H2020 5G-Transformer Project
Grant No. 761536

Definition of service orchestration and federation algorithms, service monitoring algorithms

Abstract

This deliverable describes the initial design of the service orchestrator of 5G-TRANSFORMER, including the specification of the northbound interface with the vertical slicer, southbound interface with the mobile transport and computing platform, and east/westbound interface with federated service orchestrators. This deliverable also describes the algorithmic framework to carry out service orchestration and federation decisions, including resource allocation or function placement decisions.

Document properties

Document number	D4.1
Document title	Definition of service orchestration and federation algorithms, service monitoring algorithms
Document responsible	Andres Garcia-Saavedra (NEC)
Document editor	Andres Garcia-Saavedra (NEC)
Editorial team	Andres Garcia-Saavedra (NEC), Juan Brenes Baranzano (ATOS), Carla Chiasserini (POLITO), Giada Landi (NXW), Dmitriy Andrushko (Mirantis), Jaime Garcia (UC3M), Kiril Antevski (UC3M), Ricardo Martinez (CTTC), Barbara Martini (SSSA), Adlen Ksentini (EURECOM)
Target dissemination level	Public
Status of the document	Final
Version	1.0

Production properties

Reviewers	Andres Garcia-Saavedra (NEC), Juan Brenes Baranzano (ATOS), Carla Chiasserini (POLITO), Giada Landi (NXW), Dmitriy Andrushko (Mirantis), Jaime Garcia (UC3M), Kiril Antevski (UC3M), Ricardo Martinez (CTTC), Barbara Martini (SSSA), Adlen Ksentini (EURECOM), Thomas Deiß (NOK-N), Carlos J. Bernardos (UC3M)

Disclaimer

This document has been produced in the context of the 5G-TRANSFORMER Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement Nº H2020-761536.

All information in this document is provided “as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

Table of Contents

List of Contributors

Partner Short Name	Contributors
UC3M	Kiril Antevski, Carlos Jesús Bernardos Cano, Antonio Pastor
NEC	Andres Garcia-Saavedra, Xi Li, J. Xavier
ATOS	Arturo Zurita, Juan Brenes, José Enrique Gonzalez
TID	Luis Miguel Contreras
ORANGE	Thouraya Toukabri
NXW	Giada Landi, Marco Capitani, Francesca Moscatelli
MIRANTIS	Dmitriy Andrushko, Konstantin Tomakh
CTTC	Ricardo Martínez, Iñaki Pascual, Jordi Baranda, Javier Vílchez, Manuel Requena, Josep Mangues
POLITO	Carla Fabiana Chiasserini, Francesco Malandrino
EURECOM	Adlen Ksentini, Pantelis Frangoudis
SSSA	Luca Valcarenghi, Barbara Martini
ITRI	Chia-Lin Lai

List of Figures

List of Tables

List of Acronyms

Acronym	Description
5GC	5G Core
5GT-MTP	Mobile Transport and Computing Platform
5GT-SO	Service Orchestrator
5GT-VS	Vertical Slicer
AAA	Authentication, Authorization, Accounting
AN	Access Network
API	Application Programming Interface
AppD	Application Descriptor
BSS	Business Support System
CAT	Catalogue
CN	Core Network
CSMF	Communication Service Management Function
DB	Database
DF	Deployment Flavor
E2E	End to end
EBI	Eastbound Interface
EM	Element Management
EPC	Evolved Packet Core
EPCaaS	EPC as a Service
ETSI	European Telecommunication Standardization Institute
GRE	Generic Routing Encapsulation
GS	Group Specification
HSS	Home Subscriber Server
IaaS	Infrastructure as a Service
IFA	Interfaces and Architecture
KPI	Key Performance Indicator
LC	Lifecycle
LCid	Lifecycle Operation Occurance Id
LCM	Lifecycle Management
M&E	Media and Entertainment
M(V)NO	Mobile (Virtual) Network Operator
MANO	Management and Orchestration
MEC	Multi-access Edge Computing
MEO	Multi-access Edge Orchestrator
MEP	Multi-access Edge Platform
MILP	Mixed Integer-Linear Programming
MIoT	Massive Internet of Things
MLPOC	Multiple Logical Point of Contact
MME	Mobility Management Entity
MNO	Mobile Network Operator
MON	Monitoring
MVNE	Mobile Network Enabler
NaaS	Network as a Service
NBI	Northbound Interface
NF	Network Function
NFP	Network Forwarding Path
NFV	Network Function Virtualization
NFVI	Network Functions Virtualisation Infrastructure

NFVIaaS	NFVI as a Service
NFV-NS	NFV Network Service
NFV-NSaaS	Network Service as a Service
NFV-NSO	Network Service Orchestrator
NFVO	NFV Orchestrator
NFVO-RO	Resource Orchestrator
NS	Network Slice
NSD	Network Service Descriptor
NS-DF	Network Service Deployment Flavor
NSI	Network Slice Instance
NSMF	Network Slice Management Function
NS-OE	NFV-NS Orchestration Engine
NSSI	Network Slice Subnet Instance
NSSMF	Network Slice Subnet Management Function
NST	Network Slice Template
ONAP	Open Network Automation Platform
OSM	Open Source MANO
OSS	Operating Support System
PM	Performance Management
PMON	Performance Monitoring
PNF	Physical Network Function
PNFD	PNF Descriptor
PoP	Point of Presence
QoS	Quality of Service
RAM	Resource Advertisement Management
RAN	Radio Access Network
REST	Representational State Transfer
RM	Resource Management
RMM	Resource Monitoring Management
RO	Resource Orchestration
RO-EE	RO Execution Entity
RO-OE	RO Orchestration Engine
SAP	Service Access Point
SBI	Southbound Interface
SDK	Software Development Kit
SDO	Standard Developing Organisation
SLA	Service Level Agreement
SLPOC	Single Logical Point of Contact
SLPOC-F	Single Logical Point of Contact for Federation
SO	Service Orchestrator
SPGW-C	Serving/Packet Data Network Gateway Control Plane
SPGW-U	Serving/Packet Data Network Gateway User Plane
TD	Technology Domain
TETRA	Terrestrial Trunked Radio
TN	Transport Network
TOSCA	Topology and Orchestration Specification for Cloud Applications
TSP	5G-TRANSFORMER Service Provider
UC	Use Case
UE	User Equipment
UPF	User Plane Function
VA	Virtual Application

vEPC	virtual Evolved Packet Core
VIM	Virtual Infrastructure Manager
VL	Virtual Link
VLAN	Virtual Local Area Network
VM	Virtual Machine
VNF	Virtual Network Function
VNFD	VNF Descriptor
VNFFG	VNF Forwarding Graph
VNFFGD	VNFFG Descriptor
VSD	Vertical Service Descriptor
VSI	Vertical Service Instance
WBI	Westbound Interface
WIM	Wide area network Infrastructure Manager
YAML	YAML Ain't Markup Language

Executive Summary and Key Contributions

This deliverable introduces the 5G-TRANSFORMER service orchestrator (5GT-SO), a key component of the 5G-TRANSFORMER system.

