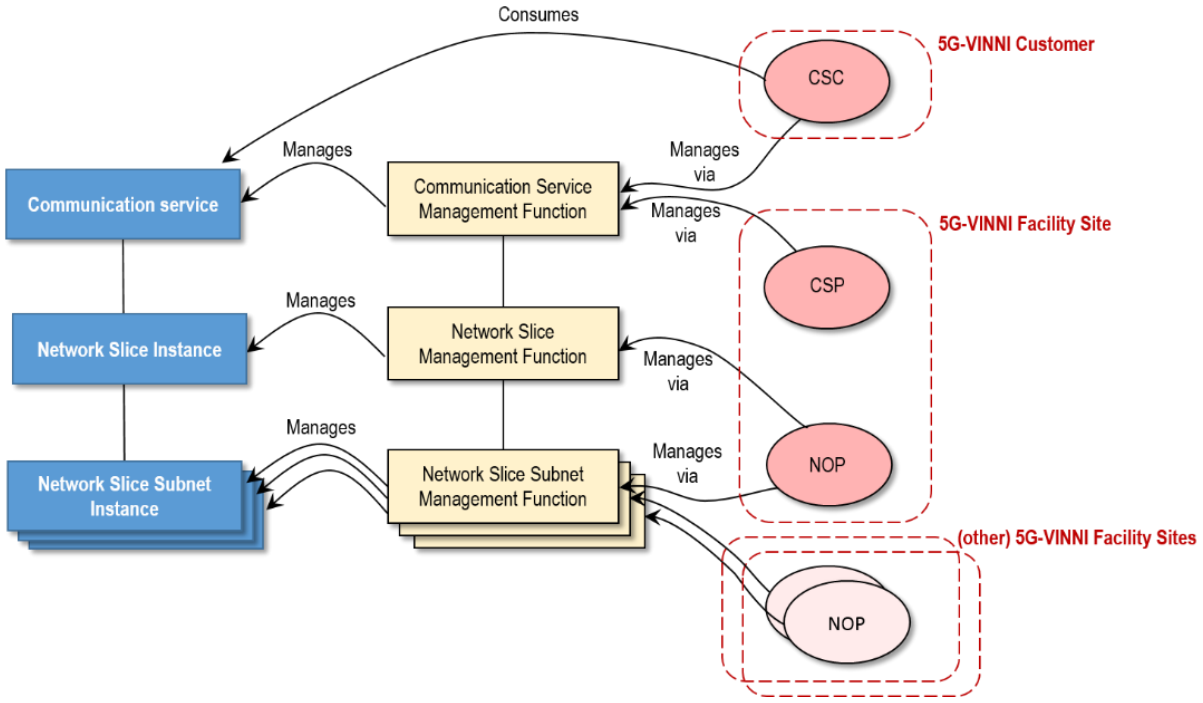


Figure 3-10: Advanced 5G-VINNI CSC, Multi-Site [3]

The actor role model of Figure 3-10 is enhanced here, and the different network slice federation options presented in section 3.3.2.1 are analysed. In particular, apart from already discussed actor roles that focus on the orchestration/functional part two additional "aggregator" roles (initially introduced in [22]) are identified that will help present the different federation scenarios more clearly.

The following roles are introduced:

- **Service Aggregator:** The entity that plays this role is responsible for operating a common 5G-VINNI portal where VINNI-SBs from multiple Facility Sites are made available through a global

catalogue. Note that this role can be adopted by a CSP, i.e. a Facility Site operator in the case of 5G-VINNI, or a 3rd-party entity that may serve as a global portal operator without maintaining infrastructure.

- **Network Aggregator:** The entity that plays this role is responsible for aggregating Network sub-slices instances from multiple NOP/Facility Sites and operating an end-to-end slice. This role can be adopted by one or more Facility Sites.

The three different scenarios presented in Section 3.3.2.1, incorporating the two roles above are presented in Figure 3-11 - Figure 3-13.

SO-SO

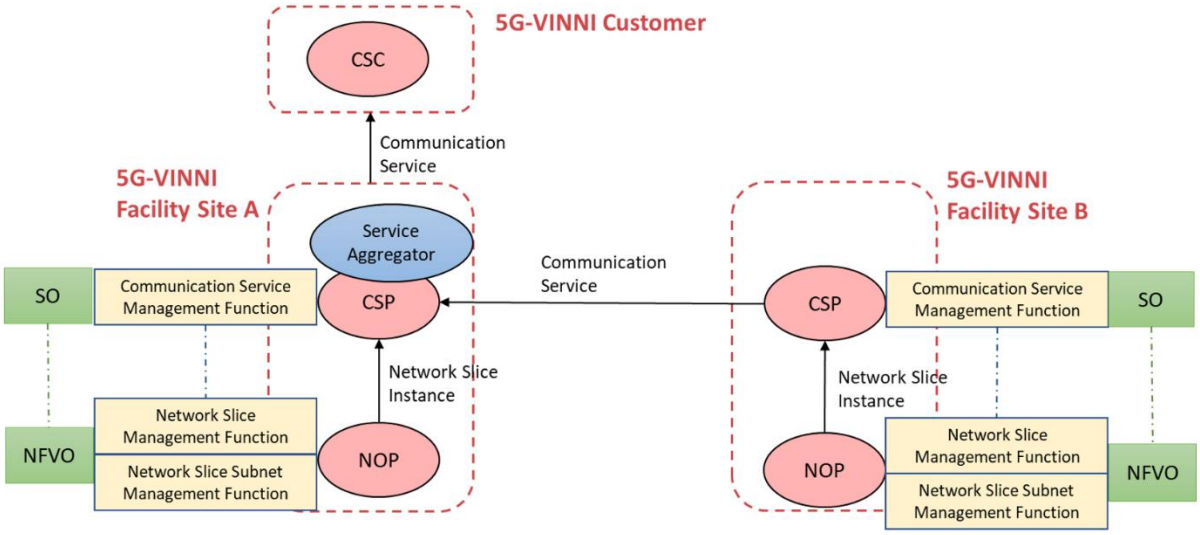


Figure 3-11: Basic 5G-VINNI CSC, SO-SO network slice federation scenario

NFVO-NFVO

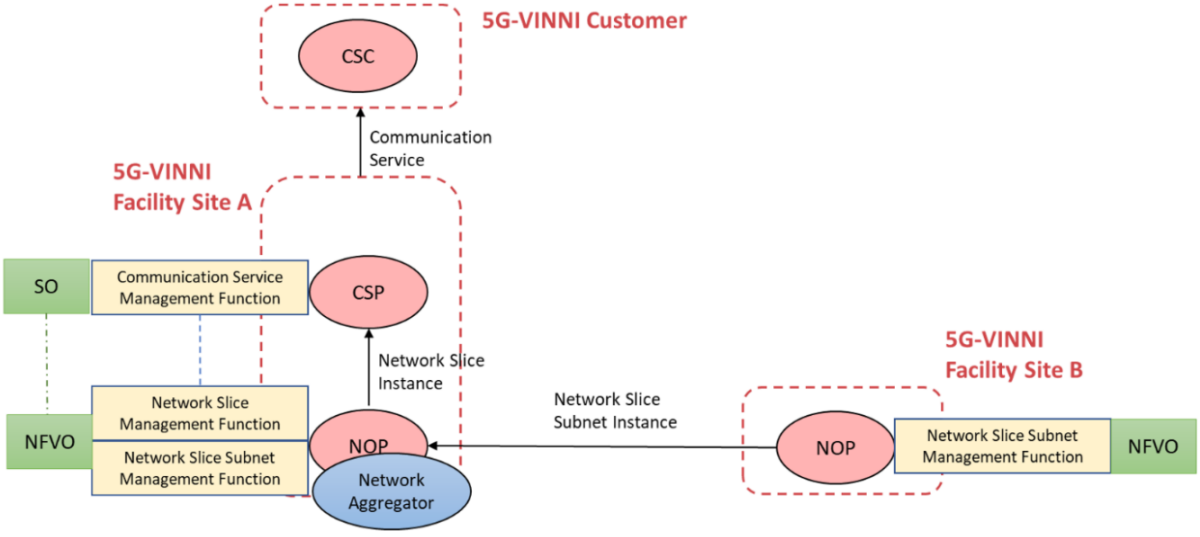


Figure 3-12: Basic 5G-VINNI CSC, NFVO-NFVO network slice federation scenario

SO-NFVO

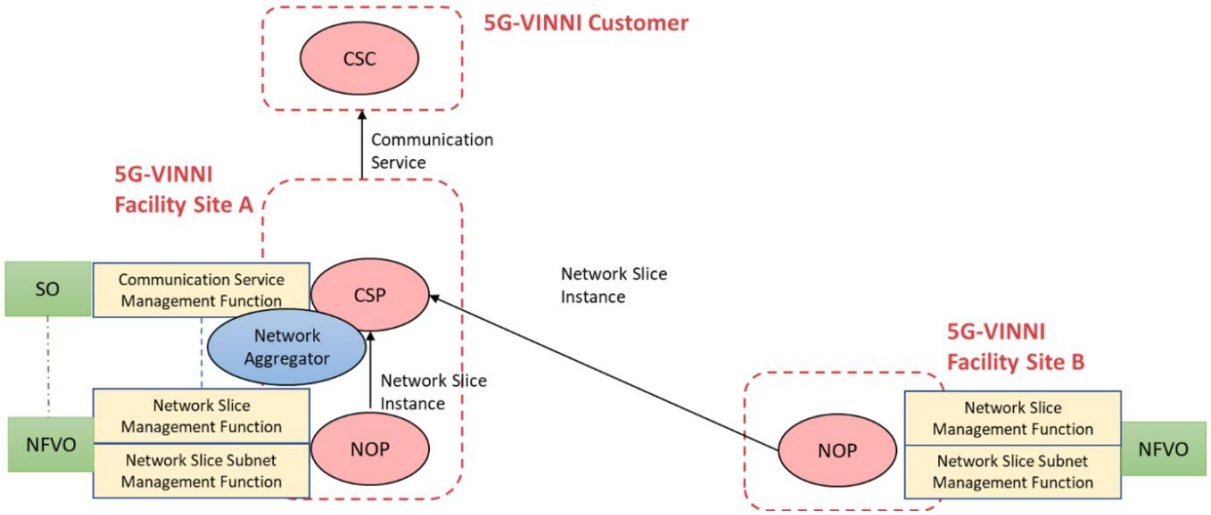


Figure 3-13: Basic 5G-VINNI CSC, SO-NFVO network slice federation scenario

3.4 5G System Enhancement: Support of Private Networks in 5G-VINNI Facility

Although the first generation of networks based on the 5G system architecture (3GPP Rel-15) were mainly conceived for public use, the possibility of having 5G networks also deployed for private use has recently raised a lot of interest in the industry community. As a result, the study of private networks has been included as part of the specifications related to the second phase of 5G networks (3GPP Rel-16 and beyond), resulting in the so-called Non-Public Networks (NPN). An NPN is a 5G system intended for the sole use of a private organization, typically an industry vertical. Unlike Public Networks (PLMNs), typically focused on providing *services* for *public* subscribers, NPNs are used for the provisioning of *services* for *private* subscribers, i.e. non-public services, including communication (i.e. telco) services as well as digital (i.e. vertical) services. From a deployment viewpoint, NPNs can be categorized into:

- Stand-alone NPN (SNPN): an NPN which does not rely on network functions provided by an MNO as part of a PLMN.
- Public Network Integrated NPN (PNI-NPN): an NPN deployed in conjunction with a PLMN. Unlike SNPN, PNI-NPN is an NPN where some network functions are made available via the PLMN.

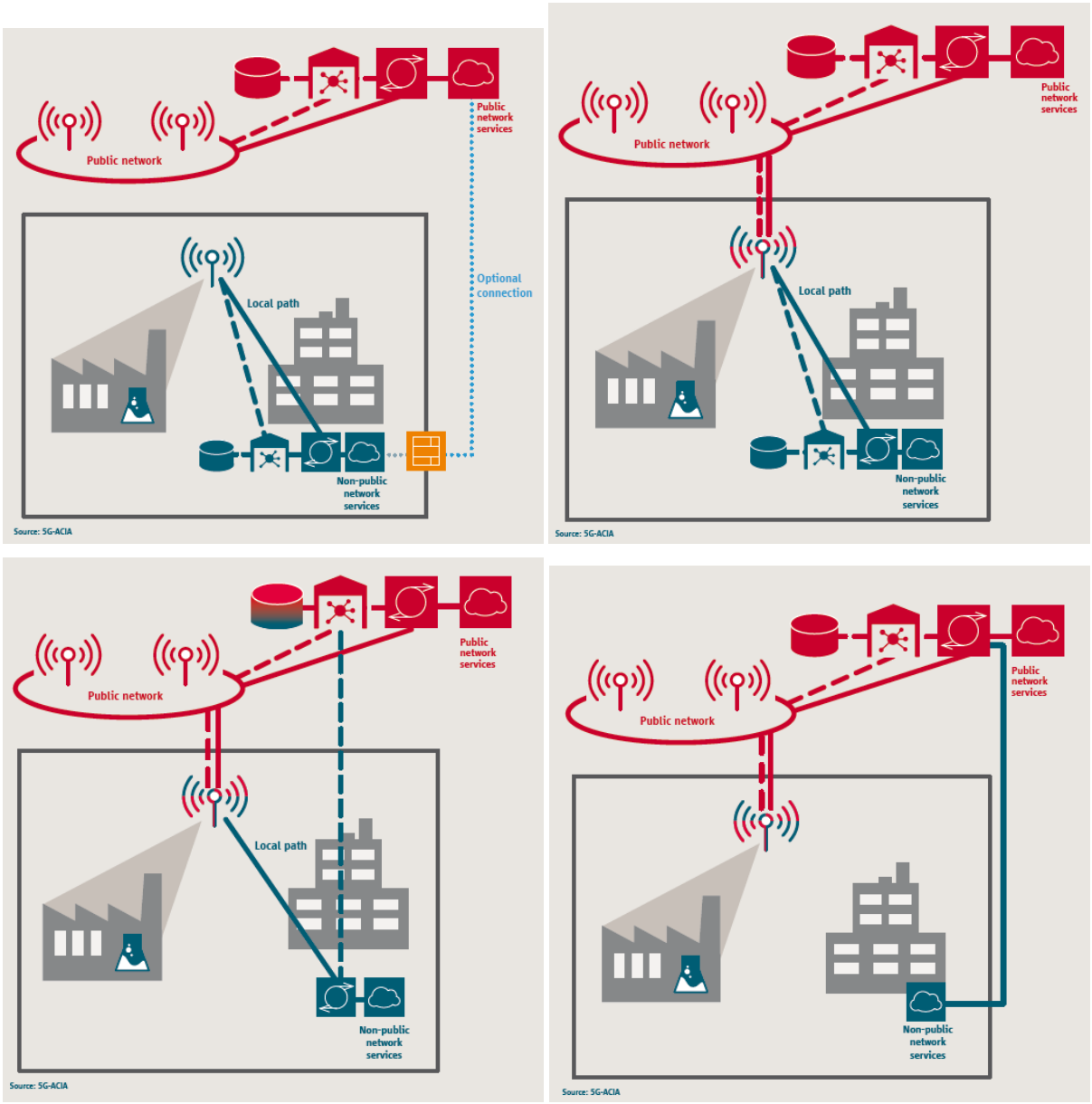


Figure 3-14: 5G-ACIA scenarios [23]

Based on the above characterization, 5G Alliance for Connected Industries and Automation (5G-ACIA) which is central global forum for shaping 5G in the industrial domain, has defined different deployment options for NPNs [23]. These scenarios are depicted in Figure 3-14. While the first one (isolated NPN) correspond to SNPN category, the second (shared RAN), the third (shared RAN and control plane) and the fourth (NPN within public network) scenarios fall into the PNI-NPN category.

The 5G-VINNI system can be a facilitator for PNI-NPN provisioning, with 5G-VINNI facility taking the role of PLMN, and vertical provider sites (e.g. private sites like industry 4.0 factories, campus, transportation hubs, etc.) taking the role of private network. There are different options to provide a PNI-NPN, e.g. access to the NPN can be made available using dedicated Data Network Names (DNNs), or a Network Slice can be dedicated to an NPN with various levels of shared resources and functions between the NPN and PLMN. In 5G-VINNI, we assume the second option.

3.4.1 Network Slicing as Enabler for Private Networks

Besides allowing the concurrent execution of multiple services on a common shared network infrastructure, 5G-VINNI can use network slicing for PNI-NPN provisioning, providing private sites

with dedicated slices using Network Slice as a Service (NSaaS). Figure 3-15 illustrates how 5G-VINNI facility operator can rely on NSaaS capabilities (e.g. OAM, exposure) for the provisioning of a PNI-NPN towards a 5G-VINNI customer, typically an industry vertical. This PNI-NPN, which is deployed across one PLMN and the vertical's premises (e.g. factory), can be seen as an end-to-end network composed of two differentiated segments: one private, consisting of network functions deployed in-house, using private 5G resources; and one public, consisting of network functions built upon public 5G network resources. Using the NSaaS approach:

- The public segment is made available by the PLMN in the form of a dedicated slice, and provisioned by the 5G-VINNI facility using NSaaS. In this service provisioning, the 5G-VINNI facility operator and the vertical play the roles of CSP-A and CSC-A, respectively.
- The vertical adds the private segment to the network slice obtained from the 5G-VINNI facility. The resulting combination (PNI-NPN) is a new network slice. Following 3GPP Network Resource Model (NRM) [24], the PNI-NPN's public segment can be modelled as a network slice subnet. In this case, the vertical plays the role of NOP-B.
- The vertical uses the network slice to provide (non-public) communication/digital services to its customer(s). In this regard, the vertical and its customer(s) play the role of CSP-B and CSC-B, respectively.