On the one hand, the 5GT-SO offers the vertical slicer (5GT-VS) (and ultimately the different verticals accessing the system) services and/or resources across single administrative domains or federated administrative domains. On the other hand, the 5GT-SO exploits either (i) a mobile transport and computing platform (5GT-MTP) in order to deploy services and expose abstracted (virtual) resources, or (ii) federated service orchestrator(s).

Consequently, the 5GT-SO manages services/resources that may be split across different administrative domains based on requirements, service/resource availability, and/or business reasons. Evidently, federation requires the peer 5GT-SO(s) to provide the abstractions of resources from their underlying 5GT-MTP, which may depend on business relationships between the federated domains.

To summarize, the main contributions in this deliverable are the following:

- An overview of the **overall 5GT-TRANSFORMER system design** is introduced in Section 2. The section briefly presents the three main functional blocks of our system (i.e., 5GT-VS, 5GT-SO and 5GT-MTP). The details about the functional blocks presented in this section have been published in [1].
- **Design of monitoring architecture and algorithms, and collection of monitoring requirements from all vertical use cases**, as a contribution of Task 4.2. In the 5G-TRANSFORMER framework, each of these architectural components includes a monitoring service able to provide performance metrics and failure reports targeting the logical entities managed by each component. Following this approach, the 5GT-MTP monitoring service will produce monitoring data about the local physical and virtual resources, the 5GT-SO monitoring service will produce monitoring data about the managed VNFs and NFV network services, while the 5GT-VS monitoring service will produce monitoring data about network slices and vertical services. Section 2.4 introduces the proposed monitoring architecture for the overall system. A detailed design of the monitoring platform and algorithms in 5GT-SO is explained in Section 4.7. The analysis of monitoring requirements of the vertical use cases are provided in Section 0 (Annex II).
- One of the objectives of 5G-TRANSFORMER is to foster business relationships between vertical sectors which request the deployment of network services via the 5GT-VS, and between service orchestrators of different administrative domains. This results in **a set of functional and business requirements** on the 5GT-SO that are enumerated in Section 3. These include business requirements of 5GT-VS, 5GT-MTP and federated 5GT-SOs, and functional requirements in different stages of a service lifecycle (discovery, fulfilment, assurance and service decommissioning).
- Section 4 introduces the **design of the 5GT-SO architecture**. Such design includes the different functional blocks comprising the service orchestrator, the northbound interface towards the 5GT-VS, the southbound interface towards the 5GT-MTP, and the eastbound/westbound interface towards federated

service orchestrators. Major components of the 5GT-SO are (i) repositories to store Network Service Descriptors (NSDs) and VNF Descriptors (VNFDs) that can be accessed via a Catalogue Manager, abstract resource elements from 5GT-MTPs, and instances of VNFs and network services; (ii) an NFV Orchestrator (NFVO) in charge of both resource orchestration (NFVO-RO) and service orchestration (NFVO-NSO) across multiple domains; (iii) VNF Managers (VFNMs) in charge of the lifecycle management of deployed VNFs; and (iv) a monitoring platform comprised of a monitoring service, a service monitoring data consumer and a Service-Level Agreement (SLA) manager.

- A series of **workflow descriptions among the components of the 5GT-SO**, illustrating a set of most important use cases, which are shown in Section 5. The relevant use cases presented in this document include (1) Service On-boarding, (2) Service Instantiation, (3) Service Modification, (4) Service Termination, (5) Service Assurance, (6) Service Federation, (7) Resource Federation, and (8) NFV-NS instantiation when including MEC applications. These workflow illustrate the interaction between the different components of 5GT-SO internally, and between these and 5GT-VS, 5GT-MTP or external 5GT-SOs.
- The **framework employed by 5GT-SO to deploy algorithms** that optimize the job of service orchestration and federation, introduced in Section 6. This framework will host the algorithms developed along with the project and represent one of the core sources of intelligence in 5G-TRANSFORMER, allowing optimizing actions based on different metrics (e.g., delay, energy, etc.) and supporting resiliency, fault-tolerance and flexibility.
- Dissemination of the results in WP4 and steering related standardization activities (activities collected by WP6), are of paramount interest within this work package and has resulted in a number of c**ontributions to academic journals and** conferences [1][2][3] and **standardization bodies** [4][5][6][7]. Reference [2] introduces the main concept of 5GT-SO and its relation to ETSI NFV, while reference [3] provides the detailed design of the 5GT-SO functional system architecture including the main function modules and the interfaces to 5GT-VS and other federated 5GT-SO(s).

1 Introduction

5G networks are envisioned to expand the service scope of current mobile networks to support various vertical services, hence enriching the telecom network ecosystem. A wide range of vertical industries (such as eHealth, automotive, media, or cloud robotics) acts as a driver to construct this ecosystem. The support of the diverse and heterogeneous service requirements of different vertical industries is not only a question of providing broadband capacity, but also a matter of ultra-reliable low-latency communications and a massive density of connections when needed. Consequently, 5G-TRANSFORMER proposes a flexible SDN/NFV-based design of the next generation mobile transport networks.

One of the most important building blocks of 5G-TRANSFORMER platform is the service orchestrator (5GT-SO). The 5GT-SO receives requests from the vertical slicer (5GT-VS) which translates the services requested by different verticals and mobile (virtual) network operators (M(V)NOs). The 5GT-SO offers service or resource orchestration and federation across multiple administrative domains. This includes all tasks related with coordinating and offering to the vertical an integrated view of services and resources from multiple administrative domains. Orchestration entails managing end-to-end services or resources that were split into various domains based on requirements and availability. Federation entails managing administrative relations at the interface between 5GT-SOs belonging to different domains and handling abstraction of services and resources.

The main function of the 5GT-SO is based on the NFV Orchestrator (NFVO) as defined in ETSI NFV. Depending on the requests from verticals, both network service (NFVO-NSO) and resource (NFVO-RO) orchestration may be used for both single and multiple domains. The NFV-NSO functionality of the 5GT-SO is responsible of deploying and managing the NFV network services (NFV-NS) requested by 5GT-VS, possibly across multiple domains, and it is also responsible for the network service lifecycle management including operations such as service on-boarding, instantiation, scaling, termination, and management of the VNF forwarding graphs associated to the network services. More specifically, the NFVO-NSO coordinates all NFV-NS deployment operations including authentication, authorization and accounting (AAA) and formal checks of service requests based on attributes retrieved from NSDs and VNFDs. It decomposes the end-to-end network services into several segments and decides to implement them either in the local administrative domain or federating with the neighbor 5GT-SOs. Afterwards, the NFVO-NSO requests the NFVO-RO of the local domain or the NFVO-NSO of a neighbor domain to deploy the Network Service segment.

In turn, the NFVO-RO functionality of the 5GT-SO handles the resources coming from the local 5GT-MTP (either physical or virtual) and from the service orchestrators belonging to other administrative domains (abstracted). The resource orchestration decision includes the placement of VNFs/VAs over such virtual networks with virtual nodes and links, as well as the resources to be allocated. The 5GT-SO will then create the service by exploiting the interface exposed by the local 5GT-MTP and also that exposed by peer orchestrators, which will eventually interact with their respective 5GT-MTPs.

The remaining of this document goes as follows. Section 2 introduces an overview of the overall 5G-TRANSFORMER platform architecture. An analysis of the business and functional requirements for the design of 5GT-SO is presented in Section 3. The actual design of 5GT-SO (including its functional components and interfaces to external 5G-TRANSFORMER blocks) is then presented in Section 4. The interaction between such external and internal 5GT-SO components is illustrated through a series of workflows implementing some common use cases in Section 5. The framework of the 5GT-SO to deploy optimization algorithms for service orchestration and federation is introduced in Section 6. Finally, Section 7 ends the main document with concluding remarks.

In order to keep the main body of the document as short as possible, several annexes are included at the end, containing additional information and results.

2 5G-TRANSFORMER System Overview

To describe the 5GT-SO within its context, we present in this section a summary[1] of the system architecture described in [10]. The use cases used to define the system architecture are presented in an annex in Section 9, requirements for the design of 5G-TRANSFORMER monitoring platform in Section 0, and relevant reference architectures for the 5G-TRANSFORMER system architecture in Section 11.

The 5G-TRANSFORMER project explores how the network can better serve the needs of 5G-TRANSFORMER customers (i.e., vertical industries and M(V)NOs) by offering the abstraction, flexibility, and dynamic management capabilities they require. In terms of notation, it is important to differentiate between (vertical) service, i.e., that is requested by the customer of the 5G-TRANSFORMER system, from the underlying network service deployed to fulfill the requirements of the vertical. An example of the former is a car manufacturer requesting the deployment of an automotive intersection collision avoidance service. The latter will be deployed in the form of an NFV network service, in general.

The key architectural concept to support such adaptation to the needs of verticals and M(V)NOs is network slicing. The term network slice aligns network functionality to business needs [34], since it allows customers to request not just functions, but also business objectives (e.g., quality of service, security, etc.), as a sort of intent. The scope of a slice may be a single customer facing service (using TM Forum terminology [35]) or a group of such services. The system will also allow infrastructure providers to share the 5G mobile transport and computing infrastructure efficiently among verticals and M(V)NOs, hence enhancing 5G-TRANSFORMER provider network usage efficiency. In terms of deployment, network slices can be implemented by means of ETSI NFV network services.

The architecture is conceived to support multiple combinations of stakeholders by introducing open Application Programming Interfaces (API) among components [9]. Through these APIs, the system hides unnecessary details from the verticals, allowing them to focus on the definition of the services and the required Service Level Agreements (SLAs). As for interfaces, particularly relevant for the goals of the project are east-westbound interfaces, which enable service and resource federation across different administrative domains, allowing 5G-TRANSFORMER service providers to enhance their service offerings to their customers by peering with other providers.

We envision a system of three major components: vertical slicer (5GT-VS), service orchestrator (5GT-SO) and mobile transport and computing platform (5GT-MTP), see Figure 1. The 5GT-VS is the entry point for the vertical requesting a service and it handles the association of these services with slices as well as network slice management. The 5GT-SO is responsible for end-to-end orchestration of services across multiple domains and for aggregating local and federated (i.e., from peer domains) resources and services and exposing them to the 5GT-VS in a unified way. Finally, the 5GT-MTP provides and manages the virtual and physical IT and network resources on which service components are eventually deployed. It also decides on the abstraction level offered to the 5GT-SO.

[1] This is text common to [11], [12], and this document.

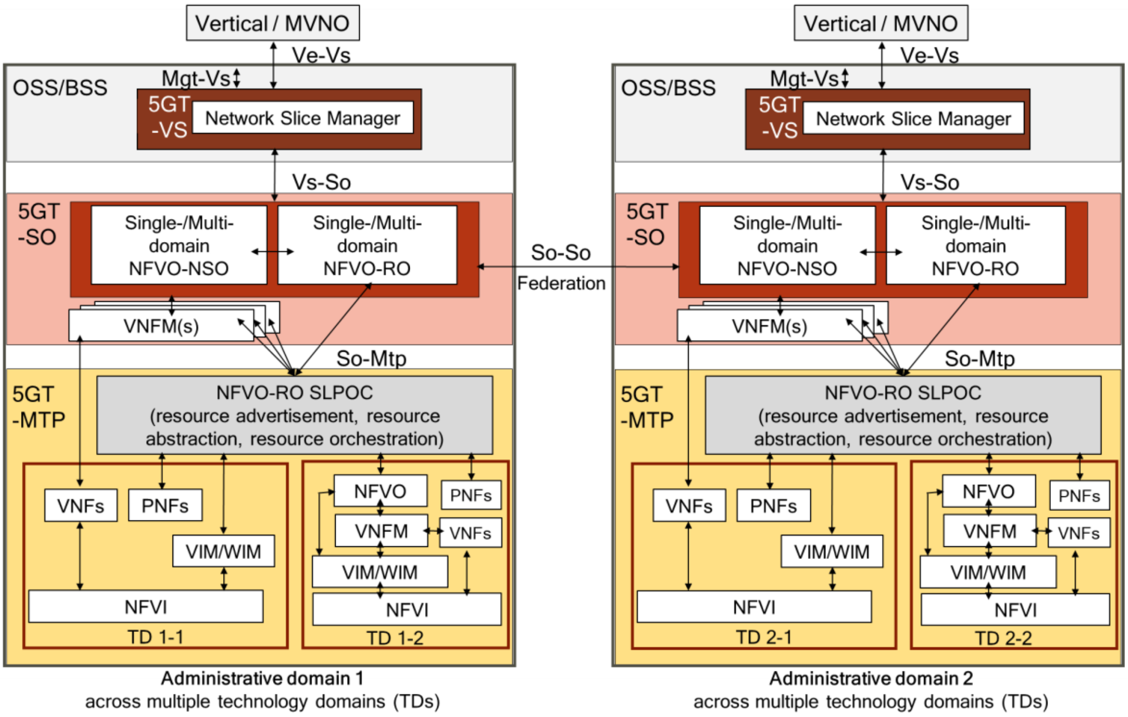

Vertical / MVNO
Ve-Vs
OSS/BSS
Mgt-Vs
5GT -VS
Network Slice Manager
Vs-So
5GT -SO
Single-/Multi-domain NFVO-NSO
Single-/Multi-domain NFVO-RO
VNFM(s)
So-Mtp
5GT -MTP
NFVO-RO SLPOC (resource advertisement, resource abstraction, resource orchestration)
VNFs
PNFs
VIM/WIM
NFVI
TD 1-1
NFVO
VNFM
TD 1-2
So-So Federation
TD 2-1
TD 2-2
Administrative domain 1 across multiple technology domains (TDs)
Administrative domain 2 across multiple technology domains (TDs)

FIGURE 1: 5G-TRANSFORMER SYSTEM ARCHITECTURE

2.1 Vertical Slicer (5GT-VS)

The 5GT-VS is the common entry point for all verticals into the 5G-TRANSFORMER system. It is part of the operating and business support systems (OSS/BSS) of the 5G-TRANSFORMER service provider (TSP) [9]. Vertical services are offered through a high-level interface at the 5GT-VS northbound that is designed to allow verticals to focus on the service logic and requirements, without caring on how they are eventually deployed at the resource level. This latter issue would be up to TSP. Therefore, vertical services, will use those services offered by the TSP. In fact, the 5GT-VS offers a catalogue of vertical service blueprints, based on which the vertical service requests are generated by the vertical. The role of the 5GT-VS is to trigger the actions allowing the 5G-TRANSFORMER system to fulfil the requirements of a given incoming service request. After the appropriate translation between service requirements and slice-related requirements by the VSD/NSD Translator and Arbitrator, corresponding to the Communication Service Management Function (CSMF), as defined in [36], a decision is taken on whether the service is included in an already existing slice or a new one is created.