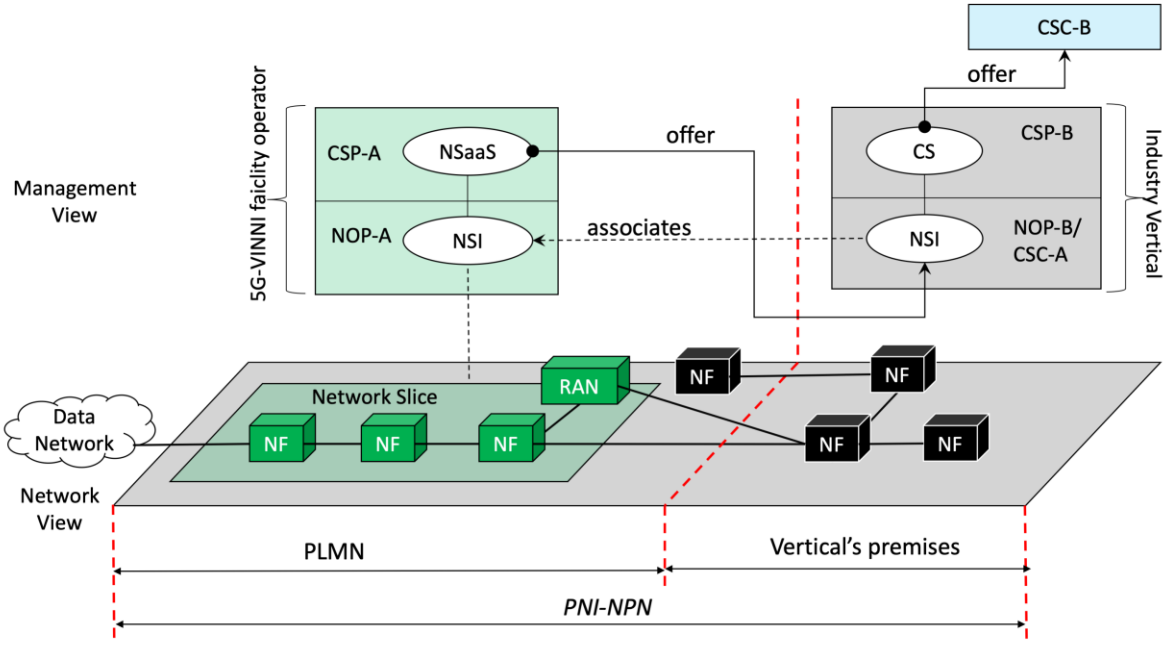


Figure 3-15: Network Slice as a Service in 5G-VINNI

Figure 3-16 shows an example of a PNI-NPN deployed using 5G-VINNI provided slice.

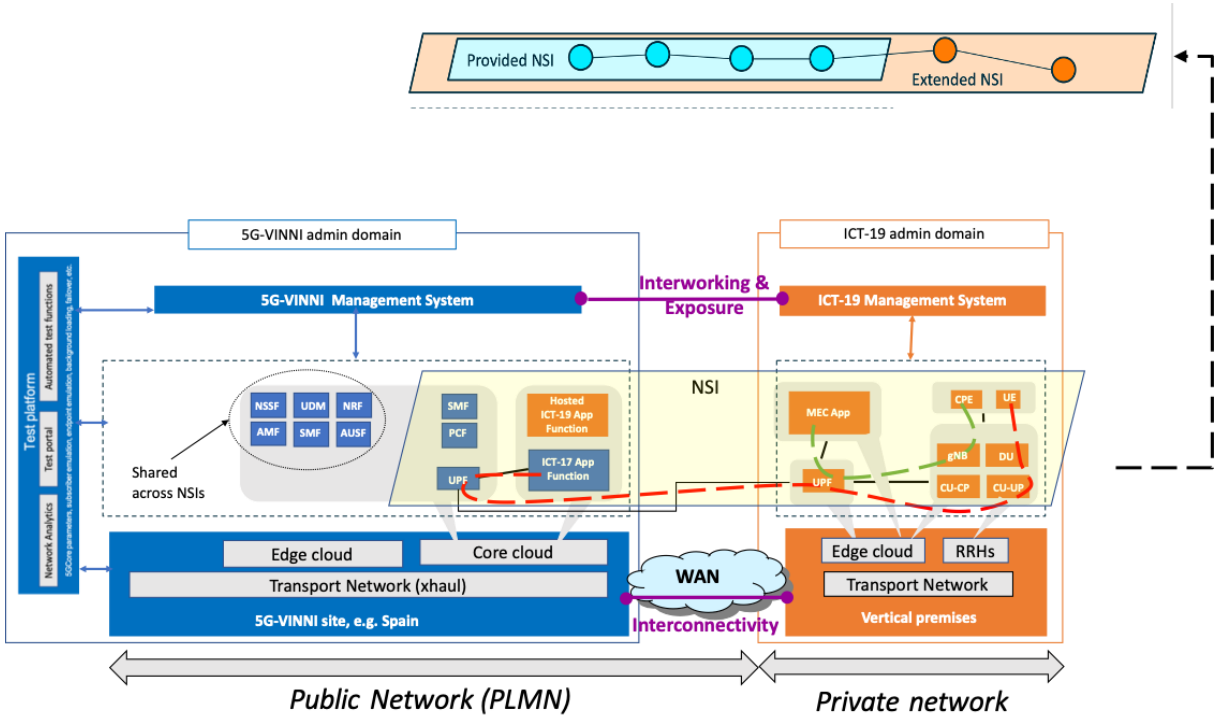


Figure 3-16: 5G-VINNI providing NSaaS for PNI-NPN provisioning.

3.4.2 Example Private Networks Use Cases and Requirements in 5G-VINNI Facility

There are a number of use cases in 5G-VINNI which are candidate for private network deployment. For example, some Network Slicing use cases offered by Norway Facility Site can be considered as NPN scenarios. The use case of autonomous edge with Norwegian defence has the full mobile network in the premise of the enterprise is the case of *isolated NPN* in Figure 3-14. This is because of the strict requirements of E2E encryption and isolation. In addition, the Norwegian defence want to have their own applications running in the isolated NPN such as mission critical push to talk services. However, there is also a requirement from the Norwegian defence that they are able to seamlessly integrate with the public network when desired and this can be achieved by the "optional connection" in *isolated NPN* case in Figure 3-14. It needs to be highlighted that there is also a requirement of synchronizing the user data records when the NPN users roams to the public network. In addition, 5G-VINNI also plans to implement the less stringent option of *shared RAN* for the Norwegian defence.

The *shared RAN* scenario is also applicable to the Industry 4.0 use case which 5G-VINNI plans to work with 5G-SOLUTIONS [25], which is an ICT-19 project. The main requirement for industry 4.0 use case is low latency both in the control plane and the data plane. The *shared RAN* option and *Shared RAN and control plane* option is relevant for the hospital use case in 5G-VINNI where the main focus is to keep the data local due to privacy reasons and in cases such as robotic control to have low latency. Then there is the use case of fish farming where the requirement is to process the data coming from the fish farms to make decisions locally using AI. As an example, monitoring the video feed from the camera in the fish farm to detect the lice on the fish and instantly kill the lice using lasers. These are some of the use cases being worked upon in the Norwegian facility of 5G-VINNI.

4 Network Slicing Security and Isolation Mechanisms

4.1 Security as a Service

In 5G-VINNI D1.1 [1] the Zone model used for potential security enhancement in the different 5G-VINNI Facility Site datacenters was described. Additional requirements from some verticals has resulted in the model being extended in order to provide what in 5G-VINNI is defined as Security-as-a-Service, which in turn allows the model to be applied to individual slices.

In order to start, first it is important to highlight the generic zone model as it is presented in Figure 4-1.

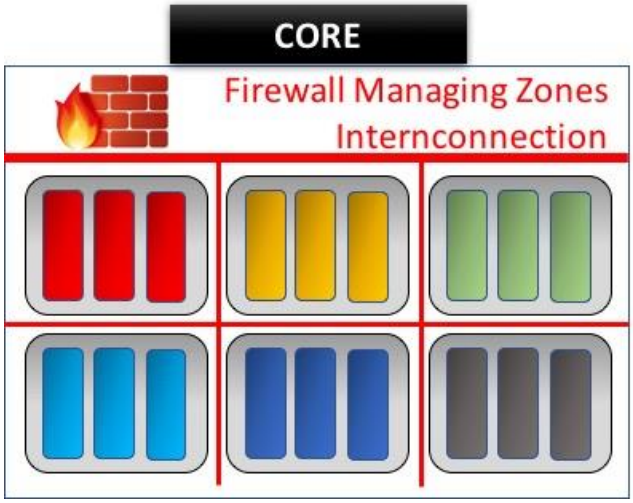


Figure 4-1: Global view of the Zone Model used in 5G-VINNI

Providing Security-as-a-Service implies that the Vertical (e.g. industry 4.0 factories) is able to manage the firewall of the respective desired domain, with full autonomy. In order to achieve this, it is important to highlight two techniques/principles that enable this goal:

- Technique 1: Zones can be split and personalized by Slice.
- Technique 2: Firewalls can be adapted either physically or virtually to fulfil the requirements of the Slice-personalized Zones.

Figure 4-2 illustrates one example of how Zones can be split and personalized by Slice. If, for instance, there are 6 different security zones in the general framework, in the architecture each slice can have 6 isolated zones for a total of 6 by *n* Zones, where *n* is the number of Slices.

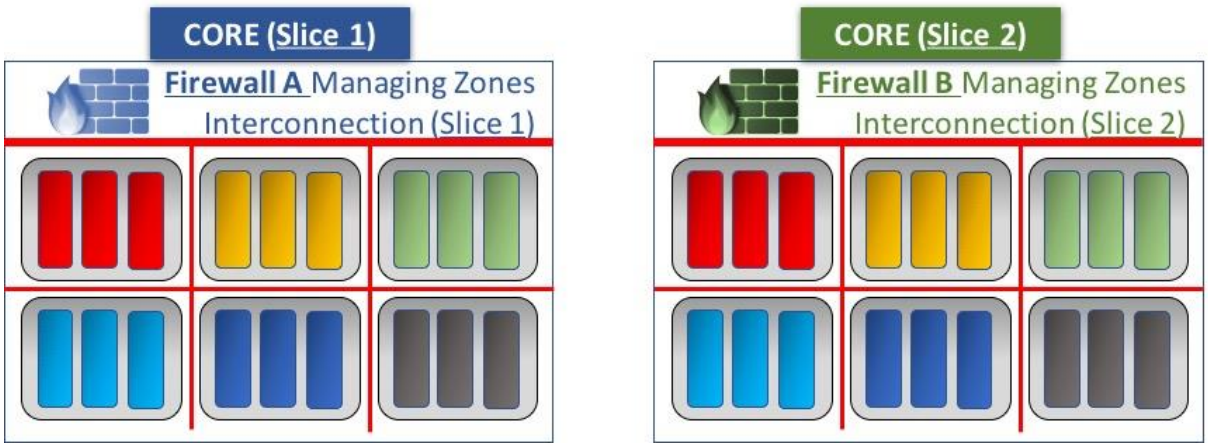


Figure 4-2: Zone Model for different Slices, using different virtual or physical Firewalls

One Specific implementation that has been achieved using this model is presented in Figure 4-3. This implementation consists of an isolated virtual Firewall (Firewall A-2) which is in charge of providing security and control on the interfaces that face the External Network, i.e., N6 or Gi. Since Firewall A-2 is an isolated virtual Firewall that targets exclusively the interfaces for external networks on a specific slice, it is possible and secure to provide full control to the Vertical on that firewall. In that sense, the Vertical can personalize not only the rules that provide security on those interfaces, but also have full access to the respective security logs and security functions that this firewall has.

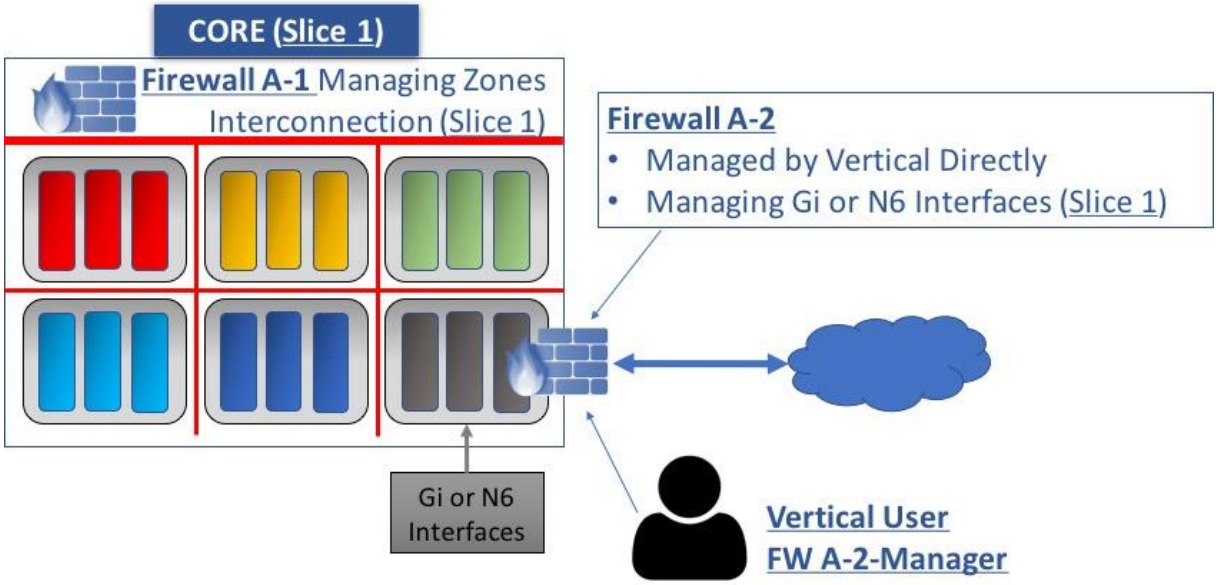


Figure 4-3: Firewall-as-a-Service for and specific Slice on Specific Zone, using different virtual or physical Firewalls

Finally, in Figure 4-4 is presented an extension of the implementation previously presented. In this extension it is possible not only to have Security-as-a-Service for the Gi/N6 interfaces, but also to any of the inter-zone communication that the Vertical users would like to control on its respective slice.

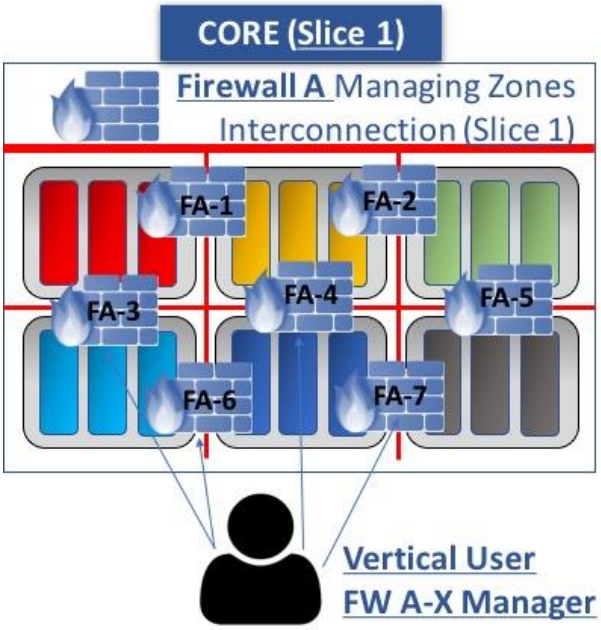


Figure 4-4: Firewall-as-a-Service for and Vertical Selected Zones, using different virtual or physical Firewalls

4.2 Security and Isolation Mechanisms

Network slicing in 5G-VINNI project may allow the concurrent execution of multiple NSIs on top of the 5G-VINNI Facility, satisfying each of their service-specific needs while guaranteeing their independence. The use of a single (shared), multi-domain network infrastructure for this makes isolation a key requirement in the support of network slicing. Isolation across NSIs running on top of 5G-VINNI Facility shall ensure that failures, attacks and lifecycle-related events (e.g. scaling out/in) of one NSI does not negatively impact other NSIs.

In 5G-VINNI D1.2 Section 5, isolation on network slicing was studied from three orthogonal perspectives:

- Isolation in terms of performance - it represents the ability of a NOP to allocate resources to individual NSIs in such a way that their specific service KPIs are always met. This requires the definition of separate resource quotas for individual NSIs, so that congestion or load surges in one NSI do not cause performance degradation in the rest of NSIs.
- Isolation in terms of management and orchestration - it represents the ability of a NOP to ensure independent control of in-slice functions, including network functions (control and user plane functions, arranged into one or more NFV services) and management functions (5G-RAN Controller, 5G-CORE controller Transport Controller, NFV MANO and E2E Service Operations & Management). This isolation perspective also covers multi-tenancy support, for those cases where two different 5G-VINNI customers are served using the same slice.
- Isolation in terms of security - it represents the ability of a NOP to ensure that any type of intentional attack occurring in one NSI must not have an impact on any other NSI. This means that every NSI shall have appropriate mechanisms to guarantee *slice protection* (i.e. the NSI is immune to attacks from any adversary attempting to distort in any means its functionality or features), *slice privacy* (i.e. the NSI integrity and confidentiality is preserved, preventing sensitive information -such as configuration, management, subscriber, accounting information - from being accessed and captured by other entities) and *slice accountability* (i.e. enforcing proper authentication, authorization and accounting within the logical boundaries of the NSI, so no action is performed without knowing the identity of requesters, verifying their rights and properly recording them for further auditing).

Different solution sets addressing the above-referred isolation perspective were also presented, including a comparative analysis among them.

This section aims at complementing the work done in 5G-VINNI D1.2 [3] Section 5 , extending the analysis on network slicing isolation use case scenarios that are quite common in 5G-VINNI experiments, including cross-site slice deployments (Section 4.2.1) and slice execution with MEC capabilities in-built (Section 4.2.2).