The vertical slicer is the component of the system that is conscious of the business needs of the vertical, their SLA requirements, and how they are satisfied by mapping them to given slices. Intra-vertical arbitration is also part of the vertical slicer, by which intra-vertical contention is resolved to prioritize those services that are more critical, according to the agreed SLA.

The VSI/NSI Coordinator and LC Manager is the core component of the 5GT-VS. It contains functionality that can be mapped to that of the Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF), as defined in [36]. More specifically, the NSMF is in charge of lifecycle management of network slice instances. All possible combinations between vertical services and network slices exist; that is, a network slice can be shared by different vertical services, but a given vertical service may be mapped to multiple network slices as well. In turn, network slices may be composed by network slice subnets, which may offer part of the functionality required by a given network slice. And network slice subnets may be shared by multiple network slices.

The final result of all this process is a request sent by the 5GT-VS towards the 5GT-SO to create or update the NFV network services (NFV-NS) that implement the slices.

In summary, through this process, the 5GT-VS maps vertical service descriptions and instantiation parameters at the vertical application (VA) level into an NFV-NS (existing or new) implementing the network slice. In turn, such NFV-NS will be updated or created through a network service descriptor (NSD), which is a service graph composed of a set of virtual network functions (VNF) chained with each other, and the corresponding fine-grained instantiation parameters (e.g., deployment flavor) that are sent to the 5GT-SO. Given the operations carried out through it, the VS-SO interface (see Figure 1) takes ETSI GS NFV-IFA 013 [23] as basis.

2.2 Service Orchestrator (5GT-SO)

The NFV-NS from the 5GT-VS is received by the 5GT-SO through the VS-SO interface. The main duty of the 5GT-SO [41] is to provide end-to-end orchestration of the NFV-NS across multiple administrative domains by interacting with the local 5GT-MTP (So-Mtp reference point) and with the 5GT-SOs of other administrative domains (So-So reference point). If needed (e.g., not enough local resources), the 5GT-SO interacts with 5GT-SOs of other administrative domains (federation) to take decisions on the end-to-end (de)composition of virtual services and their most suitable execution environment. Even if a service is deployed across several administrative domains, e.g., if roaming is required, a vertical still uses one 5GT-VS to access the system, and so, the 5GT-SO hides this federation from the 5GT-VS, and thus, the verticals.

The 5GT-SO embeds the network service orchestrator (NFV-NSO) and the resource orchestrator (NFVO-RO) with functionalities equivalent to those of a regular NFV orchestrator and it may be used for single and multi-domains [16].

Since the network slices handled at the 5GT-VS will in general serve complex end-to-end services, in the general case, the corresponding network service will be a composition of nested NFV-NSs. The lifecycle management of this complex NFV-NS is the role of the NFV-NSO.

In case a network service is requested that must be distributed across multiple domains, the 5GT-SO receiving the request becomes the parent NFV-NSO and sends ETSI GS NFV-IFA 013 [23] requests for each of the constituent NFV-NSs to other NFV-NSOs. Therefore, a hierarchy of NFVO-NSOs is established. The child NFVO-NSOs may belong to the same 5GT-SO that received the request from the 5GT-VS or to a peer 5GT-SO, which, in turn, may establish an additional hierarchy, which is transparent for the parent NFVO-NSO. The child NFVO-NSO belonging to the same 5GT-SO would be in charge of the lifecycle management of the constituent service that is eventually deployed over the local 5GT-MTP, i.e., the 5G-MTP with which the 5GT-SO has a direct relationship through the So-Mtp interface. When a child NFVO-NSO belongs to a different domain, there is service federation.

Eventually, a resource-related request is generated towards the underlying NFVO-RO to assign virtual resources towards the deployment of the (constituent) NFV-NS. The NFVO-RO functionality of the 5GT-SO handles resources coming from the local 5GT-MTP (real or abstracted) and from the 5GT-SOs of other administrative domains (abstracted). The NFVO-RO will decide on the placement of the Virtual Network Functions (VNF) of the NFV-NS based on the information available in the NFVI resources repository and the NFV instances repository. Most likely, the information available in these repositories will be more detailed when coming from the local 5GT-MTP than from a federated domain.

As for the NFV infrastructure as a service (NFVIaaS) use case, the 5GT-VS will request the 5GT-SO for a set of virtual resources, as opposed to a complete E2E NFV-NS as before. Therefore, this request is directly handled by the NFVO-RO, which is in charge of allocating resources either from the local 5GT-MTP or from a peer 5GT-SO. The latter option corresponds to resource federation. In this case, the request from the local NFVO-RO will reach the NFVO-RO of the peering domain. In all cases, the interaction between NFVO-ROs is based on ETSI GS NFV-IFA 005 [17]. This also includes the interface with the 5GT-MTP, where an additional NFVO-RO lower in the hierarchy is embedded, as explained below.

Notice that the NFVI resources handled by the NFVO of the 5GT-SO based on which decisions are taken will have a higher or lower abstraction level depending on the policies applied in this respect by the 5GT-MTP and the peering 5GT-SO. In general, the NFVO-RO of the local 5GT-SO will take coarse-grained decisions and the 5GT-MTP and peer 5GT-SO will take finer-grained ones, since they are closer to the actual resources.

The 5GT-SO also embeds the Virtual Network Function Managers (VNFM) to manage the lifecycle of the VNFs composing the NFV-NS. ETSI GS NFV-IFA 006-based interfaces [18] will be used to allow the VNFM interacting with the NFVO-RO Single Logical Point of Contact (SLPOC) entity in the 5GT-MTP, as well as peer SOs for resource allocation requests involving the VNFs under its control. For managing the VNF instances, ETSI GS NFV-IFA 008-based interfaces [20] will be used to allow the VNFM to directly configure the VNF instances running in the 5GT-MTP.

2.3 Mobile Transport and Computing Platform (5GT-MTP)

The 5GT-MTP [42] is responsible for orchestration of resources and the instantiation of VNFs over the infrastructure under its control, as well as managing the underlying physical mobile transport network, computing and storage infrastructure. In general, there will be multiple technology domains (TD) inside a 5GT-MTP (e.g., data centres, mobile network, wide area network). The 5GT-MTP NFVO-RO acts as end-to-end resource orchestrator across the various technology domains of the 5GT-MTP. The computing and storage infrastructure may be deployed in central data centres as well as distributed ones placed closer to the network edge, as in MEC [43]. Therefore, the 5GT-MTP is in charge of managing the virtual resources on top of which the NFV-NSs are deployed.

In terms of resource orchestration, the NFVO-RO acts as single entry point, i.e., single logical point of contact (SLPOC) in ETSI GS NFV-IFA 028 [26] terminology, for any resource allocation request coming from the SO. The So-Mtp interface is based on

ETSI GS NFV-IFA 005 [17] and ETSI GS NFV-IFA 006 [18]. The former allows the NFVO-RO of the 5GT-SO to request resource allocations to the NFVO-RO of the 5GT-MTP, whilst the latter allows the VNFM of the 5GT-SO to request resource allocations for the VNF under its control.

In terms of managing VNF instances, the So-Mtp interface also consists of ETSI GS NFV-IFA 008-based interfaces [20] to allow the VNFM of the 5GT-SO to directly configure the VNF instances running in the 5GT-MTP.

Depending on the use case, the 5GT-MTP may offer different levels of resource abstraction to the 5GT-SO. However, the 5GT-MTP NFVO-RO has full visibility of the resources under the control of the Virtual Infrastructure Managers (VIM) managing each technology domain, since they belong to the same administrative domain. ETSI GS NFV-IFA 005-based interfaces [17] are deployed between the 5GT-MTP NFVO-RO and the 5GT-MTP VIMs. Therefore, when receiving a resource allocation request from the 5GT-SO, the 5GT-MTP NFVO-RO generates the corresponding request to the relevant entities (e.g., VIM or WAN Infrastructure Manager (WIM)), each of them providing part of the virtual resources needed to deploy the VNFs and/or configure the relevant parameters of the PNFs that form the NFV-NS. As a special case, a resource request may be translated into an ETSI GS NFV-IFA 013-based NFV-NS request [23] towards a mobile network technology domain [16]. This option is offered to hide the complexity of the mobile network to the rest of the system whilst keeping the required flexibility inside the mobile domain (e.g., to decide on the most appropriate functional split). Therefore, a full ETSI MANO stack is represented in technology domain 1-2 (see Figure 1) even if the focus of the 5GT-MTP is handling virtual resources and not NFV-NSs. In any case, this NFV-NS is hidden to the 5GT-SO, since it is abstracted as a virtual link.

2.4 Monitoring Architecture

In the 5G-TRANSFORMER framework, each architectural component (i.e. 5GT-VS, 5GT-SO, 5GT-MTP) includes a monitoring service able to provide performance metrics and failure reports targeting the logical entities managed by each component. Following this approach, the 5GT-MTP monitoring service will produce monitoring data about the local physical and virtual resources, the 5GT-SO monitoring service will produce monitoring data about the managed VNFs and NFV network services, while the 5GT-VS monitoring service will produce monitoring data about network slices and vertical services. This hierarchy of monitoring services is shown in Figure 2, where the arrows indicate a consumer-provider interaction. In particular, the 5GT-SO monitoring service can be a consumer of the monitoring service provided by the underlying 5GT-MTP or by a federated 5GT-SO, while the 5GT-VS can be a consumer of the monitoring service provided by the local 5GT-SO.

The monitoring data generated at each layer can be used to feed internal decisions within each architectural component or to serve external consumers of monitoring data. For example, the 5GT-SO monitoring service can elaborate performance metrics about an NFV network service, and these metrics can be used by the 5GT-SO to take scaling decisions for the involved VNFs. On the other hand, the performance metrics computed at the 5GT-SO monitoring service can be provided to the 5GT-VS monitoring service for further elaboration. When metrics and alerts are exchanged between two monitoring services, the level of visibility and disclosure of monitoring information should be regulated based on authorization policies and business agreements, in particularly when monitoring data that belongs to different administrative entities. This may be the case, for example, between the 5GT-MTP and the 5GT-SO monitoring services when they are handled by different actors or between the monitoring services of federated 5GT-SOs.

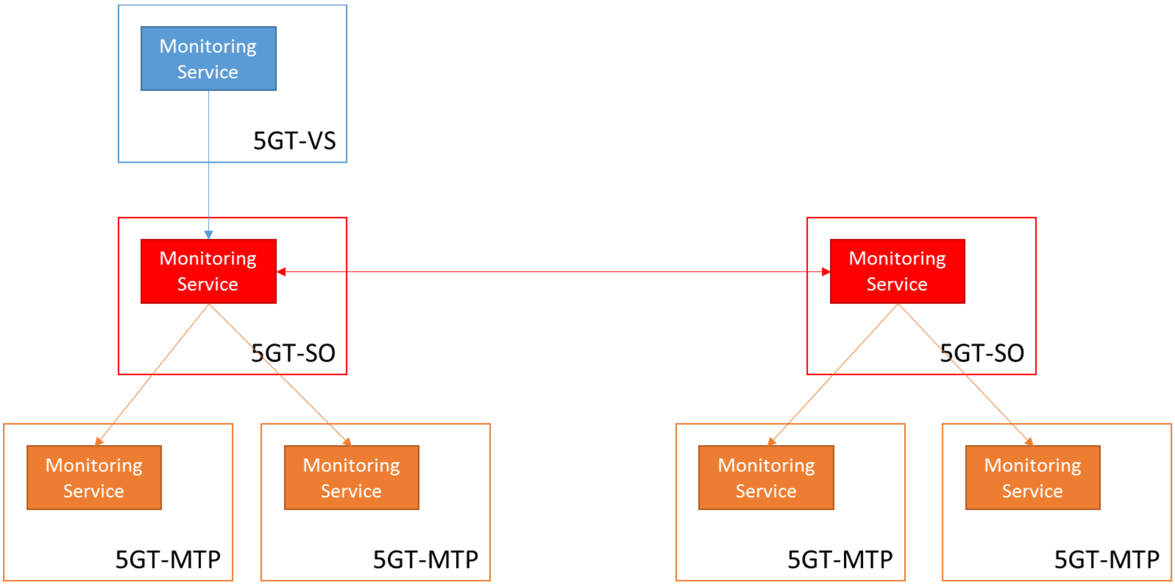


FIGURE 2: HIERARCHY OF MONITORING SERVICES IN 5G-TRANSFORMER ARCHITECTURE

It is important to highlight that the 5G-TRANSFORMER architecture does not impose any constraint on the monitoring platform implementation, but defines just the expected behavior of the service and the external APIs that each monitoring platform should expose to the consumers of its monitoring data. This means that each actor may implement its own specific monitoring platform and in case of overlapping roles, like in the 5GT-VS and 5GT-SO case where they are owned and managed by the same administrative entity, a single monitoring platform may be deployed for both of them.

3 Requirements on the 5GT-SO

Technical requirements on the overall 5G-TRANSFORMER system have been defined in [9]. The requirements covered in [9] focus on properties related to vertical services and relevant use cases. General requirements related to the overall system are described in [10]. In this section, we define business (Section 3.1) and functional (Section 3.2) requirements specific to 5GT-SO. The notation used to refer to the different requirements is described in Section 11 (Annex III).