4.2.1 Security and Isolation Mechanisms for inter-slice interconnection

The solution sets presented in 5G-VINNI D1.2 are valid when the NSI is deployed within a single 5G-VINNI Facility Site (i.e. a single administrative domain). In this case, the NOP corresponds to the operator of that Facility Site. However, lesson learnt from engagement with 5G-VINNI facility customers (e.g. ICT-19 verticals) have shown that E2E service delivery may typically require the execution of slices involving two or more 5G-VINNI Facility Sites. This brings new isolation requirements on cross-site communications, including control-plane communication (e.g. exchange of signalling information) and data-plane communication (e.g. exchange of IP packets).These requirements depend on the cross-site slice deployment scenario under consideration:

- Scenario #1: E2E Network Slice Federation across 5G-VINNI Facility Sites (see Figure 4-5). In this scenario, there is a single NSI. The NSI constituent network functions are distributed

along the participant sites, each responsible for hosting a subset of those functions. For more details, please see 5G-VINNI D2.1 Section 4.2.1.

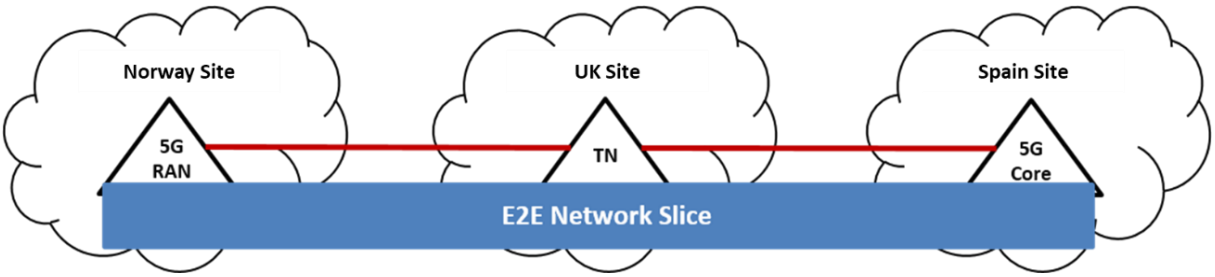
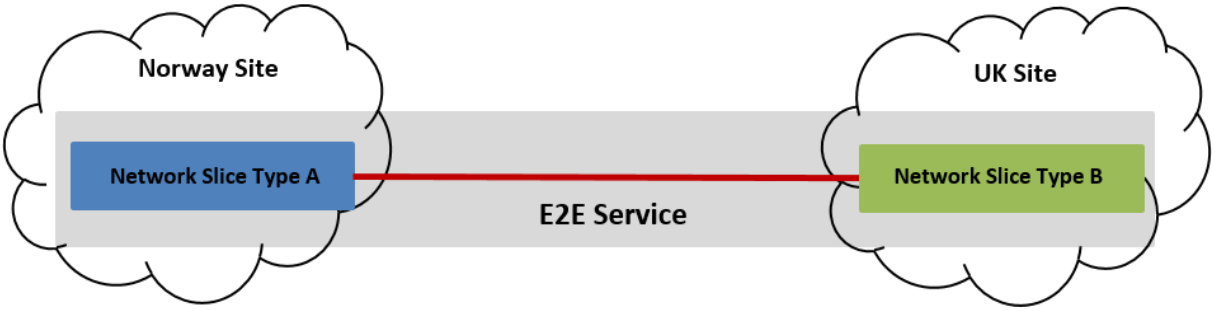
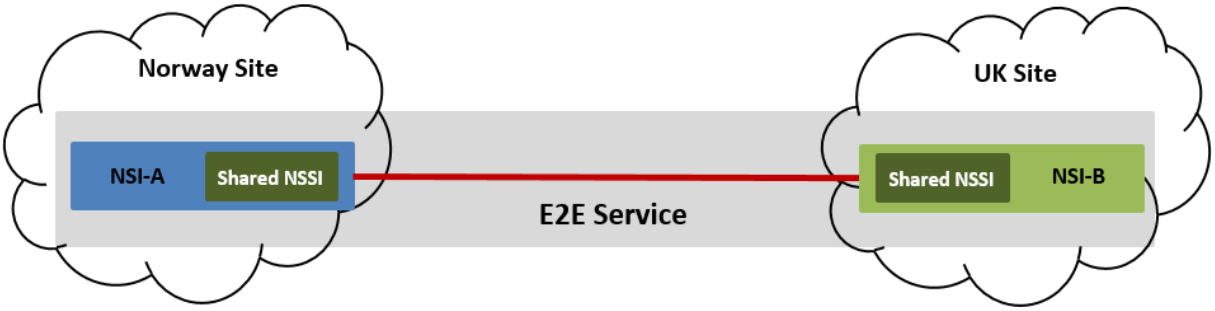


Figure 4-5: E2E Network Slice Federation across three 5G-VINNI Facility Sites

- Scenario #2. E2E Service Level Federation across 5G-VINNI Facility Sites (see Figure 4-6). This scenario is based on the concatenation of individual NSIs, each entirely hosted by a different 5G-VINNI Facility Sites. For more details, please see 5G-VINNI D2.1 Section 4.2.2.



(a)



(b)

Figure 4-6: E2E Service Level Federation across two 5G-VINNI Facility Sites

For the scenario #1, end-to-end NSI isolation requires extending single-domain solution sets from 5G-VINNI beyond the logical perimeter of each Facility Site. To support the exchange of control and data-plane information among sites, 5G-VINNI facility leverages on the use of secure overlay solutions based on IPSec Virtual Private Networks (VPNs). To meet the performance (SLS fulfilment and assurance) across the entire NSI, QoS mechanisms are applied to the VPN conveying data-plane information. These QoS mechanisms allow for an effective Traffic Engineering (TE)-based processing of packets over the IP/MPLS transport network connecting the different facility sites.

In the scenario #2, the concatenation of multiple NSIs results in a construction with multiple 5GC's. This is because (i) every NSI is deployed on a single 5G-VINNI facility site (Figure 4-6(a)), (ii) individual 5G-VINNI facility sites have their own 5GC, thereby taking the role of PLMNs (Figure 4-6(a)). For this scenario, IPSec VPN solutions described for scenario #1 shall be accompanied with 3GPP defined

mechanisms for secure inter-PLMN communication. These mechanisms are based on the introduction of:

- The Security Edge Protection Proxy (SEPP): a 3GPP entity sitting at the perimeter of the PLMN for protecting control-plane communication. The SEPP enforces inter-PLMN security on the N32 interface.
- Inter-PLMN User Plane Security (IPUPS): a 3GPP functionality sitting at the perimeter of the PLMN for protecting user-plane communication. IPUPS is a functionality of the UPF that enforces GTP-U security on the N9 interface between UPFs of the visited and home PLMNs.

5 Research Directions for Future Mobile Network Architecture

This section provides the progress of research programs described in Deliverable D1.2 [3] by highlighting the key technologies and potential research directions for future mobile network architecture.

5.1 Machine Learning (ML) for Edge resilience

One of the expected evolutions of public networks, including 5G, is the massive deployment of IT infrastructure at the edge. This will be motivated by multiple factors, one of which will be for sure the emergence of technologies such as AR/VR, autonomous cars, drones, IOT with smart cities, with efficient real-time processing requirements at the network edge. An additional factor is the emergence of Cloud RAN, based on the virtualization, disaggregation and partial centralization of the RAN components. As a result, a significant part of the network infrastructure is likely to become distributed through a high number of physical locations, which makes global network security and dependability much more challenging to guarantee.

From a security point of view, a decentralized computing architecture will tend to make the network more vulnerable to attacks by creating a high number of potential backdoor entry points. On the other hand, unexpected faults and technical malfunctions will be much more difficult to detect, avoid and mitigate in a massively distributed infrastructure.

This scenario calls for new approaches to handle network security and resilience. The traditional solutions for protection of the infrastructure against disruptive events, either as a result of accidental technical faults or intentional cyber/physical attacks, are no longer adequate to guarantee the fulfilment of carrier-grade targets.

Machine Learning has enabled new possibilities to enable autonomous network management, towards the materialization of self-configuration, self-optimization and self-healing, to cope with the new challenges raised by the proliferation of edge points of presence. Machine Learning provides the required toolset to evolve from a reactive paradigm to a proactive one. Applying ML techniques to available operational data allows the prediction of future problems and the implementation of new processes to prevent degradations from occurring, creating a new pipeline of precocious diagnosis followed by preventive actions. Machine learning enables the precocious diagnosis of network failures, malfunctions and cyber/physical attacks and ultimately avoids the manually intensive management operations.

In addition, network slicing, another key ingredient, provides a solution for on-demand creation and deployment and reconfiguration of network and computation resources by leveraging NFV and SDN techniques and enable the effective mitigation of technical faults and security attacks.

Altice Labs is in the process of setting up a proof of concept on the 5G-VINNI Portugal Facility Site infrastructure, partly in collaboration with the H2020 RESISTO Project [26] (for whom the use case aspects related to network resilience and security are the main target).

Figure 5-1 illustrates the basic concept and use case workflow. The use case is based on a 5G network in which the probability of infrastructure fault is assessed making use of machine learning techniques through continuous analysis of alarms and trouble tickets. If a potential fault is identified, a set of different mitigation actions can be executed depending on the perceived probability of failure or malfunction.

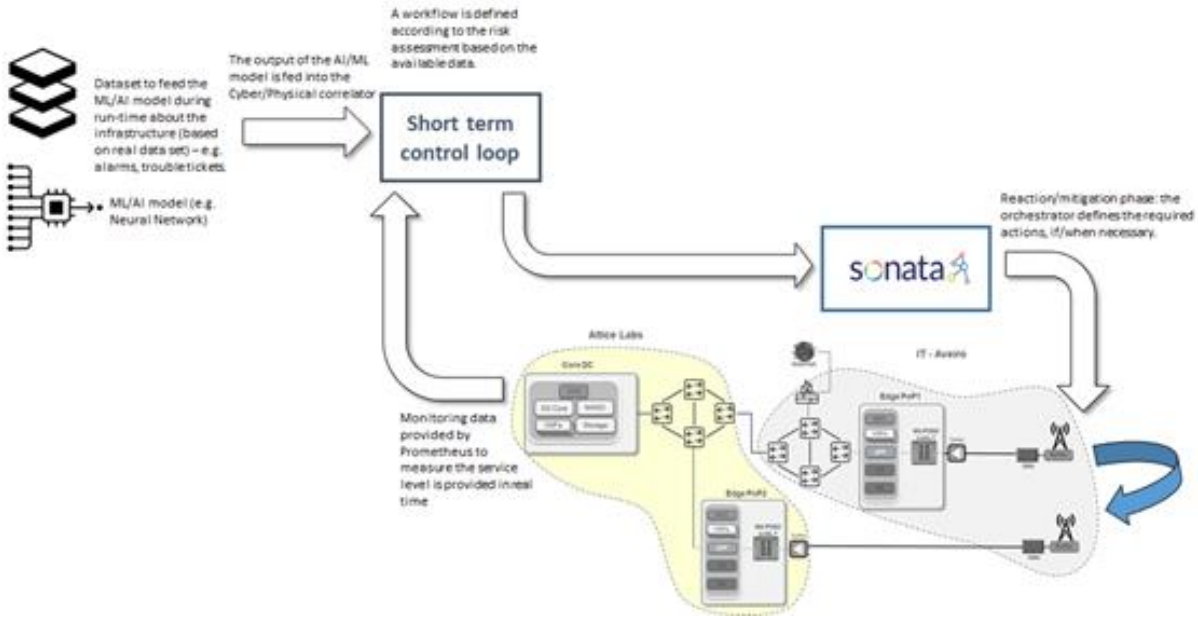


Figure 5-1: Basic use case workflow

The use case is focused on the migration of edge infrastructure from an edge Point of Presence (PoP) to another edge PoP (from Edge PoP1 to Edge PoP2, as illustrated in Figure 5-2). The use case workflow is triggered when the probability of service loss affecting resources (output of the AI/ML model, on the left side of Figure 5-1) goes above a certain threshold (e.g. 35%). The cause is an infrastructure-related problem, e.g. temperature rising in Edge PoP. The event may be accidental, caused by a natural event, or by a malicious action. The Service Provider orders the instantiation of an edge slice subnet, in case the relocation of resources from the affected Edge PoP proves to be necessary. The second phase of the use case is triggered when the service loss probability goes above a second threshold (e.g. 50%). The slice subnet that had been instantiated in the previous step is now activated. The third phase is triggered by a third service loss probability threshold (e.g. 65%). The SP decides to migrate the affected C-RAN and edge components to a different Edge PoP; the service to end users is not supposed to be impacted.

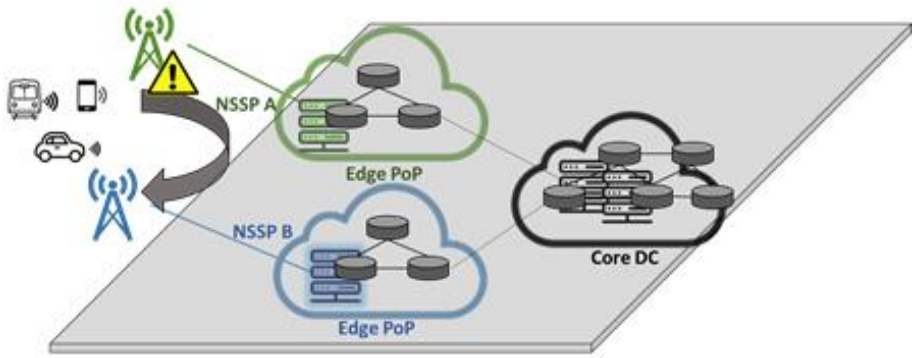


Figure 5-2: Edge infrastructure migration

The use case is built on a data set provided by Altice, related to fixed and mobile infrastructures. The data set includes alarms of KPI threshold violations, alarms of network element failures (partial, total), trouble tickets and inventory of network elements.

5.2 Service Based Architecture (SBA) for Decentralized Network Services

According to 3GPP specifications [9], every NF instance has to register at a repository (called NRF in 3GPP) first. After that, the NF instance can be discovered and provide its services. The registration

procedure is recently extended to contain a certification procedure, wherein an NRF will issue an NF instance a certificate (NF_cert) with the signature of the NRF [27]. In future, it is expected that many NF instances could be deployed closer to the edge of mobile networks. Thus, it will be inefficient if those NF instances at the edge still contact one single NRF entity for registration due to longer latency and the bottleneck effect at the centralized NRF. Naturally, a distributed architecture to realize the NRF will be considered. However, the consequence is that different NF instances may register and then be certified by distinct NRF entities. Such a distributed NRF architecture creates a technical problem to authentication between two NF instances if the two NF instances register at two different NRF entities. The problem is that one NF cannot verify the certificate of the other NF because the certificate is issued by a different NRF entity. Such an authentication is imperative because it is defined as one of key issues of security enhancement for 5G core network in [27].

The authentication between two NF instances can be done in a distributed way but a root of trust is required. A typical solution is a public key infrastructure (PKI). In a PKI, a certificate authority (CA) issues a certificate to a client by signing the certificate with a signing key of the CA. The signing key is usually a private key that is kept confidentially by the CA. The certificate then can be verified by using the counter-part of the signing key (i.e. a public key). Specifically, the signature of the CA in the certificate is verified with the public key of the CA. In this way, a client can verify the certificate of another client if the client possesses the public key of the CA.

In existing solutions, CA's public keys are usually pre-installed on the entities. A typical example is the certificates contained in our browsers, which will be used to verify the certificate of a website. Note that the CA can be built in a hierarchical way where a CA at a higher layer can issue certificates to CA at lower layer. Similarly, the public key of a CA at a higher level can be used to verify the certificate issued by a CA at a lower level.

The main disadvantages of the prior art are summarized as follows.

1. An NF instance can be instantiated at anytime and anywhere as needed. Thus, with several magnitudes more virtual instances than before, configuring, associating and updating every instance authentication information with a centralized authentication are unrealistic, especially if NF instances can be instantiated and de-instantiated in a dynamic way. This causes significantly repeated work to configure every new NF instance.
2. Future mobile networks will deeply integrate with edge computing. An NF instance could be deployed at an edge node whose registration will be done at different NRF entities. According to the current 3GPP standard, where an NF instance will be certified by the NRF the NF instance registers, this means that every NF instance has to possess the public keys of every other NRF entities in order to authenticate an NF instance registering at a different NRF. This will be very inefficient when the numbers of NF instances and NRF entities are large and both of them can dynamically change.
3. In future, NF instances belonging to different domains may interact with each other, existing authentication solutions require a centralized entity, which behaves as a root of trust. Establishing such an interoperability is also quite difficult.

We consider a distributed environment where multiple NRF entities could exist. An NRF entity is assigned to manage one domain wherein all NF instances in one such domain will register at the same NRF entity. We call NF instances that register at one NRF entity the managed NF instances of the NRF.

We also consider that there is no centralized control to the distributed NRF entities. This means that we do not assume the NRF entities having the same owner, which represents a typical multi-operator scenarios where every operator only controls its own NRF. Based on the settings, we propose a decentralized authentication solution by extending the traditional NRF. Such an extended NRF acts as a peer node interacting with other NRF entities and provides authentication services to its managed NF instances.

In specific, our solution introduces two key new features to an NRF entity. The first extension is an NRF entity sharing its own VerTool to other NRF entities and all NRF entities together maintain an identical copy of all shared VerTools. The second extension is to provide a VerTool of a particular NRF to respond to a VerTool retrieval request from its managed NF instance. The general solution framework is depicted in Figure 5-3.

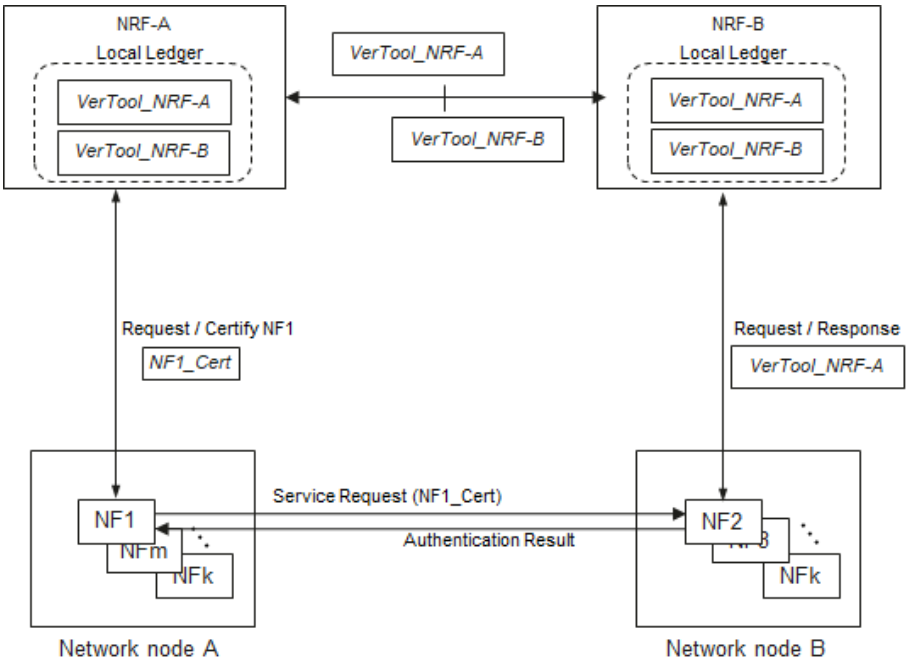


Figure 5-3: Decentralized NF Authentication Framework based on Distributed Ledger

At the beginning, every NF (e.g. NF1) sends a registration request with its profile data to register at an NRF of its own management domain (e.g. NRF1). NRF1 receives the registration request from NF1 and creates an NF entry if all information of NF1's profile data is verified; otherwise, NF1 does not register successfully.

NRF1 issues a certificate to NF1, which is certified with NRF1's own signature that is generated with the private key of NRF1. This certificate can be verified by any party if it processes the corresponding public key. In our solution, we generally call any mechanism that can verify a certification result from an NRF as a VerTool. In this example, it is specific to be of a public key of an NRF.

Meanwhile, NRF1 and NRF2 mutually exchange VerTools that may include both their own VerTools and any VerTool that is received from other NRFs (other than NRF1 and NRF2). In specific, every NRF periodically propagates the newly received VerTools, and those propagated VerTools among distributed NRFs go through a distributed consensus process so that after a while, the same copy of VerTools are replicated across all NRFs. Note that NRFs can employ different types of consensus protocols as needed, depending on the security risk level.

Whenever a registered NF (NF1) needs to access the service of another NF (NF2), the consuming NF sends a service request to the serving NF (NF1 to NF2) with its issued certificate (NF1_Cert).

After the serving NF (NF2) receives the service request, it parses the certificate included in the request, from which the host NRF of the consuming NF (NRF1) can be retrieved. Based on the retrieved NRF information (usually the identifier of the NRF), the serving NF retrieves it's VerTool from the NRF of its own domain (NRF2 in this example) by sending a VerTool request containing the NRF identifier. The local NRF (NRF2) replies to the requesting NF (i.e. NF2) with a response containing the desired VerTool from its local ledger.

With the response, the serving NF can verify the certificate of the consuming NF with the corresponding VerTool, with which the signature of the certificate can be checked. If the local verification succeeds, the serving NF replies the authentication result to the consuming NF. After the authentication is done, the following service access can be granted. Since NF1 and NF2 are symmetric, a mutual authentication can be done accordingly.

5.3 Satellite integration in 5G/B5G: Accent on Dynamic Network Slicing over Satellite

5G introduces the concept of network slicing, in which users may dynamically create E2E services on-demand. While this concept works well in purely terrestrial networks, questions arise when a satellite backhaul is introduced to bridge disparate terrestrial network segments. Can the backhaul link satisfy the network slice's requirements? If so, how does the terrestrial network dynamically activate the satellite backhaul segment of a network slice? It stands to reason that in order to answer these questions, the satellite link must provide some level of visibility and/or control for Third Parties. Depending on the level of service a different resource, including satellite resources, may be required.

Existing satellite networks are largely built as independent systems, creating challenges to efficiency, programmability, and agility; precluding the operators effectively leveraging satellite communications (SatCom) in mobile networks. Further, satellite and terrestrial networks today are managed by independent systems, i.e. each segment holds its own Operations Support System (OSS), Business Support System (BSS) and Network Management System (NMS), although several attempts for system integration have been made in the community for different aspects. Furthermore, SatCom systems have traditionally controlled and managed the network in a static fashion. They allocate resources a priori according to the QoS agreed well in advance with the users and keep the allocation regardless of its use. It requires reconfiguration, typically manual, of the satellite system to modify the satellite service in any way.

However, this is changing for next generation satellite constellations (see e.g., SES's next-generation Medium Earth Orbit (MEO) constellation system "O3b mPOWER" [28]). No longer are satellite resources static but in order to meet the dynamic satellite industry use cases and beyond, the satellite resources need to be dynamic and flexible. To this end, with the advent of SDN and NFV technologies which are driving a huge innovation in networking, there has been a growing interest towards integration of satellite and terrestrial networks into 5G.

The adoption of SDN and NFV changes the principles of network and resources control and management from static to dynamic. Users will request operators to set up and tear down end-to-end services with minimum lead-time, for a variety of applications with utterly different requirements. The recent standardisation work in 3GPP and ETSI, see e.g., 3GPP TR 22.822 [29], 3GPP TR 23.737 [30], 3GPP TR 28.808 [31], and ETSI TR 103 611 [32], are a few examples towards this direction. Furthermore, various EU and ESA funded projects, such the ESA ARTES projects CloudSat [33] and INSTINCT [34], the ESA ARTES project SATis5 [35], the H2020 project VITAL [36], the H2020 5G PPP Phase II project SaT5G [37], as well as the H2020 5G PPP Phase III projects 5G-VINNI and 5GENESIS [38] have developed the key technology enablers for SatCom integration into 5G networks by leveraging on the SDN/NFV paradigm.

However, limited work has been conducted so far on network slicing over integrated satellite-terrestrial 5G networks and this corresponds to a key future direction towards satellite-enabled B5G systems. The network slicing concept leverages the SDN and NFV technologies promising innovative service integration strategies on shared network infrastructures. Notably, among the prior related work, the ESA ARTES project SATis5[8] has successfully demonstrated through an SDN/NFV/MEC-enabled PoC testbed the benefits of satellite integration into 5G for most of the 5G use cases deployments.

Among other key innovative features, such as the 3GPP 5G core network seamless integration with satellite ground segment, the SATis5 testbed provides an edge-central 5G core network functionality split, which is expressed as the split of the 5G system between edge and central network. Building upon synergies with the SATis5 testbed, the Berlin and Moving Experimentation Facility Sites developed within 5G-VINNI provide SatCom-enabled fixed and nomadic 5G edge nodes implementing various slice models that adopt the edge-central 5G core network functionality split for various 5G use cases deployments, including private mobile and nomadic network deployments.

Nonetheless, an effective network slicing requires tight control on the QoS and therefore on the infrastructure performance. Building upon the results of such past relevant projects and elaborating on the applicability of network slicing over integrated satellite-terrestrial B5G networks, future research direction will enable SatCom systems to dynamically allocate network resources in order to set up new services, while continuously managing them to meet the performance requirements without compromising resources unnecessarily. The framework on resource management API defined within SATis5 could be further advanced to integrate with technologies like SDN, SD-WAN and AI/ML to dynamically reconfigure the satellite resources depending on the learned network KPIs rather than waiting for manual intervention. Nonetheless, in order to be able to understand and to react automatically to specific triggers, there is a need for advanced R&D in which these optimal automatic reactions are determined for a large set of situations.

In order to support the new dynamic satellite use cases, driven by 5G and other verticals, the satellite resource and service manager elements are introduced into the Satellite Ground Segment (SGS). These are not new elements in the satellite world, but they are being enhanced in order to make the SGS dynamically configurable by external third-party resource and service orchestration systems. The SGS satellite resource manager will manage its satellite resources based on the information received for the resource orchestrator via the resource management API. The satellite operator will do likewise for the space segment in order to align resources between the SGS and space segments. The resource orchestrator that will reside in the satellite network operator domain, will coordinate the SGS and space segment resources to meet the requested satellite quality of service (e.g. more capacity).

Along with exposing an API to manage the SGS satellite resources, the SGS will also provide a service level API interface which will allow a service orchestrator to request a service from the satellite SGS. This is separate from the resource management sub-system and orchestration, which focuses on the satellite resources, as there may be more factors involved here other than satellite resources (e.g. user profile limits, regulatory constraints, class of service). The SGS will notify the management and orchestration system if it can provide the service. One consideration may be the available satellite resources, but this is not the only aspect.

Within the context of the proposed satellite service-level API, a *satellite service* can be comprised of the following:

- One or more pre-existing SVNs (*Satellite Virtual Networks*) over a particular satellite constellation. These are logical satellite links, which a satellite network operator configures and deploys via the satellite NMS. The satellite network operator may configure each SVN with its own specific link characteristics, such as QoS, Committed Information Rate (CIR), Maximum Information Rate (MIR), satellite type (e.g. GEO, MEO), etc. for differentiated Over the Air (OTA) services.
- A *Satellite Gateway.* This is an endpoint, towards which the Third Party directs their IP traffic, for transmission over a specific satellite service.
- A Satellite Service Identifier (SSI). This is an alphanumeric string, which intuitively describes the underlying level of service provided. The specific characteristics of each service type is defined may be agreed upon at a business-level by the satellite network operator and the Third Party, and subsequently deployed as SVNs by the satellite network operator. The Third

Party later uses the service identifier when invoking the Satellite Service API to query and/or modify individual satellite services. Note that the mapping between 5G NSSI and the Satellite SSI has to be considered by each of the network operators during the deployment phase, as architecture wise it is too generic. Nonetheless, this mapping between 5G NSSI and the Satellite SSI has to be executed in order to be able to differentiate the deployed slices.

One of the added values beyond the state-of-the-art will be the design of a unified orchestration layer that allows to manage resources and deploy services spanning the different and heterogeneous network segments involved in the end-to-end communication. A key enabler to reach this goal would be the adoption of an appropriate level of abstraction in the E2E service specification offered by the virtualized SGS orchestration layer, such that service providers and/or customers are allowed to express service requirements, as well as to manage service lifecycle, by using high-level specifications that are as much independent as possible of the particular domains, solutions, and technologies involved in the requested service deployment. This is even more critical in view of seamless terrestrial-satellite integration as part of a more general 5G end-to-end service deployment, involving multiple network infrastructures. A significant step beyond the current state of the art in service abstraction is represented by the intent-based approach to network service specification. Expressing service requirements as an "intent" allows to declare the requested service in terms of "what" must be achieved and not "how" to achieve it, leaving the underlying orchestration and management mechanisms to deal with the specific details of how to verify, fulfil and maintain the request. This approach allows complete separation of the high-level service expression from the vendor- and technology-specific implementation, achieving the level of abstraction mentioned above and fostering improved interoperability.

In a nutshell, progress beyond the state-of-the-art will address the design of a new satellite resource management approach based on a suitable API level definition, allowing the dynamic configuration and reconfiguration of satellite resources and services in order to provide real-time services akin to what is available in the cloud and terrestrial domains. To this aim specific APIs will be defined, with a particular attention on the intent based approach allowing to achieve a service abstraction where each service can declare what to do, avoiding declaring how. As a continuation of the prior work started in relevant R&D projects, the major advancement would be the ability to have flexible payloads on the satellite which opens up to dynamically changing the satellite resources and requesting same in real-time. Such dynamic configuration and reconfiguration of satellite resources is needed in next-generation satellite communication carrying flexible payloads (e.g., SES-17 GEO, O3b mPOWER MEO, etc).

5.4 Flexible Architecture for Verticals

The research directions of enhanced 5G flexible architecture were proposed in 5G-VINNI D1.1 [1]. The progress of research activities on architecture flexibility to support verticals is discussed in 5G-VINNI D1.2 [3] and 5G-VINNI D1.4 [2].

In this section, the progress of architecture flexibility to support private 5G networks, a so-called Non-Public Networks (NPN) is studied. In particular, a number of open issues and challenges will be discussed as a guideline for future research direction and as a high level direction for the future work of PNI-NPN standardization in 3GPP mobile network architecture proposed in [39].

- **Security Gateway between PLMN and NPN**

Secure communication and interworking between PLMN and NPN is a fundamental and critical feature to be supported in PNI-NPN infrastructure. 3GPP [9] defines a Security Edge Protection Proxy (SEPP), a non-transparent proxy, between PLMNs as inter-PLMN control plane interface to support the functionalities of (i) message filtering and policing, and (ii) topology hiding. For SEPP to be effective a Service Communication Proxy (SCP) [9] may be used in parallel, enabling the unification of the control plane communication between two domains into a single interface to increase the

coordination of the exchanged messages and provide the means for a better vendor interoperability. Although SEPP is introduced as a security proxy between PLMNs (e.g., for Roaming scenario), a similar functionality of SEPP might be applied as a security gateway between NPN and PLMN. Due to the requirement of interworking feature between NPN and PLMN, the security and privacy control mechanisms should work in bi-directional approach. In addition, NPNs and NPN services will have different levels of security and privacy requirements, hence the enhanced mechanisms should be designed to enable a secure communication and internetworking between PLMN and NPNs.

- **NPN management for verticals**

According to recent surveys, there exists industry verticals who have expressed their willingness to take the lead in the construction and operation of their NPNs, thus becoming NPN operators. While fulfilling this should be easy in stand-alone NPN scenarios, as long as the verticals have deep networking experience, the situation changes when PNI-NPN scenarios are considered. In such a case, to take the role of NPN operator, a vertical needs to extend the scope of his management domain beyond the private segment of the NPN; indeed, the vertical needs to retain some control over the slice made available by the MNO, which constitutes the public segment of the NPN, using the NSaaS model as presented in Figure 3-16 . For this end, slice capability exposure mechanisms are required. Slice capability exposure can be defined as the ability of a network slice provider to securely expose management capabilities of a slice instance (e.g. policy administration, execution of performance assurance and fault supervision activities, lifecycle management of network functions deployed on virtualized environments as VNFs) towards an authorized customer. By regulating this exposure, the MNO can define how much control the vertical can take over the slice to freely adapt its behavior in terms of performance, functionality and scalability.

Note that the slice capability exposure shall be agreed on a per vertical basis, considering that different verticals could want to have different levels of control over their slices. This fact makes it necessary to define exposure levels, each allowing the vertical to gain access to more or less management capabilities.

- **Non-repudiation for cross-domain NPN operation**

This is in line with the above-mentioned topic. To allow a vertical to consume management capabilities from a slice made available by a MNO, the management systems of both actors need to interact with each other, exchanging request-response messages between them. To make these systems verifiable, trusted orchestration and control stacks, non-repudiation mechanisms must be defined. The non-repudiation principle means that each pair of actors taking part in any interaction can demonstrate that a certain request or response message has been effectively generated by the other one, relating them to previous relevant messages, and associating them with a consistent temporal line. To achieve this, both the MNO and the vertical shall save their (equivalent) evidence of the exchanges on a private trusted store. These evidences will provide means for external verifiability of all messages exchanged, allowing system auditability (i.e. to keep audit trials for traceability purposes), and further applications enabled by it.

- **Feasibility study on emerging business models**

The deployment of PNI-NPN scenarios enables a synergistic relationship between MNO and verticals, allowing one actor to offer services to the other. In NSaaS, where the MNO taking the role of slice provider, the service delivery model is typical in industry 4.0 scenarios. However, public venues, including stadiums, museums or transportation hubs are another use case for NPNs. In such venues there are dense concentrations of end-users that demand advanced services, including Virtual/Augmented Reality. However, appropriately covering these venues is expensive, hence it is difficult to define a business model from the perspective of a single MNO. Instead, an alternative business model is emerging based on the figure of a neutral host, whereby the public venue owner (i.e. the vertical) invests in network infrastructure, which is used for his own private services and also

leased to MNOs, allowing them to provide connectivity to end-users. All these business models shall be validated using techno-economic analysis and proof-of-concepts.

5.5 Analytics Driven Service Automation

5G-VINNI proposed the slice architecture for 5G network slicing. The architecture aims at service orchestration but does not cover service assurance and thus does not really solve the zero-touch automation issues as expected. As service assurance is a key mechanism to guarantee the success of 5G network slicing, it is important to propose an architecture for service assurance in the context of network slicing.

To align with the slice orchestration architecture proposed in [1], a hierarchical service assurance architecture is proposed as shown in Figure 5-4 [40].