3.1 Business Requirements

5G-TRANSFORMER platform is designed to foster business relationships with other administrative domains via service and resource federation. The 5GT-SO is the entity responsible to manage these business relationships and to achieve so, the following table enumerates the business requirements we consider to be fulfilled by the 5GT-SO within the 5G-TRANSFORMER architecture.

TABLE 1: BUSINESS REQUIREMENTS

ID	Requirement	F/NF
ReqSO.B01	The 5GT-SO shall include a REST interface with the 5GT-VS.	F
ReqSO.B02	The 5GT-SO should be able to accept a network service specification from the 5GT-VS including both functional and non-functional requirements expected for the requested service.	F
ReqSO.B03	The 5GT-SO should manage the collaboration of federated 5G-TRANSFORMER administrative domains for the completion of the NFV NS	F
ReqSO.B04	The 5GT-SO shall expose appropriate interfaces to federated 5GT administrative domains and to the 5GT-MTP.	F
ReqSO.B05	The 5GT-SO shall orchestrate the services requested by the 5GT-VS using the resources exposed by the 5GT-MTP and/or the resources exposed by federated 5GT administrative domains.	F
ReqSO.B06	The 5GT-SO shall expose a subset of the resources coming from the 5GT-MTP to federated 5GT administrative domains.	F
ReqSO.B07	The 5GT-SO shall expose a subset of its own network services to federated 5G-TRANSFORMER administrative domains	F
ReqSO.B08	The 5GT-SO shall use the PNFs exposed by the 5GT-MTP for service completion.	F
ReqSO.B09	The 5GT-SO must adhere to industry multi-tenancy requirements including isolation, scalability, elasticity and	NF

	security.	
ReqSO.B10	The 5GT-SO shall allow monitoring with an appropriate granularity according to the network service characteristics.	F
ReqSO.B11	The 5GT-SO must provide the 5GT-VS with a network service catalogue with information about available service offers and capabilities, in order to facilitate the automated provision of services.	F
ReqSO.B12	The 5GT-SO must provide a mechanism to perform network service accounting and charging. This information should be available internally and externally (for the 5GT-VS).	F
ReqSO.B13	The 5GT-SO should be able to support long-live and short-lived services.	F
ReqSO.B14	The 5GT-SO should be reliable.	NF
ReqSO.B15	The 5GT-SO should be available (as carrier class component).	NF
ReqSO.B16	The 5GT-SO should keep responsiveness for the 5GT-VS and federated 5GT domains.	NF
ReqSO.B17	The 5GT-SO shall allow defining composed network services.	F
ReqSO.B18	The 5GT-SO shall support the selection of preferred, non-preferred, and prohibited virtual infrastructure providers for the network service instantiation.	F
ReqSO.B19	The 5GT-SO shall support to select the deployment area based on KPIs[2] of another service.	F

3.2 Functional Requirements

The 5GT-SO is involved in the service lifecycle at different stages. Thus, different requirements can be considered according to each stage, namely (1) Discovery, (2) Fulfilment, (3) Assurance, and (4) Decommissioning.

3.2.1 Discovery

The discovery phase facilitates the 5GT-SO to understand which capabilities and services (in terms of descriptors) are supported by 5GT-SO. That information will be exposed to the 5GT-VS and to federated 5GT domains for 5G-TRANSFORMER service offering.

The following requirements are identified:

[2] As an example, intersection collision avoidance should cover critical intersections, where ‘critical’ is defined in terms of occurrence of abrupt braking manouvers in the past.

TABLE 2: REQUIREMENTS ON THE DISCOVERY PHASE

ID	Requirement	F/NF
ReqSO.Di.01	The 5GT-SO must provide the 5GT-VS with the means to send detailed requests including information regarding the placement of resources, the location of service points, QoS, charging options.	F
ReqSO.Di.02	The NS/VNF catalogue entries may contain a service manifest and a price tag (or an indicative price range from which the exact price can be extracted at run-time).	F
ReqSO.Di.03	The 5GT-SO shall keep an up-to-date network service catalogue with the network services and VNFs received from other federated 5GT administrative domains.	F
ReqSO.Di.04	The 5GT-SO should provide a mechanism to set-up, re-size and terminate network services.	F
ReqSO.Di.05	The 5GT-SO shall allow a TSP to store network service descriptors persistently and to: retrieve, update, and delete them.	F
ReqSO.Di.06	The 5GT-SO shall keep an up-to-date catalogue with the resources, links, connection points and PNFs as exposed by the 5GT-MTP abstraction.	NF
ReqSO.Di.07	The 5GT-SO shall keep an up-to-date catalogue with the resources exposed by other federated 5GT administrative domains.	NF

3.2.2 Fulfilment

During the service fulfilment phase, the 5GT-SO orchestrates (namely, creates and instantiates) network services requested by 5GT-VS.

The following requirements are identified:

TABLE 3: REQUIREMENTS ON THE FULFILMENT PHASE

ID	Requirement	F/NF
ReqSO.Fu.01	The 5GT-SO should allow scaling (up / down) as part of the lifecycle management	F
ReqSO.Fu.02	The 5GT-SO shall support to manage the lifecycle of each of the service instances separately.	F
ReqSO.Fu.03	The 5GT-SO shall support the management of NSDs	F
ReqSO.Fu.04	The 5GT-SO shall support the management of VNFDs.	F
ReqSO.Fu.05	The 5GT-SO shall allow connecting network service instances.	F
ReqSO.Fu.06	The 5GT-SO shall be able to translate the network service request to one or more resource allocation or lifecycle	F

	actions reflecting the agreed service levels in each case.	
ReqSO.Fu.07	The 5GT-SO shall allow creating several instances of the same network service.	F

3.2.3 Assurance

5GT-SO is responsible to guarantee the performance agreements made with 5GT-VS for orchestrated network services and provide 5GT-VS with sufficient monitoring information of said network services.

The following requirements are identified:

TABLE 4: REQUIREMENTS ON THE ASSURANCE PHASE

ID	Requirement	F/NF
ReqSO.As.01	The 5GT-SO must provide the vertical slicer with APIs to monitor the QoS attained for the requested service.	F
ReqSO.As.02	The 5GT-SO should provide isolation among service requests.	NF
ReqSO.As.03	The 5GT-SO could provide resources from other 5GT-SO using federation.	F
ReqSO.As.04	The 5GT-SO shall be able to collect and provide performance information related with the NSs.	F
ReqSO.As.05	The 5GT-SO shall manage fault information, reacting when necessary and generating alarms to the 5GT-VS when the fault cannot be solved at the 5GT-SO level.	F

3.2.4 Decommissioning

Once a network service is decommissioned, 5GT-SO shall properly release the used resources and terminate the required VNFs.