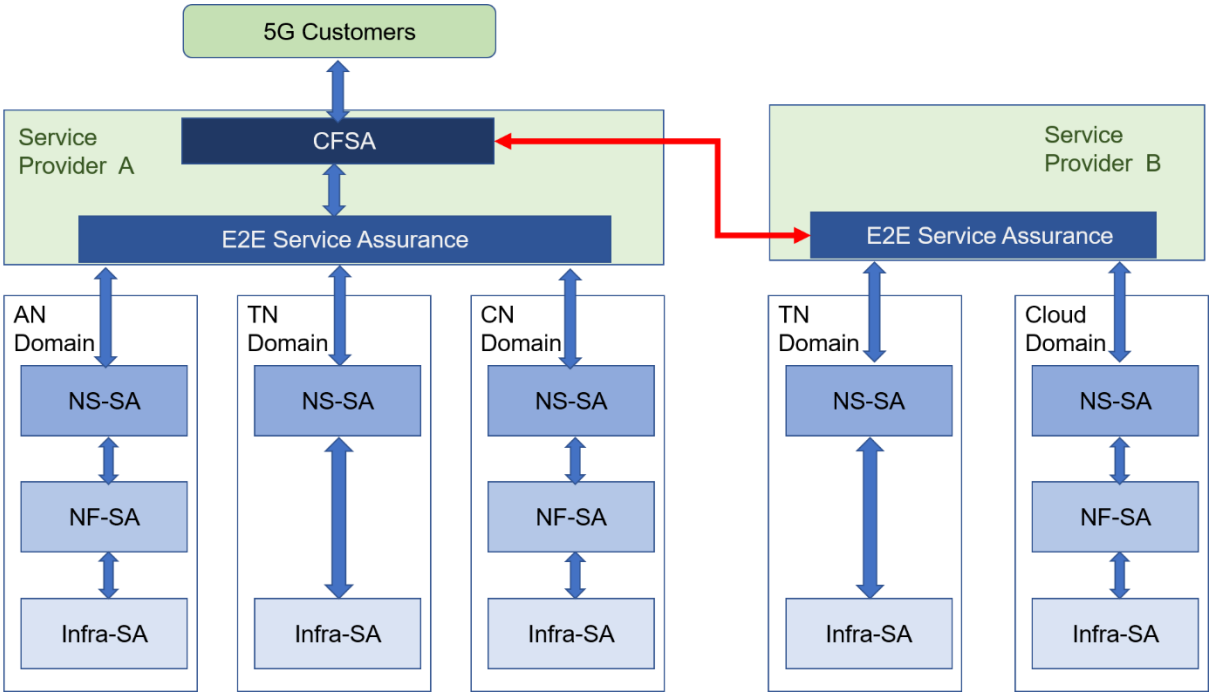


Figure 5-4: Service assurance architecture for network slicing

The bottom three layers, Infrastructure-SA, NF-SA, and NS-SA correspond to the three NFV layers defined in the ETSI MANO framework, infrastructure, Network Function (NF), and Network Service (NS), respectively. The E2E Slice Assurance (E2E-SA) is responsible for assuring the network slices provisioning for the CFS, whose assurance is achieved by the CFS Assurance (CFSA). This hierarchy reflects how a CFS is constructed recursively from simpler components.

The top layer CFSA interacts with the 5G customers and can be offered by the service provider that receives service request from the 5G customers (e.g., Service Provider A in Figure 5-4). 5G customers usually request communications services rather than network slices. The CFSA translates the customer's service request, e.g., service level agreement (SLA) and/or quality of experience (QoE) requirements, into the SLA suitable for individual slices that could be used by E2E-SA. If a CFS requires network slices provided by multiple service providers (e.g., service provider A and B in Figure 5-4), the CFSA decomposes the CFS-SLA into SLAs for each E2E-SA. Furthermore, CFSA receives and aggregates service assurance related data from each E2E-SA (the red line from Service Provider B and blue line from Service Provider A) to generate an overall service assurance view for the CFS and assess if the CFS-SLA is guaranteed.

E2E-SA is responsible for the network slice provided by one administrative provider, e.g., in service provider A or B. The E2E term is used because one slice often spans multiple technology domains,

such as AN, CN and TN. Each domain has its own service assurance and realized by NS-SA in Figure 5-4. Similar to CFSA, E2E-SA decomposes the slice SLA into the SLA of each domain, and gathers and aggregates service assurance related data from each domain to generate an E2E view of the network slice within the provider's domain. The similar relationship exists between NS-SA and NF-SA, and between NF-SA and Infrastructure-SA.

Service assurance is realized in a distributed way to allow for flexibility and scalability, e.g., each layer and domain has its own service assurance, which is referred to as local service assurance. Local service assurance can evolve independently as some local service assurance are developed faster than others, e.g., infrastructure-SA and NF-SA is more developed than E2E-SA. The domain can flexibly construct its service assurance. For instance, the TN (e.g., Software Defined WAN) does not have NFs and thus contains infrastructure-SA and NS-SA (Figure 5-4). Distributed service assurance can separate the layer and domain service assurance issues from the E2E-SA and CFSA issues. Local service assurance has more detailed and in-depth knowledge of the assured entity and thus could make decisions more quickly and even accurately, especially when edge clouds are deployed in 5G. More importantly, with local service assurance, the complexity of assuring CFS is significantly reduced. In this way, the abstraction feature inherited from NFV is well maintained such that the changes in one layer or domain does not affect other layers. For example, if the infrastructure layer is switched from virtual-machine (VM)-based to container-based, the corresponding change of orchestration and assurance from OpenStack-based to Kubernetes-based is agnostic for upper layers. On the other hand, distributed service assurance may suffer from performance degradation for the network slice and CFS, especially when local SAs operate independently. Therefore, coordination is demanded. The higher layer service assurance is responsible for coordinating the lower layer service assurances, e.g., by properly and effectively aggregating service assurance data from lower layer service assurances. How coordination is achieved relies on the functional components of each service assurance layer.

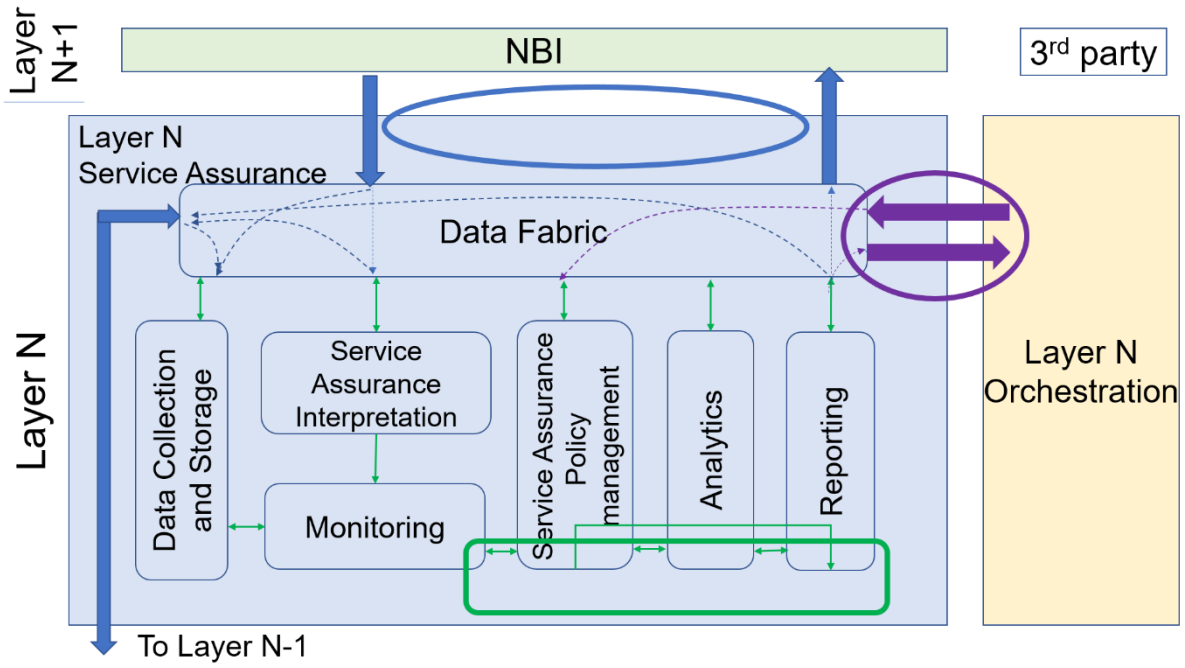


Figure 5-5: Service assurance functions in layer N

Each service assurance layer contains at least seven functional components, including four basic service assurance functions and three enhanced service assurance functions (Figure 5-5). Like conventional service assurance, there are four basic service assurance functions: monitoring, data collection and storage, analytics, and reporting. Three enhanced functions are introduced to support

coordination and enable automation: service assurance interpretation, service assurance policy management, and data fabric.

Although the proposed service assurance architecture allows for creating closed control loops to enable zero-touch automation internal to the network and service provisioning, it is more desired to expose service assurance services to external or 3rd party customers as a means to take advantages of their powerful AI/ML capabilities and enhance the service assurance performance. Accordingly, the service assurance architecture should be modified to reflect the service exposure capabilities. In [41], a modified service assurance architecture is proposed with inclusion of four service exposure levels (Figure 6.6 in [33]), in accordance with D3.1. In addition, the service assurance functions of each layer are assessed with respect to their exposure to external customers with special AI analytics capabilities (Figure 6.7 in [33]). The overall service assurance performance and achieved automation level depends on both the internal service assurance functions and the service exposure capabilities.

5.6 Resource Scheduling for Service Function Chaining

NFV has introduced a high degree of flexibility for orchestrating service functions. The provisioning of chains of service functions requires making decisions on both placement of service functions and scheduling of traffic through them. The placement problem can be tackled during the planning phase, by exploiting coarse-grained traffic information, and has been studied extensively. However, runtime traffic scheduling for optimizing system utilization and service quality, as required for future edge cloud and mobile carrier scenarios, has not been addressed so far.

In our work published in [42], we filled this gap by presenting a queuing-based system model to characterize the runtime traffic scheduling problem for service function chaining. We proposed the integer allocation maximum pressure policy (IA-MPP) for SFC, Service Function Chain (SFC) scheduling, a derivation of maximum pressure policy (MPP) [43], which we showed is throughput optimal. It is also asymptotically optimal for minimizing a cost function of buffer occupancy levels in the network, providing approximate guarantees on latency. Furthermore, we showed that the time complexity of IA-MPP is bounded by a linear term on the number of sites in the network. Importantly, IA-MPP requires no a priori information about network traffic patterns. The proof of optimality and other properties of IA-MPP, as mentioned above, can be found in the paper.

Based on practical constraints in large deployments, we also presented a novel distributed variant of our solution dubbed multi-site cooperative IA-MPP (STEAM), where a scheduler instance is running at each site using only site-local state, and is invoked for batches of packets. We studied the performance of STEAM using a packet-level simulator as well as a prototype implementation based on Data Plane Development Kit (DPDK) [44]. We observed that STEAM performs close to the optimum (IA-MPP) and significantly outperforms (possible adaptation of) existing static or coarse-grained dynamic solutions. Specifically, STEAM improves resource usage, requiring much fewer resources to achieve similar service quality.

We conducted performance evaluation with large-scale simulations as well as a prototype implementation. Our packet level discrete event simulator simulates scenarios in compliance with RFC 7665, comprising the network topology including link latencies, packet handling at Service Function Forwarders (SFFs), Service Function Instances (SFIs), and servers, the processing of the SFIs running on servers, and the schedulers. We studied success rate, which is the ratio of successfully served packets to the total number of arrivals. The higher the values for these metrics, the better the solution. We compared the performance of our solutions with variants of existing solutions:

- OSPP: As a variant of [45], [46], the offline static planning policy (OSPP) performs offline planning ahead of traffic arrival, but applies runtime load balancing to react to sudden traffic changes.
- SGHP: The second baseline is shortened greedy heuristic policy (SGHP), which adapts the most recent existing heuristics SGH [47] and SPH [48].

Figure 1 shows the performance results for a topology in the image of publicly available information on data center locations of an Internet service provider (ISP) [49]. The topology comprises 50 sites, each with one SFF and 6 to 12 servers. There are 10 SFs in the network with a total of 1600 SFIs across all sites and 30 SFCs each with up to four SFs. An SFC in the system is specified by an ordered set of SFs that a flow packet should be processed through. In addition, each SFC is given a set of quality of service (QoS) metrics that the handling of packets undergoing the chain has to conform to, which in our considered scenarios contains the end-to-end delay. The flow arrivals are time-varying and bursty. We use a Markov modulated process (MMP) [50] to simulate flow arrivals. Each flow randomly selects an existing SFC and a pair of ingress/egress SFFs. Each SFC has a QoS deadline, set as a function of the service rates of involved SFs, which specifies the maximum allowed latency observed by a packet. A packet is successfully served, if it can be served within its QoS deadline.

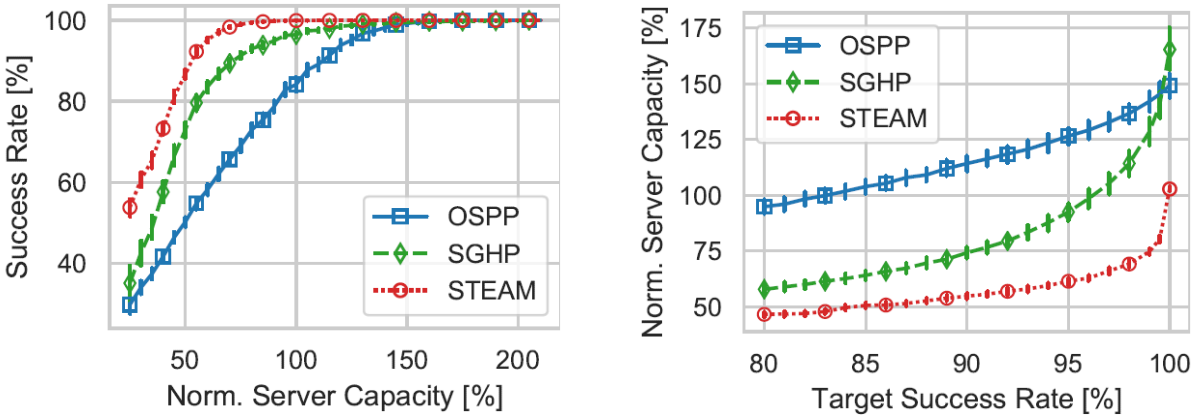


Figure 5-6: 50 sites running distributed scheduling: STEAM vs baselines.

We compare STEAM with baseline solutions, all using only site-local state information. The experiment conducts varying server capacities at sites to reach full success rate. Server's capacities are normalized to STEAM's at 100% success rate. Figure on the left zooms on higher percentiles, highlights the gain achieved using STEAM. We observe that STEAM shows best performance, reaching full success with 50 - 70% less server capacity. This is as STEAM, driven by our optimal solution, tries to maximize the resource multiplexing in the network and hence can efficiently use available resources.

Moreover, we have implemented a prototype of STEAM based on DPDK, including the NSH protocol [51] to check feasibility running on a standard server (each 2 x E5-2630, 128GB memory, Intel X520-2 10G SFP+; Linux 4.15.0-48- generic; DPDK 18.11.1). Our results show that STEAM can achieve 1-4 million scheduling decisions per second, using 1 CPU core, and hence can be run in real-time. For a completed, more detailed study, covering different scenarios and scales, we refer the readers to our paper.

In summary, we proposed a runtime SFC scheduling policy, which can be deployed in a distributed manner, and demonstrated that, given fixed resource capacities, it can achieve significantly higher success rates and better service quality than existing static or coarse-grained solutions. It thus decreases the amount of resources in the network that need to be allocated to provide a target quality of service guarantee.

5.7 Network Edge Management Infrastructure

5G enables the concept of distributed network across various edge nodes. One of the key limitations in the deployment of a data acquisition network across a distributed environment is the capability of automated management of such an infrastructure. Currently, large number of operations are executed by human administrators, which also is a factor that leads to decreases in deployment capabilities. The main functionality of Network Management is to provide system configuration,

monitor & log its performance as well as to observe and mitigate faults that would help in orchestrate system-wide operations through the application of policies. Figure 5-7, illustrates the implementation of the Network Edge Management Infrastructure (NEMI) network management solution. This describes the implementation in a layered architecture divided into Active System, Data and control plane. In Figure 5-8, the remote edge management is mapped one-to-one with the reference architecture described in Figure 5-7. Additionally, it demonstrates the components and technology selection of various layers and processes.