The following requirements are identified:

TABLE 5: REQUIREMENTS ON THE DECOMMISSIONING PHASE

ID	Requirement	F/NF
ReqSO.De.01	The 5GT-SO should be able to identify the monitoring mechanisms to be de-activated as a result of a service termination.	F
ReqSO.De.02	The 5GT-SO should have means for receiving acknowledgement of successful actions of resources release.	F
ReqSO.De.03	The 5GT-SO should be able to notify the vertical slicer about a service termination.	F

4 5GT-SO Architecture

In this section we introduce the up-to-date design of 5G-TRANSFORMER's service orchestrator (i.e. the 5GT-SO). A study on the state-of-the-art solutions that have served as a basis for our design can be found in the annex (Section 13). In the sequel, we will first present an overview of the role of 5GT-SO in the overall system architecture. We will then introduce the architectural description of 5GT-SO and key supported operations. Finally, we will provide details on the design of the northbound interface to the vertical slicer, southbound interface to the mobile transport and computing platform, east/westbound interface to federated orchestrators and an overall architectural description of the monitoring platform.

4.1 5GT-SO Overview

The 5GT-SO is in charge of end-to-end (E2E) service orchestration and federation of networking, computing and storage resources across one or multiple 5GT-MTP domains, in addition to managing the allocation of different vertical slices. 5GT-SO receives the service requirements from M(V)NOs and/or vertical industries via the SO-SAP interface (see Figure 1 in Section 2) in the shape of a Network Service Descriptors (NSD). In order to do so, the 5GT-SO (i) decides the optimal resource allocation for the whole NFV-NS, (ii) decides the optimal placement of VNF/VAs,[3] (iii) decides the optimal deployment of virtual links connecting VNF/VAs, (iii) and requests federated services and/or resources when needed, in addition to some other tasks related to monitoring and management (e.g. VNF management, consistency check on requested NSDs, etc.).

Service orchestration focuses on management, instantiation, and migration of VNFs/VAs at local, edge and cloud NFVIs. The problem of mapping VNFs/VAs to (virtual) computing entities (nodes, NFVI-PoPs) and the mapping of virtual links between VNFs/VAs into (virtual) paths, depending on the granularity of abstraction offered by the 5GT-MTPs, can be tackled by different optimization strategies, namely heuristics or mixed-integer linear programming. Moreover, automatic network service management and self-configuration algorithms (e.g., failure recovery) are also required to adapt to network changes and special events triggered by the monitoring platform.

In case the 5GT-SO detects that one 5GT-MTP domain alone has not enough infrastructure resources to orchestrate the required service, it interacts with other SOs via the So-So interface (see Figure 1 in Section 2) to compose a service across multiple federated administrative domains. In this case, 5GT-SO will dynamically discover the available administrative domains by exchanging the view with the neighboring 5GT-SOs, and negotiate with them the needed services and resources.

4.2 5GT-SO Architecture and Key Operations

Figure 3 presents a high level overview of 5GT-SO subsystems and their interactions designed to achieve the essential 5GT-SO operation described before. The described 5GT-SO design follows ETSI guidelines [16] and is in line with orchestration system

[3] The granularity that 5GT-SO has when placing functions (NFVI-PoPs, servers, etc.) depends on the level of abstraction negotiated with the different 5GT-MTPs.

designs developed in related EU projects (i.e., 5GEx [33] and 5G-Crosshaul [44]). The main building blocks comprising 5GT-SO are the following:

- **NBI Exposure Layer:** This layer offers a Northbound API towards the 5GT-VS to support requests for service on-boarding, service creation, service instantiation, service modification, and service termination. Details on supported API are given in subsection 4.4.

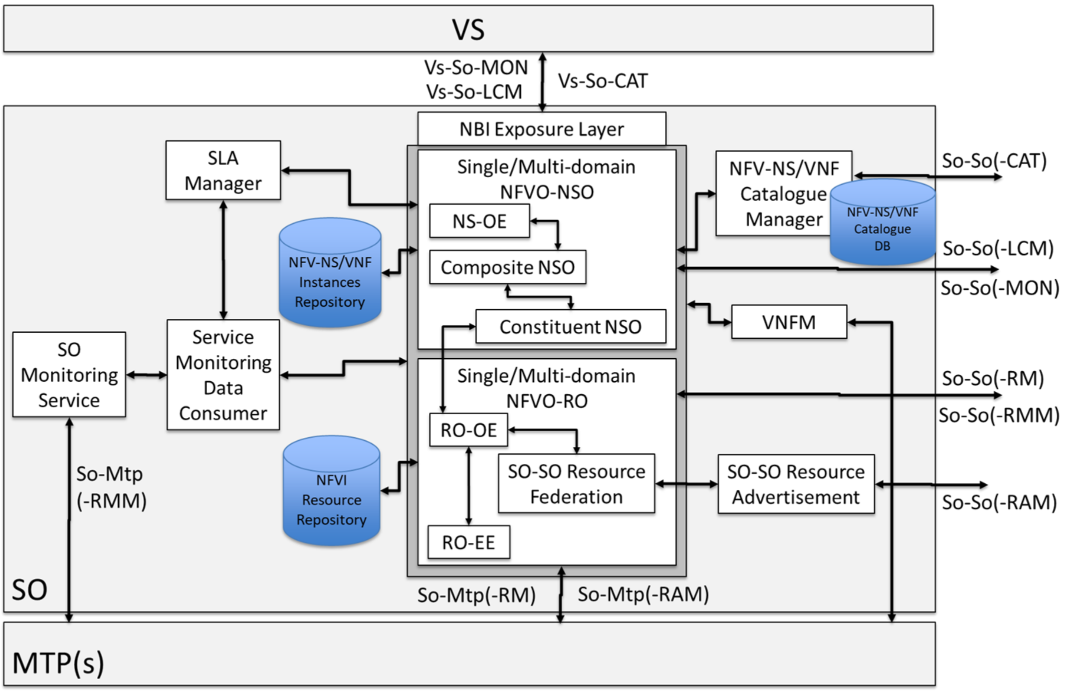


FIGURE 3: 5GT-SO SYSTEM ARCHITECTURE - BUILDING BLOCKS AND THEIR INTERACTIONS

- **NFV-NS/VNF Catalogue DB/Manager:** Catalogue DB is the repository of all usable Network Service Descriptors (NSDs) and VNF Descriptors (VNFDs) that can be accessed through the Catalogue Manager. A NSD describes a Network Service (NFV-NS) that 5GT-SO is able to provide (either by its own or by leveraging neighboring SOs); and it is expressed in terms of chaining of VNF components and providing description of their connectivity (i.e., virtual links) and resource requirements. A VNFD describes a VNF in terms of its deployment and operational behavior requirements. The NSD/VNFD is used by the 5GT-SO in the process of NFV-NS/VNF instantiation and its lifecycle management to obtain relevant information, e.g., deployment flavors or out-scaling rules. The Catalogue Manager also takes care of the advertising of NFV-NSs for federation purpose.
- **NFV Orchestrator (NFVO):** NFVO has the responsibility of orchestrating virtual resources across multiple domains, fulfilling the Resource Orchestration (NFVO-RO) functions, as well as of coordinating the deployment of NFV-NSs along with their lifecycle management, thus fulfilling the Network Service Orchestration (NFVO-NSO) functions. More specifically:
    - NFVO-NSO coordinates all the NFV-NS deployment operations including Authentication, Authorization and Accounting (AAA) as well as formal checks of service requests based on attributes retrieved from NSDs and

VNFDs. In particular, the Composite NSO, using the algorithms implemented in the NFV-NS Orchestration Engine (NS-OE), decomposes the NSDs into several segments and decides where to deploy them, i.e., whether using a local 5GT-MTP or leveraging neighbor SOs. Accordingly, the Composite NSO requests (i) the Constituent NSO and then the local NFVO-RO to implement the NFV-NS segment into its administrative domain; and/or (ii) the federated NFVO-NSO to implement the NFV-NS segment(s) into the other administrative domains. Finally, the NFVO-NSO is responsible for the network service lifecycle management including operations such as service on-boarding, instantiation, scaling, termination, and management of the VNF forwarding graphs associated to the network services.