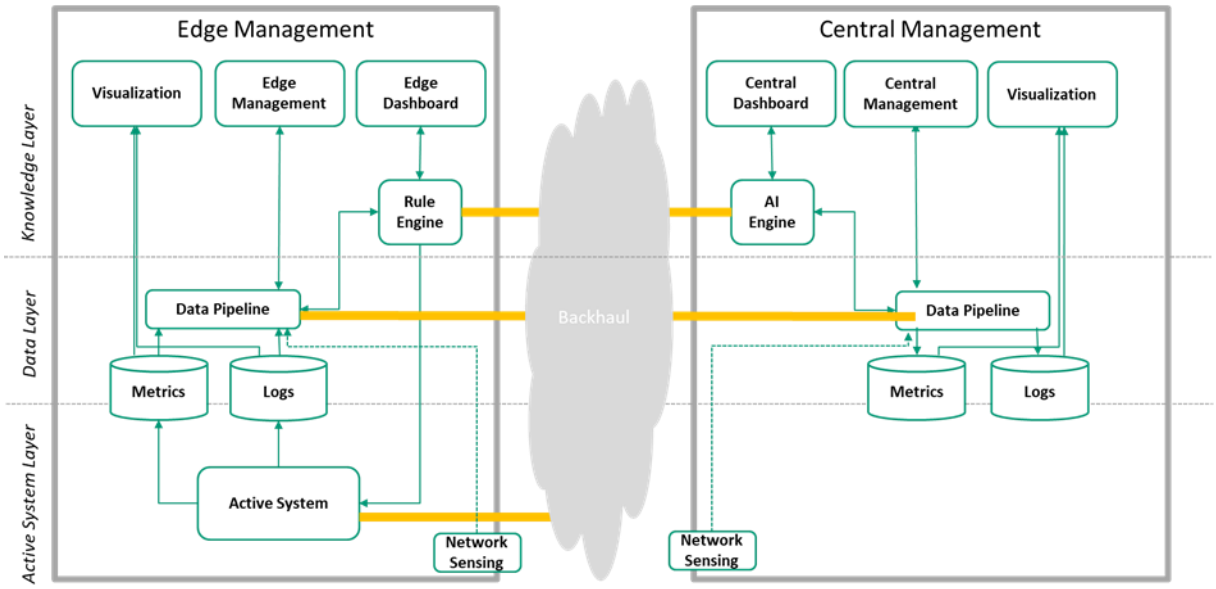


Figure 5-7: Network Management Architecture

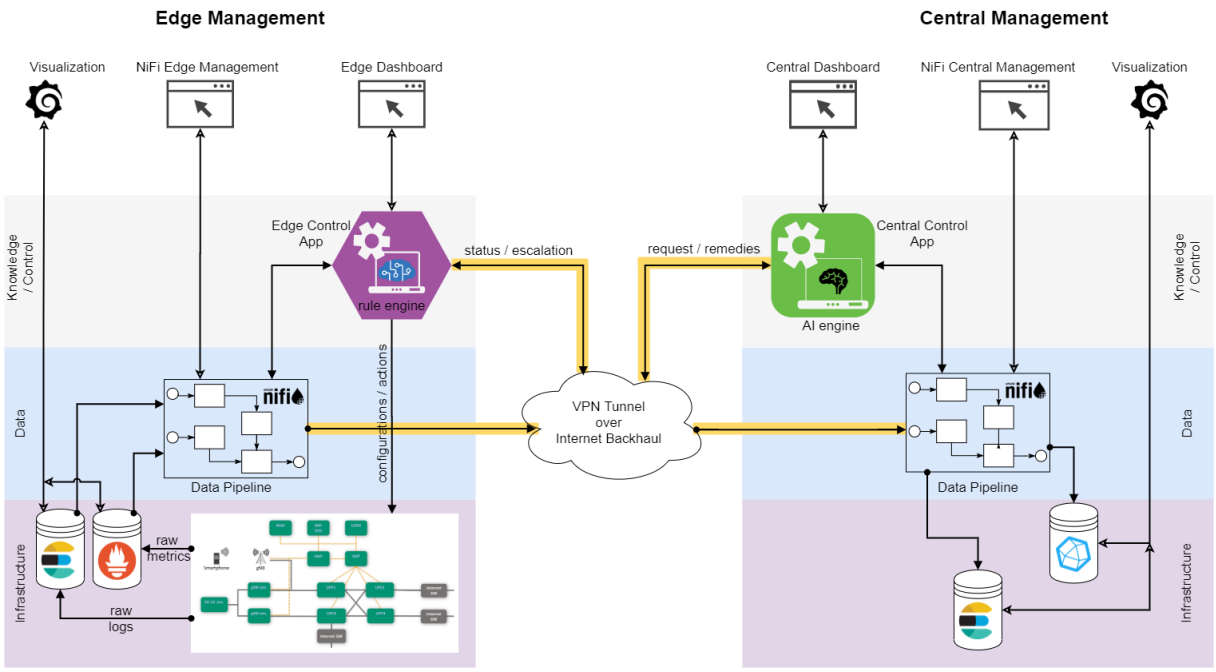


Figure 5-8 : NEMI Remote Node Management

At Edge, on the infrastructure /active system layer, the raw metrics and raw logs are collected from the active systems. The hardware, OS as well as process specific metrics are collected from the active systems and persisted into Prometheus, an open source monitoring system tailored with an efficient time series database. Prometheus works on pull based mechanism, where the agents or exporters collects the metrics and Prometheus pulls those metrics with the help of the agents. Raw Logs are

pushed to Elasticsearch [52] with syslog. This set-up ensures that historical system data is archived and retrievable for both online and offline processing. Tools such as Grafana [53] and Kibana [54] are used to explore and visualize the metrics and logs. At central side, the aggregated metrics and logs collected from several edges are stored in InfluxDB [55] and Elasticsearch repositories respectively. InfluxDB is also an efficient time series database with push based approach.

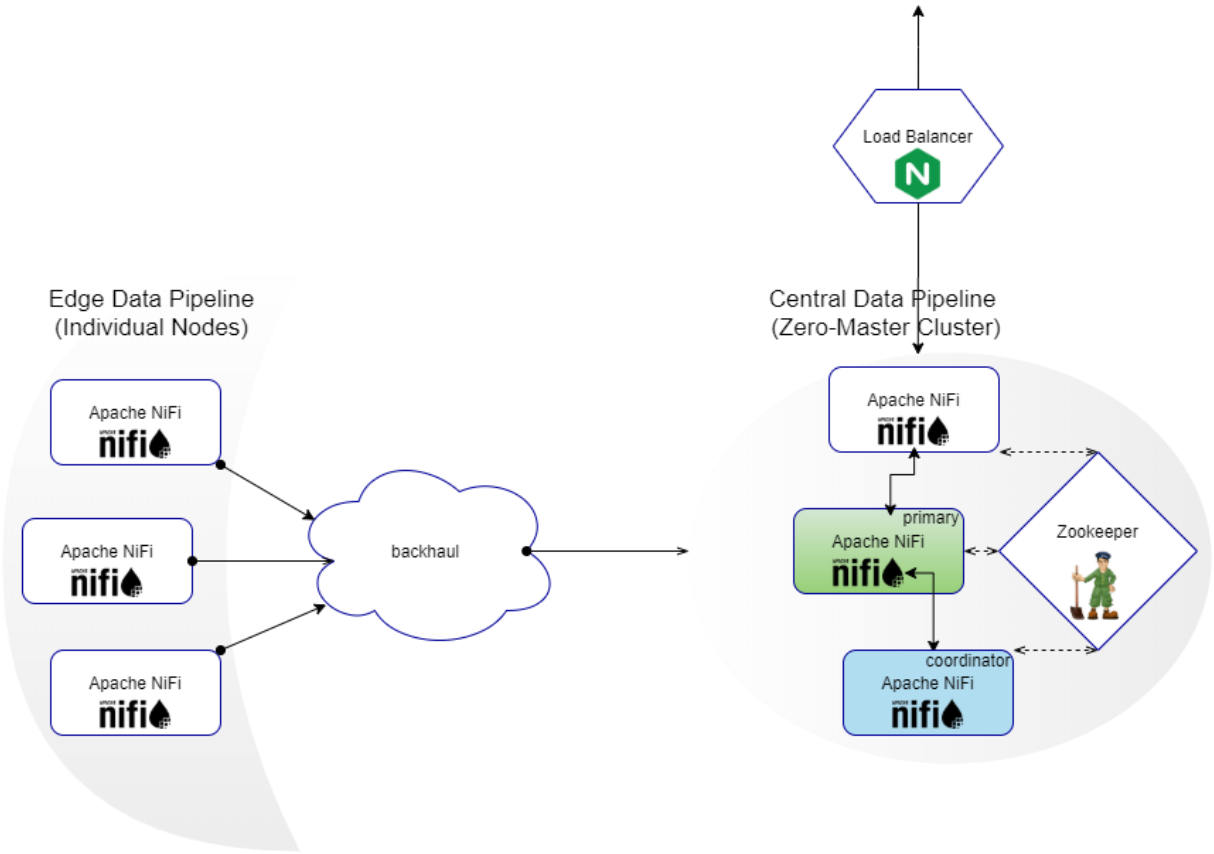


Figure 5-9: Data Layer Implementation

Both in the edge and in central, the data layer is composed of a peer-to-peer based distributed data processing component, actualized through Apache NiFi instances. Apache NiFi [56] provides a powerful and reliable data Flow management system with backpressure support that tracks the data flow and provides the provenance for the data it handles from the start to the end. Thus, in NEMI, Apache NiFi is used for defining the various data pipelines for collecting metrics and logs and for routing them reliably from the edge to the central. Figure 5-9 illustrates the communication and implementation of data layer between edge and central. In order to process and maintain the data coming from various edges, central side data layer is equipped with a cluster of Apache NIFI. The cluster is configured in a Zero-master cluster formation. This implies that each cluster node performs the same data processing tasks but on different data sets. Apache ZooKeeper [57] is set up for the NiFi cluster coordination. For North bound clients for the data layer at central, Nginx is configured for load balancing and reverse proxy. Thus, requests to this single endpoint are routed to the individual cluster nodes in a balanced manner.

Knowledge layer deals with the monitoring and observation of the active system and automated actions (either based on rules or AI) to manage and operate the system. On the edge-side, the reaction to system events must be near-real time. The control-loop at the edge handles cases that are autonomic in nature, i.e., without cognition / intelligence and the need for deeper analysis. Thus, on the edge-side, a rule-engine, is realized by Drools Fusion [58], a complex event processing solution. Drools allows the definition of rules in its own Domain Specific Language (DSL) that is both human readable and machine friendly.

On the central side, data and events escalated by all the edges are processed. Events escalated by the edges require cognitive actions, either in the form of an AI or human intervention. Thus, the edge-central control-loop is much slower than the edge-local control loop. The control app at the central is envisioned to be an AI based knowledge engine.

Figure 5-10 and Figure 5-11 shows the bootstrap and registration of edge nodes with the central cluster. Edge tries to register itself with the central control app on starting of the edge. Once, discovered by the central, the central control loop requests for the capabilities of the edge. Capabilities are mapped to the set of processors and processors groups outlined in the data layer. Once the capabilities of an edge are retrieved, the edge is considered to be registered. Upon change in the capabilities, the edge register itself with the central cluster. The active connection between edge and central is verified by the heartbeat. In special circumstances where the central control app is down, the edge will wait until the central app is up before trying to register itself. All communication between the edge and central control apps are carried out over REST APIs. The status and management of the edge and central control apps are carried out through their respective dashboards.

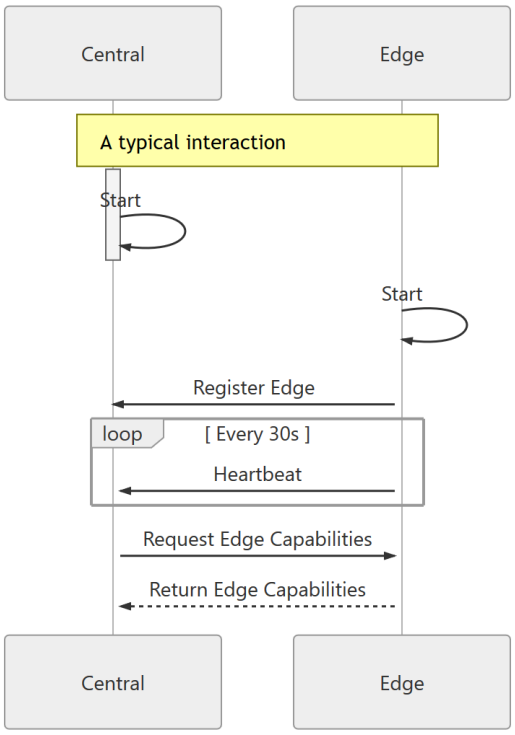


Figure 5-10: Typical Edge Central Registration

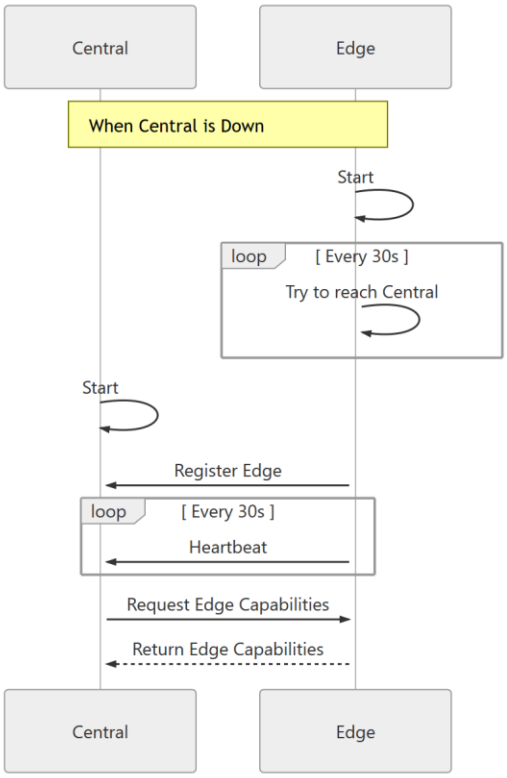


Figure 5-11: Edge Central Registration when central is down

The edge-local control loop**,** which is illustrated in Figure 5-12, is the fastest control loop, where the rule-based decisions are taken directly at the edge. They are autonomic in nature, i.e., reactive to the observed conditions. The data (metrics and logs) from the active system (available in their respective metrics and log repositories) are processed and streamed using the data pipelines to the edge rule-engine based control app. The rule-engine which is a complex event processing (CEP) solution reacts to the events in an event-condition-action paradigm. When an action to the causal event and applicable condition is found at the edge, the appropriate active system action (e.g., reconfiguration / allocation of additional resources) is carried out.

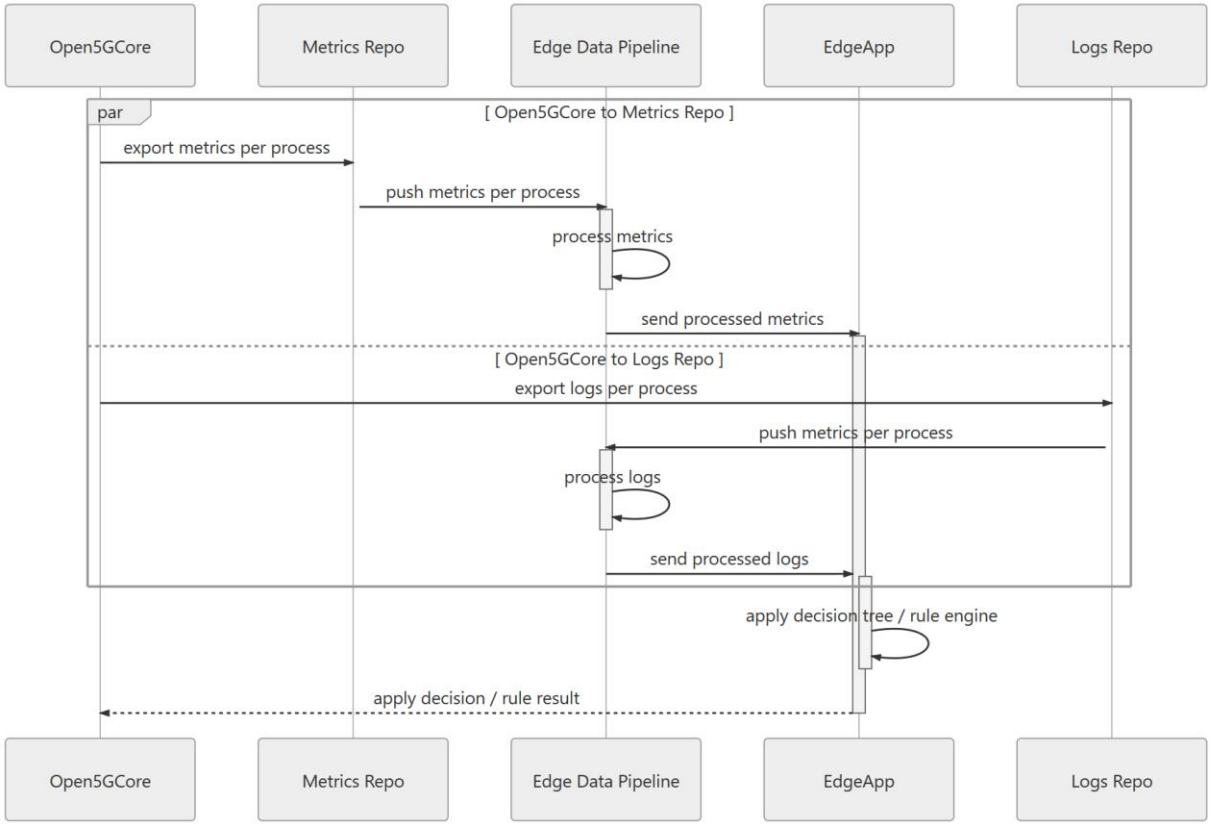


Figure 5-12: Edge Local Decision Control Loop

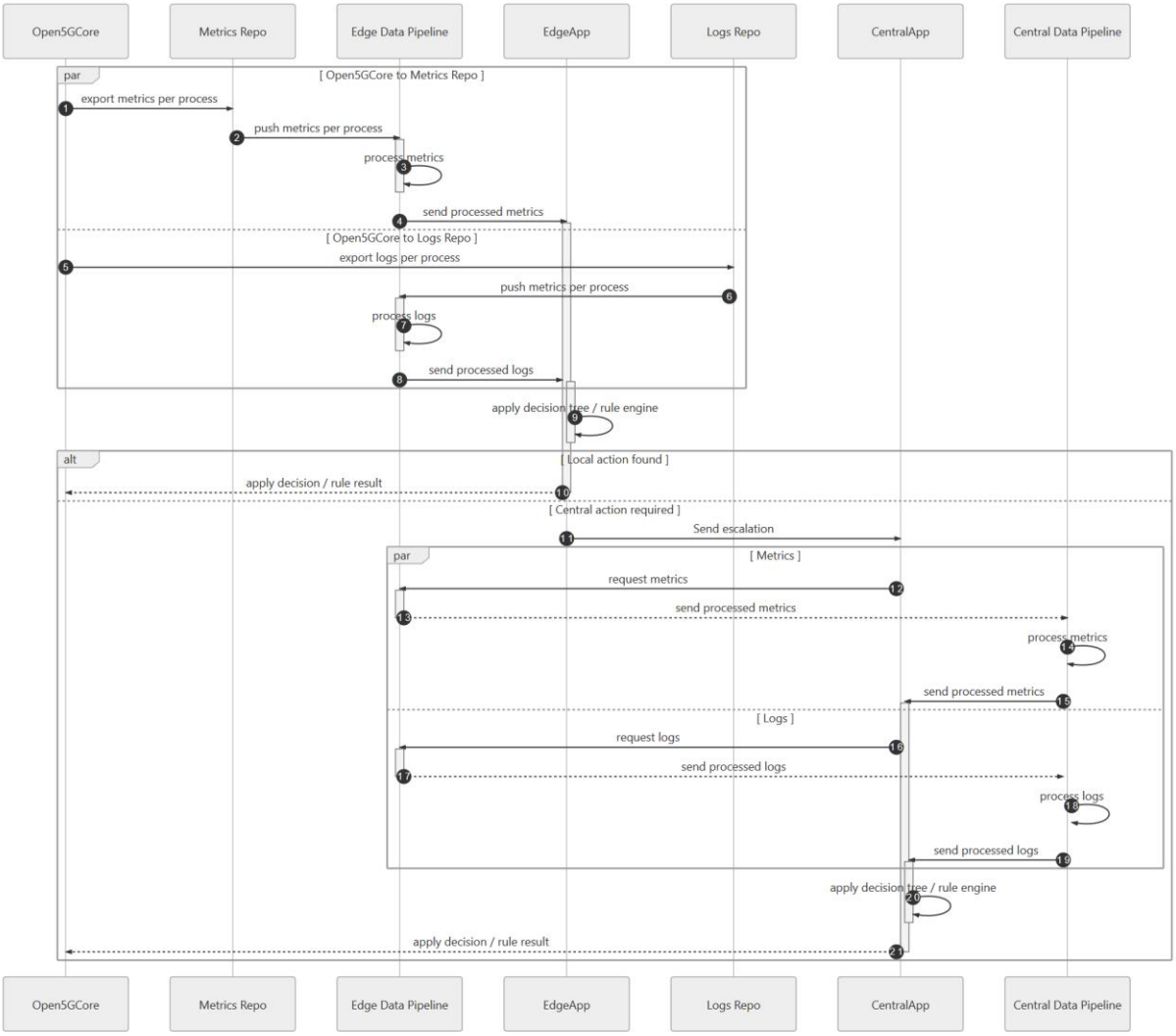


Figure 5-13: Edge Central Decision Control Loop

Figure 5-13, Illustrates the Edge Central Decision Control Loop. When no pertinent action can be decided at the edge, the edge-control app escalates the issue to the central control-app. The central AI-based control app can request for additional metrics and logs from the edge and provide a concrete, individual solution in form of an action (e.g., reconfiguration) for that edge. When multiple edges escalate similar issues to the central, the central AI-based control app would issue a system-wide policy that must be applied by every edge connected to the central. This second control loop is very useful for situations when insight is available only through a wider view from the central node or when a more dynamic decision has to be taken which requires ML algorithms.

Figure 5-14, depicts the control loop with Human interval. When no solution can be found by the central AI-based control app, the issue is escalated for human / administrator intervention. The administrator can inject specific policies for individual edges or provide a generic policy for all the edges. Through the Central Dashboard, the administrator / operator can view the status of individual edges and override any policies applied by the AI based control app in the central.

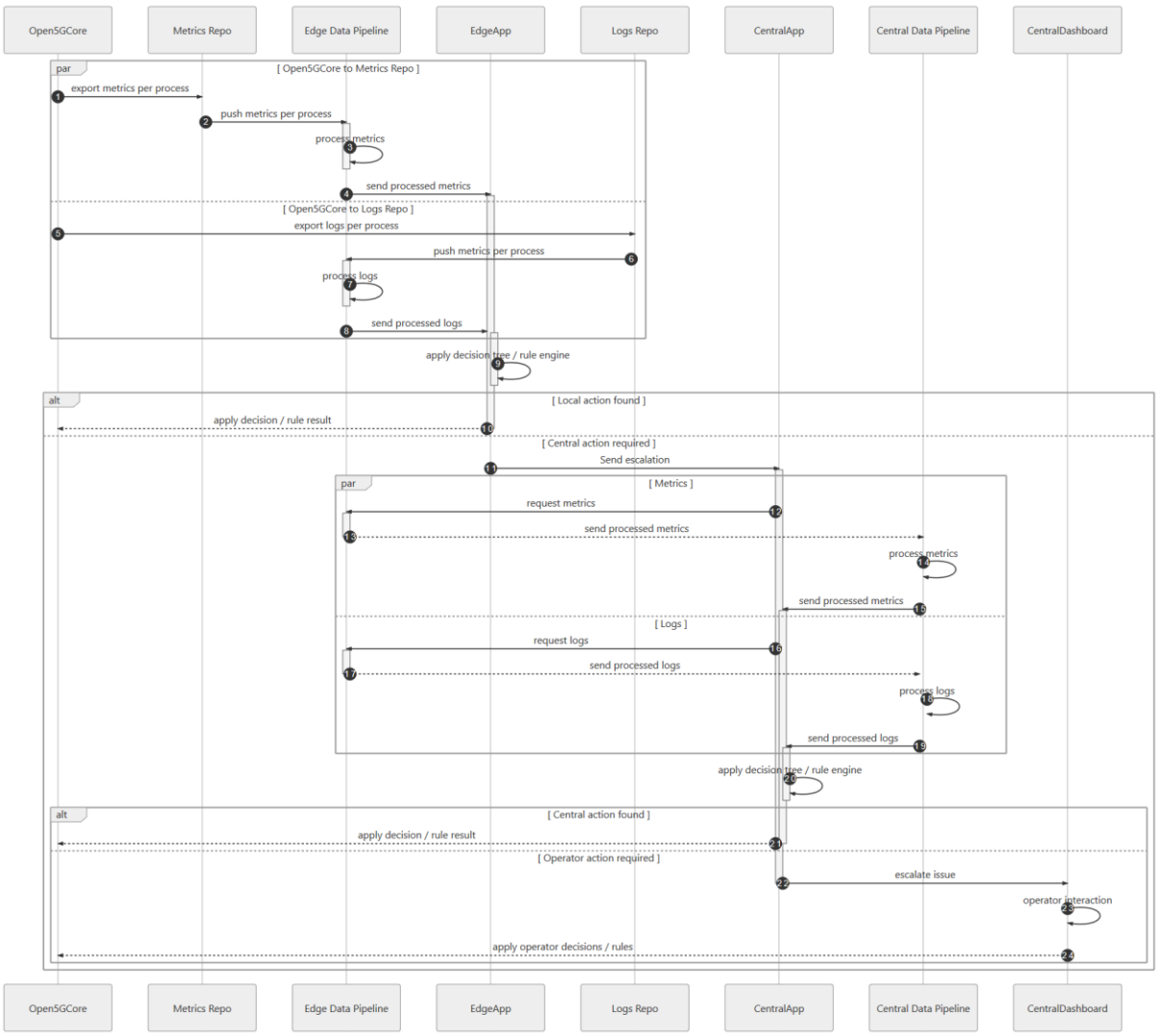


Figure 5-14: Edge Central Human Decision Control Loop

5.8 MEC-enabled NFV MANO

Multi-access Edge Computing (MEC) is increasingly adopted by mobile network operators as a means to bring enterprise applications closer to the user equipment (UE). The UE that is considered in our examined scenarios is linked to resource-constrained (e.g. processing, memory) IoT devices, capable of sensing/actuating autonomously without any human intervention. Additionally, edge computing technologies such as the Mobile Edge Platforms (MEPs), are located within UE's proximity (e.g. nearest base station).

We consider two main schemes for edge computing technologies:

- *Standalone scheme*: In this scheme the IoT device is equipped with a cellular interface in order to communicate with the nearest base station, where edge computing resources are located.
- *Gateway-assisted*: In this scheme the IoT device only has non-cellular wireless interfaces for data exchange. Hence, an additional network element (i.e. gateway) is added in order to provide cellular connectivity to the nearest base station.

In both schemes, edge computing technologies are specifying edge resources that are participating to end-to-end network slices, the latter being initiated on NFVI spanning across cloud/core and edge. To enable the integration of edge resources with the mobile core and cloud resources, we are

currently investigating two main options, where we put emphasis on automation of MEP-supported service layer interactions:

A. *Converged Cloud-to-Edge MANO*: This option uses the same NFV MANO orchestrator for both Cloud and edge resources. Hence, the MANO orchestrator is responsible for configuration and lifecycle management of the MEPs as well as the edge resources and additionally including them to network slices as appropriate. This option is illustrated in Figure 5-15.

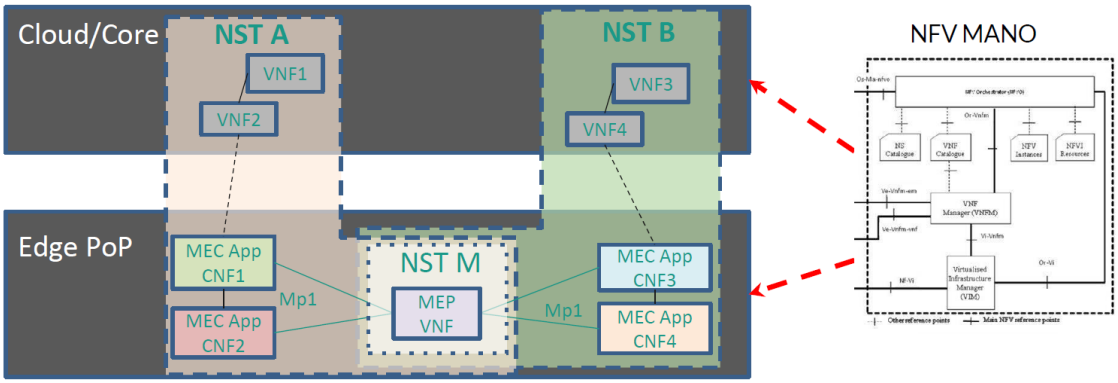


Figure 5-15: Converged Cloud-to-Edge MANO

This option has as a main benefit the unified view over Cloud/Core and edge resources. However, the drawbacks are in management complexity and scalability, as MANO has to control multiple Edge PoPs and their MEPs. Additionally, there is neither autonomy on the Edge PoP nor in the management of edge resources, as they are offered through a centralized control scheme.

B. *Cloud/Core MANO interoperable with the Edge Orchestration Platform*: This option is using a different orchestrator for each Edge PoP, however the edge is not fully autonomous. Specifically, the edge orchestration platform has to receive instructions for configuring the edge resources and including them to a network slice from the MANO orchestrator. To achieve this, Edge resources are exposed to MANO as platform services. This option is illustrated in Figure 5-16.

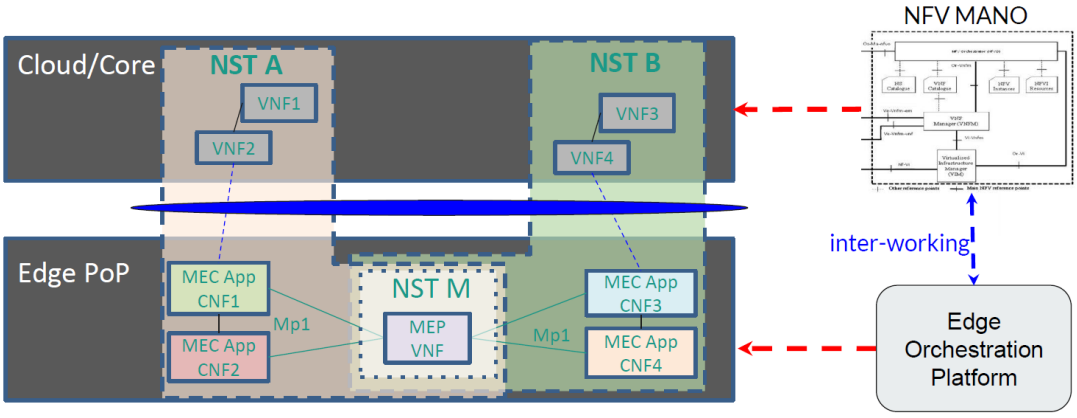


Figure 5-16: Cloud/Core MANO interoperable with the Edge Orchestration Platform

The benefits of this option are:

- High flexibility, efficiency and scalability, since the edge PoPs retain semi-autonomous operation to interoperate with multiple MANOs as well as the Edge infrastructure/service layer management is decoupled from Cloud/Core-level MANO.

- Independence from continuous edge and MANO interconnection, as each provider can use their own edge orchestration platform and connect/disconnect dynamically to the MANO based on demand.
- Mobile Network Operators can extend edge coverage by leasing existing edge (cloud/radio) resources

Nevertheless, the limitations of this option lie on the absence of standardized mechanisms or APIs to allow the interaction between the Edge Orchestration Platform in different Edge PoPs and the MANO. Additionally, an automated synchronization mechanism is required to update MANO when changes are made on the edge resources in different Edge PoPs.

As a next step for this work we are planning to compare/contrast prototype solutions of the options for the integration of MEC-enabled edge and Cloud orchestration. Specifically, in the forthcoming 5G-VINNI deliverables, we will investigate scenarios of these two options and we will elaborate on orchestration workflows to support MEC-enabled cloud/core to edge network slicing. Additionally, we will investigate the extensions that are required for the NFV MANO stack, in order to support MEC-enabled information models and integration of MANO with Edge Platforms.

5.9 Edge mMTC Slicing (ICOM)

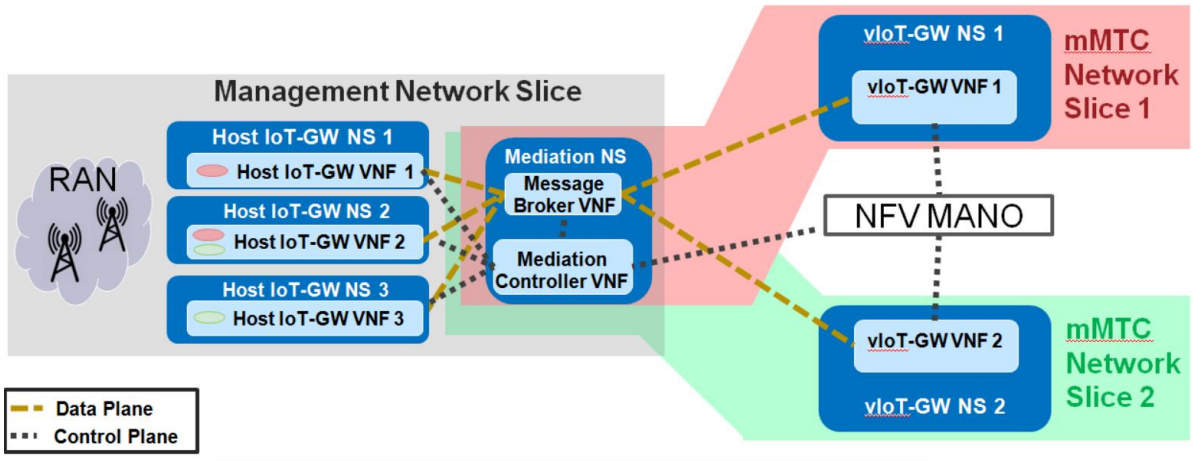


Figure 5-17: Edge mMTC Network Slicing NFV MANO artefacts

In this subsection, we present our design to support the orchestration and management of the Edge mMTC Network Slices, as introduced in [2]. This approach raises the necessity for exclusivity over IoT devices and allows multi-tenant edge processing over shared IoT infrastructure. Orchestration automation plays a vital role for the feasibility and practicality of Edge mMTC slicing and to this end the orchestration design needs to appropriately reflect and support envisioned functionalities and separation of concerns of different stakeholders.

The Edge mMTC Network Slicing architecture has a modular design with clear-cut interfacing for explicit management and orchestration of the lifecycle and configuration of mMTC Network Slices. To showcase this capability, we present the NFV constructs in context of the ETSI NFV MANO architectural framework. As illustrated in Figure 5-17, the end-to-end deployment of Edge mMTC slicing involves two distinct classes of Network Slices, which according to the NFV MANO framework are each described using a Network Slice Template (NST). The Management Network Slice consists of one or more Host IoT GW Network Services (NS) as slice subnets, each including in turn one Host IoT GW Virtual Network Function (VNF). In addition, the Management Network Slice consists of a Mediation NS, which is declared as a shared slice subnet, essentially translating to its inclusion by other Network Slices without the need for the corresponding services' re-instantiation. The Mediation NS includes a Message Broker VNF and a Mediation Controller VNF. The second class of

Network Slices, i.e., the mMTC Network Slice, consists of one vIoT NS including a vIoT VNF, as well as the Mediation NS shared with the Management Network Slice.

Multiple mMTC Network Slices share the same Mediation NS and the Data Plane interaction between vIoT NS and the Management Network slice, only takes place between vIoT GW VNF and the Message Broker VNF of the Mediation NS. Furthermore, control plane interactions from the Orchestrator (NFV MANO) to the Host IoT GW are indirectly propagated via the Mediation Controller VNF, raising the necessity of Host IoT GW being orchestrated by the NFV MANO and allowing for integration with external Host IoT device management platforms. Hence, this design enables various degrees of autonomicity of the Host IoT GWs, facilitating at the same time different orchestration schemes based on interoperability of different edge domains. In addition, the Management Network Slice preserves its administrative control over Host IoT GWs and Mediation VNFs, offering a unified view over all edge IoT resources to a single administrative entity, optimizing this way management efficiency. Finally, mMTC slices can be orchestrated in a dynamic fashion, with relatively isolated lifecycle management operations and with minimum overhead towards the IoT resource management.

6 Conclusion

This document contains design considerations, the enhancements of 5G Network Slicing and slice operation learned during the implementation of 5G services with Network slicing to support Verticals.

Chapter 2 has addressed an updated 5G-VINNI Network Slicing architecture of 5G-VINNI E2E facility with a customized network slice type defined by 5G-VINNI.

Chapter 3 contained 5G System architectural and functional considerations on 5G Core, RAN and Transport Network (TN). It is also discussed the architectural considerations of MEC solutions in 5G-VINNI Facility Sites, providing different edge deployment scenarios. To support vertical applications, requiring E2E Network Slice deployed across facility sites, the enhanced network slice federation design and mechanisms are provided in this release. In particular, how federation enables the deployment and operation of an E2E Network Slice instance across two facility sites upon vertical request under consideration of different federation options at service, network, or service and network levels. Network Slice federation actors, roles model for the different federation options are also given. To support the enhancement of 5G System for future releases, private networks integration in 5G-VINNI are discussed with potential enablers and a number of 5G-VINNI use cases which are candidate for PNI-NPN provisioning.

Chapter 4 has captured Network Slicing security and isolation mechanisms. The Security-as-a-Service defined by 5G-VINNI is discussed which allows to be applied to individual slices according to the requirements of some verticals. Security and isolation mechanisms are further provided by complementing the work done in the previous release [3].

Chapter 5 has highlighted the progress of 5G-VINNI research projects with the focus of the key enabling technologies and potential research directions for future mobile network architecture.

A wide range of slicing use cases to the vertical industries and their applications supported by 5G-VINNI E2E facility are provided in Annex A, along with an analysis of the gaps and an early assessment of the gap type of use cases currently being run at the 5G-VINNI facilities.

In summary, this document intends to serve as a baseline for upcoming development and implementation activities within the project, particularly in dealing with platform implementation for Network Slicing Operation during the final phase of project completion. In addition, this document aims to provide a reference point for vertical industries and vertical applications seeking to conduct Testing on 5G systems for Network Slicing by providing the development and implementation for Network Slicing operation. The key enabling technologies from various 5G-VINNI research projects are proposed as potential research directions for future mobile network architecture.

Annex A Network Slicing Use Cases and Gap Analysis in 5G-VINNI facility sites

A.1 Network Slicing Use Cases in 5G-VINNI

5G-VINNI will support a wide range of use cases to vertical industries and their applications, that wish to take advantage of the capabilities of the 5G systems offered by the various 5G-VINNI Facility Sites. These use cases will be supported by communication service providers (CSPs) who may offer Network Slice as a Service (NSaaS) as a means of delivering them.

The following sections outline the network slices offer at the 5G-VINNI facilities, categorised by slice type (eMBB, URLLC, mIoT).

A.1.1 5G-VINNI Use Cases with eMBB slice type

eMBB is a set of services characterized by their need for high bandwidth and throughput, typically in excess of 1Gbps. The following eMBB slice use cases will be offered by 5G-VINI facilities.

Table 6-1: eMBB slice use cases with 5G NSA scenario offered by 5G-VINNI facility sites.

	#	Use Case	Description	Facility Location
5G NSA	1	Fish farming	Providing the remote monitoring of water and fish quality	Norway
	2	Healthcare	Supporting remote ultrasound, paramedic support and management of critical health events	Norway
	3	Enhanced media	Providing live event experience across geographies	Norway
	4	AR/VR	Providing an immersive experience during robot control	UK
	5	Media broadcast with QoS	Providing low packet loss and jitter on upstream	UK
	6	Open air interface	Traffic and experiment testing	Spain
	7	Intercontinental slice	Scalable network slice to USA	Spain
	8	OPTARE edge solution	MEC-to-MEC assessment	Spain
	9	Ultra-high-fidelity media	Bitrate and QoE assessment	Greece
	10	User and machine-generated content	Bitrate and QoE assessment	Greece
	11	Co-operative media production	Bitrate and QoE assessment	Greece
	12	Subscriber mobile broadband	Basic connectivity for NSA	Berlin

Table 6-2: eMBB slice use cases with 5G SA scenario offered by 5G-VINNI facility sites.

	#	Use Case	Description	Facility Location
5G SA	1	AI recognition	Recognising objects from video analysed with AI	UK
	2	Neuro regeneration	Monitors brain activity and other vitals to assess patient health	Spain
	3	Ultra high-fidelity media	Bitrate and QoE assessment	Greece
	4	User and machine-generated content	Bitrate and QoE assessment	Greece
	5	Co-operative media production	Bitrate and QoE assessment	Greece
	6	Multi-CDN selection	Bitrate and QoE assessment	Greece
	7	CDN services	In dense, static and mobile environments	Greece
	8	360° immersive experience	Bitrate and QoE assessment	Greece
	9	On-site live event experience	Bitrate and QoE assessment	Greece
	10	High speed mobility	Heterogeneous technology access for on-board network connectivity in a railway setup	Greece
	11	Real-time video	Content transmission at level crossing	Portugal
	12	Smart metering	Video camera and augmented reality streaming	Portugal
	13	Local offload	Providing edge offload and centralized control	Berlin & Luxembourg
	14	Local control	Providing local control next to local offload	Berlin
	15	Autonomic edge	Includes all functionality needed for on premise 5G communication	Berlin & Luxembourg
	16	Mobile hospital	Data transfer between moving point-of-care and university hospital	Munich
	17	Embedded video conference	Streaming of video and audio from cameras in both hospital and remote ambulance	Munich

A.1.2 5G-VINNI Use Cases with URLLC slice type

URLLC is a set of services characterized by their need for low latency (typically less than 10ms) and highly reliable (typically greater than 99.999%) links. The following URLLC slice use cases will be offered by 5G-VINI facilities.

Table 6-3: URLLC slice use cases with 5G NSA scenario offered by 5G-VINNI facility sites.

	#	Use Case	Description	Facility Location
5G NSA	1	Defence core	Dedicated Slice with independence and isolation on all the CORE components.	Norway
	2	Defence edge	Fully autonomous Edge with all CORE functionalities	Norway
	3	Factory of the future	Rapid deployment, auto/re-configuration and testing of new robots	Norway
	4	Remote robotic control	Providing latency <10ms in the control path	UK
	5	Low latency content streaming	Providing guaranteed stable bit rate	UK
	6	OPTARE edge solution	MEC-to-MEC assessment	Spain

Table 6-4: URLLC slice use cases with 5G SA scenario offered by 5G-VINNI facility sites.

	#	Use Case	Description	Facility Location
5G SA	1	Factory of the future	Rapid deployment, auto/re-configuration and testing of new robots	Norway
	2	Tactile internet	Haptics for healthcare applications	UK
	3	Safety critical communications	From approaching train detectors to the level crossing controllers	Portugal
	4	Smart metering	Critical last-gasping features and enhanced synchronization	Portugal
	5	Industry 4.0	Providing basic TSN integration for deterministic communication	Berlin
	6	Mobile hospital	Remote control of sensor applications	Munich
	7	Streaming video	Remote control of fixed camera	Munich

A.1.3 5G-VINNI Use Cases with mIoT slice type

mIoT is a set of services characterized by their need to support a large set of, often, IoT devices, typically in excess of 1 million per square km. The following mIoT slice use cases will be offered by 5G-VINI facilities.

Table 6-5: mIoT slice use cases with 5G NSA scenario offered by 5G-VINNI facility sites.

	#	Use Case	Description	Facility Location
5G NSA	1	Healthcare	Pillcam for remote colonoscopy and vital-sign patches	Norway
	2	Smart city	Intelligent street lighting; smart parking	Norway
	3	HV/LV energy metering	Dynamically re-configurable ICT infrastructure to facilitate the smart energy operation	Greece
	4	Digital utilities	Fully automated Digital Utility Management system	Greece

Table 6-6: mIoT slice use cases with 5G SA scenario offered by 5G-VINNI facility sites.

	#	Use Case	Description	Facility Location
5G SA	1	Smart port	Autonomous assets & logistics	Norway
	2	Neuro regeneration	Monitors brain activity and other vitals to assess patient health	Spain
	3	IoT	Providing support for IoT communication	Berlin & Luxembourg

A.2 Gap Analysis and Early Assessments

The following table shows use cases currently being run at the 5G-VINNI facilities, in particular, Norway, UK, Spain, Greece, Berlin and Luxemburg. A description of each use case is provided along with an analysis of the gaps that exist in the network slice implementation. An early assessment of the gap type is also given.

Table 6-7: Gap Assessment of selected Network Slicing use cases offered by 5G-VINNI facility sites.

#	Use Case [facility, slice type, SA/NSA]	Use Case Description	Issue Type (S: Service, P: Platform)	Problem Description	Gap Assessment
1	Defence [Norway, URLLC, NSA]	Dedicated Slice with independence and isolation on all the CORE components.	1. S: SRTP 2. S: Prioritisation/ Differentiation	1. Limited support of SRTP in 5G handsets, which is critical for Defence requiring E2E encryption for communication services (e.g. Voice, video, PTT) 2. Flexible way of prioritizing traffic for verticals apps that is not using Rx interface (e.g. IMS)	1. Implementation gap 2. Standards gap

2	Health [Norway, eMBB & mMTC, NSA]	Remote Ultrasound, Paramedic Support and Manage of Critical Health Events, Pillcam for remote colonoscopy and Vital-Sign Patches (IoT)	1. S: Vital sign patches use case requires low power class feature	1. There is a need for lower power class in Rel14 for vital sign patches which allows devices to transmit messages with a maximum transmit power of 14 dBm (power class 6). This feature is not yet available both on RAN side and on device side.	1. Implementation gap
3	Robotics [UK, eMBB, NSA]	eMBB provides path for 360° camera streaming to VR headset	1. P: control and user plane	1. Providing different slices requires multiple instances of the 4G core network (one for each slice). System currently designed with only single core. The only way to support any form of slice in this release is by using different QoS profiles via separate APNs.	1. Implementation gap
4	OAI Testbed [Spain, eMBB, NSA]	Open Air Interface (OAI) TestBed deployment implemented as a hybrid (we use PNFs and VNFs) E2E NSA NS. Mainly used for traffic, radio and experiment testing.	1. P: User and control plane 2. S: Performance	1. gNB still in development branch, certain parameters make the deployment unstable and it does not support as many USRPs as the eNB. 2. Unstable signalling under certain circumstances and transmission errors around 1.9GHz and 6GHz with USRP B210-mini.	1. Implementation gap 2. Implementation error & Standard gap
5	User & machine generated content [Greece, eMBB, NSA]	Generation and simulation of various subscriber profiles in terms of bitrate and quality requirements and the evaluation of the corresponding QoE.	1. P: Mgmt plane 2. S: Throughput	1. A single slice is offered. We cannot have different profiles from the 5G System. The bitrate and quality are controlled from the video quality. Maybe on SA we could have advanced scenario 2. We cannot guarantee the throughput. We tested only with a single UE currently (the CPE) and a mobile phone so we don't know how the use case is affected with multiple UEs	1. Standard gap 2. Standard gap
6	Subscriber mobile broad-band [Berlin, eMBB, NSA]	Basic connectivity for NSA	1. P: Control plane	1. Interoperability between base stations and core network offering 5G NR SA connectivity	1. Standard gap
7	Local commun-	Providing edge offload and	1. P: Control and data plane split	1,2. Still limited throughput	1. Standard gap 2. Implementation

	ications support [Berlin & Lux., eMBB, SA]	centralized control	2. S: Performance		gap/ interoperability gap
8	Support for auton-omous edge nodes [Berlin & Lux., eMBB, SA]	Includes all functionality needed for on-premise 5G communications	1. P: Control, data and mgmt plane 2. S: Performance	1,2. Still limited throughput	1. Standard gap 2. Implementation gap/ interoperability gap
9	mIOT [Berlin & Lux., mMTC, SA]	Providing support for IoT communication	1. P: Control, data and mgmt plane	1. Limited availability of base stations and devices	1. Third party Implementation gap

References

[1] "5G-VINNI deliverable D1.1 - Design of infrastructure architecture and subsystems v1".

[2] "5G-VINNI deliverable D1.4: Design of infrastructure architecture and subsystems v2".

[3] 5GVINNI, "Deliverable 1.2 - Design of network slicing and supporting systems v1," 2019.

[4] GSMA, "Generic Network Slice Template, Version 2.0," Oct 2019.

[5] Foukas, X., Marina, M. K., & Kontovasilis, K. , "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture.," in *In Proceedings of the 23rd annual international conference on mobile computing and networking*, 2017, October .

[6] Foukas, X., Nikaein, N., Kassem, M. M., Marina, M. K., & Kontovasilis, K., "FlexRAN: A flexible and programmable platform for software-defined radio access networks.," in *In Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, 2016, December.

[7] Nikaein, N., Chang, C. Y., & Alexandris, K. , "Mosaic5G: Agile and flexible service platforms for 5G research.," in *ACM SIGCOMM Computer Communication Review*, 2018.

[8] "https://www.openairinterface.org/," [Online].

[9] 3GPP TS23.501, "System Architecture for the 5G System (5GS); Stage 2; Release 16".

[10] "3GPP TS 23.502: Procedures for the 5G System (5GS);Stage 2 (Release 16)".

[11] "IETF Definition of Transport Slice draft-nsdt-teas-transport-slice-definition-03," July 12, 2020.

[12] "Categories and Service Levels of Network Slicing, White Paper, 5G SLICING ASSOCIATION," March 2020.

[13] "3GPP TS2 28.530 V16.2.0 (2020-07) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Management and orchestration; Concepts, use cases and requirements (Release 16)".

[14] "5G-VINNI deliverable D3.1, Specification of services delivered by each of the 5G-VINNI facilities".

[15] "Is microwave transport ready for 5G? The Ericsson Blog," Feb 11, 2020.

[16] "Class of Service User Guide (Routers and EX9200 Switches), Juniper Networks," June 25, 2020.

[17] "5G-VINNI deliverable D1.3, Design for systems and interfaces for slice operation v1".

[18] "https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf," [Online].

[19] "https://www.etsi.org/deliver/etsi_gr/MEC/001_099/017/01.01.01_60/gr_MEC017v010101p.pd

] f," [Online].

[20 "https://www.etsi.org/deliver/etsi_gr/MEC/001_099/017/01.01.01_60/gr_MEC017v010101p.pd
] f," [Online].

[21 "https://www.mef.net/resources/technical-specifications/download?id=44&fileid=file1,"
] [Online].

[22 "5G-VINNI deliverable D5.1: Ecosystem analysis and specification of B&E KPIs".
]

[23 5G Alliance for Connected Industries and Automation (5G-ACIA), "White Paper: 5G Non-Public
] Networks for Industrial Scenarios," July 2019.

[24 "3GPP TS 28.541, Management and orchestration; 5G Network Resource Model (NRM); Stage 2
] and 3," 2018.

[25 "https://www.5gsolutionsproject.eu/," [Online].
]

[26 "RESISTO Project, RESIlience enhancement and risk control platform for communication
] infraSTructure Operators,".

[27 "3GPP TS33.855, Study on security aspects of the 5G Service Based Architecture (SBA)".
]

[28 "https://www.ses.com/networks/networks-and-platforms/o3b-mpower," [Online].
]

[29 "3GPP TR 22.822: Study on using satellite access in 5G".
]

[30 "3GPP TR 23.737, Study on architecture aspects for using satellite access in 5G".
]

[31 "3GPP TR 28.808, Study on management and orchestration aspects of integrated satellite
] components in a 5G network".

[32 "ETSI TR 103 611, Integration of satellite and/or HAPS (High Altitude Platform Station) systems
] into 5G and related architecture options".

[33 "https://artes.esa.int/projects/cloudsat," [Online].
]

[34 "https://artes.esa.int/projects/instinct," [Online].
]

[35 "https://satis5.eurescom.eu," [Online].
]

[36 "http://www.ict-vital.eu," [Online].
]

[37 "http://sat5g-project.eu/," [Online].
]

[38 "https://5genesis.eu/," [Online].

]

[39] W. Y. Poe, “Provisioning Private 5G networks by means of Network Slicing: Architectures and Challenges,” 2020. [Online].

[40] M. X. e. al., “Towards closed loop 5G service assurance architecture for network slices as a service,” in *EuCNC*, 2019.

[41] M. X. e. al., “AI-driven closed-loop service assurance with service exposures,” in *EuCNC*, 2020.
]

[42] M. Bloecher, R. Khalili, L. Wang, and P. Eugster, “Letting off STEAM: Distributed Runtime Traffic Scheduling for Service Function Chaining,” in *IEEE INFOCOM* , 2020.

[43] J. G. Dai and W. Lin, , “Asymptotic optimality of maximum pressure policies in stochastic processing networks,” The Annals of Applied Probability, vol. 18, no. 6, pp. 2239–2299, 2008.,” *The Annals of Applied Probability,* Vols. vol. 18, no. 6, p. pp. 2239–2299, 2008.

[44] “Intel, “Data plane development kit,” https://www.dpdk.org.”.
]

[45] A. Satyam, M. Francesco, C. F. Chiasserini, and D. Swedes, “Joint VNF Placement and CPU Allocation in 5G,” in *IEEE INFOCOM*, 2018.

[46] X. Fei, F. Liu, H. Xu, and H. Jin, “Adaptive VNF Scaling and Flow Routing with Proactive Demand Prediction,” in *IEEE INFOCOM*, 2018.

[47] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, ““Deploying chains of virtual network functions: On the relation between link and server usage,” in *IEEE INFOCOM*, 2016.

[48] Q. Zhang, F. Liu, and C. Zeng, “Adaptive interference-aware VNF placement for service-customized 5g network slices,” in *IEEE INFOCOM*, 2019.

[49] “Cogent Comms., “Cogent Network Map,”,” [Online]. Available: http://cogentco.com/en/network/network-map.

[50] W. Fischer and K. S. Meier-Hellstern, “The Markov-Modulated Poisson Process Cookbook,” *Perform. Eval,* Vols. vol. 18, no. 2, p. pp. 149–171, 1993.

[51] P. Quinn, U. Elzur, and C. Pignataro, “Network Service Header (NSH),” RFC 8300,” 2018.
]

[52] “https://www.elastic.co/,” [Online].
]

[53] “https://grafana.com/,” [Online].
]

[54] “https://www.elastic.co/kibana,” [Online].
]

[55] “https://www.influxdata.com/,” [Online].
]

[56] “https://nifi.apache.org/,” [Online].
]

[57 “https://zookeeper.apache.org/,” [Online].
]

[58 “https://docs.drools.org/5.6.0.Final/drools-fusion-docs/html/,” [Online].
]