- NFVO-RO maps the NFV-NS segment into a set of virtual resources through the RO Orchestration Engine (RO-OE) by deciding the placement of each VNF within the virtual infrastructure, based on specified computational, storage and networking (e.g., bandwidth) requirements. The decision is based on available virtual resources that are exposed by the 5GT-MTP via the So-Mtp Southbound Interface (SBI) or by from other domains through the So-So/East-Westbound Interface (EBI/WBI). In the latter case, the sharing of abstract views is needed to build-up a comprehensive view of resources available from different domains and is carried out by the SO-SO Resource Federation element. Then, the RO Execution Entity (RO-EE) takes care of resource provisioning by managing the coordination of correlated actions to execute/forward the allocation requests to either 5GT-MTP or to the 5GT-SO NFVO-RO of other domains.

- **VNF Manager (VNFM):** the VNFM is in charge of the lifecycle management of the VNFs deployed by the 5GT-SO using either local or remote resources (or a combination of thereof). It receives relevant VNF lifecycle events from the local NFVO and provides reconfiguration according to specified counteractions decided by the NFVO based on VNFDs (e.g., auto-scaling).
- **SO-SO Resource Advertisement:** This block is in charge of exchanging abstract resource views (e.g., abstract topologies, computing and storage capabilities) with other domains while feeding the 5GT-SO Resource Federation entity that consolidates inputs and stores federated resources into the NFVI Resource Repository.
- **NFVI Resource Repository:** This repository stores consolidated abstract resource views received from the underlying 5GT-MTPs, either from the So-Mtp Southbound Interface (SBI) or from the SO-SO Resource Federation block in case of abstract resource views received from other SOs/domains through the So-So/East-Westbound Interface (EBI/WBI).
- **NS/VNF Instance Repository**: This repository stores the instances of VNFs and NFV-NSs that have previously been instantiated.
- **SO Monitoring Service**: This block provides the measurement reports for the 5GT-SO to support 5GT-SO monitoring management including performance monitoring and fault management, based on the collected monitoring data provided by the 5GT-MTP.
- **Service Monitoring Data Consumer:** This block supports the lifecycle management of instantiated VNFs/NFV-NSs by collecting measurement reports from the 5GT-SO

Monitoring Service and reports data to the NFVO (e.g., to trigger auto-scaling actions based on scaling rules in the NSD) and/or to the SLA Manager (e.g., to enable SLA on-line verification). Performance reports can be also used to trigger healing actions to recover from failures or service degradations. The aim is to adapt deployed services or provisioned resources while preventing service degradations due to the concurrent usage of resources from different services.

- **SLA Manager:** This block elaborates performance reports from the Service Monitoring Data Consumer during the service lifecycle and assures that the agreed SLAs are continuously satisfied through on-line SLA verification. In the event a requested SLA is not met, the SLA Manager may trigger scaling actions to prevent or recover from SLA violations.

Those functionalities are supposed to interwork together towards the support of a number of operations. Such operations are described below and are the basis for the identification of relevant workflows that are described in the Section 5.

1. *Detection of other SOs for federation*

This operation is a prerequisite to achieve federation; that is, 5GT-SO shall be made aware of the existence of other SOs, and thus become part of the service ecosystem. Indeed, as 5GT-SO is aware of other SOs, it is able to (i) exchange resource/service capabilities with neighboring SOs, and (ii) cooperate to accomplish service deployments across administrative domains. The detection of neighboring SOs can be the result of either a static (i.e., manual pre-configuration) or automatic mechanisms (i.e., auto-discovery). Without loss of generality, the static case is firstly considered in 5G-TRANSFORMER.

2. *Resource/Service capabilities advertisement*

With this operation, 5GT-SO becomes aware of the resource and service capabilities of SOs in other administrative domains in order to make resource and service orchestration decisions. The exchanged capabilities are related both to supported VNFs and services (i.e., catalogues) and to topology and other resource-related information (e.g., abstract network topology, cloud resource capabilities).

3. *Deployment*

This operation is triggered as a request for a NFV-NS arrives either from the Vertical Slicer or from neighbor SOs. The deployment operation is carried out by the orchestration engine (i.e., NFV Orchestrator (NFVO)) both at the *Network Service* level, handled by the Network Service Orchestrator (NFVO-NSO), and at the *Resource* level, handled by the Resource Orchestrator (NFVO-RO), according to the ETSI MANO guidelines [16]. The deployment operation may involve either 5GT-MTP(s) in the same administrative domain of 5GT-SO (single-domain case) or other SOs in different administrative domains (federation case).

4. *SLA assurance*

Once a network service is deployed, an assurance phase begins. This phase consists of collecting and processing monitoring data during the service lifecycle in order to check if the agreed SLAs are continuously met (i.e., on-line SLA verification). The assurance phase is carried out by an SLA manager supported by a Service Monitoring Data Consumer that collects performance metrics by interacting with the 5GT-SO

Monitoring Service. In case SLA parameters are not satisfied and/or failures/degradations occur, 5GT-SO may trigger either scaling actions (through the NFVO-NSO or SLA Manager) or healing actions, respectively, by means of proper resource/service reconfigurations or service lifecycle commands.

5. *Reconfiguration*

Reconfiguration operations can be triggered in case of *(i)* requested SLA parameters are not satisfied (then the SLA Manager or NFVO-NSO trigger scaling actions); *(ii)* failures in, e.g., a VNF (then different resources could be selected to support such VNF); (iii) service degradation (then resources should be re-allocated to preserve the performance of high-priority services while optimizing the overall resource usage).

The interfaces to other SOs, 5GT-MTPs and 5GT-VS are described in Section 5.

4.3 Federation

Federation is a mechanism for integrating multiple administrative domains at different granularity into a unified open platform where the federated resources and services can trust each other at a certain degree (see Figure 4). An administrative domain is a collection of network services and resources operated by a single organization. The administrative domain is viewed as complete entity and its internal structure is hidden or unimportant from outside. The resources and services inside the administrative domain operate with high degree of mutual trust among themselves, but the interaction with other administrative domains - is subject to stringent trustworthiness constraints, with a default high level of alert. The federation is formed in order to increase the degree of trust among different administrative domains with a goal of better interoperability of services and resources. Embodiment of a service/business-level agreement or partnership between two administrative domains is a federation of trust [59].

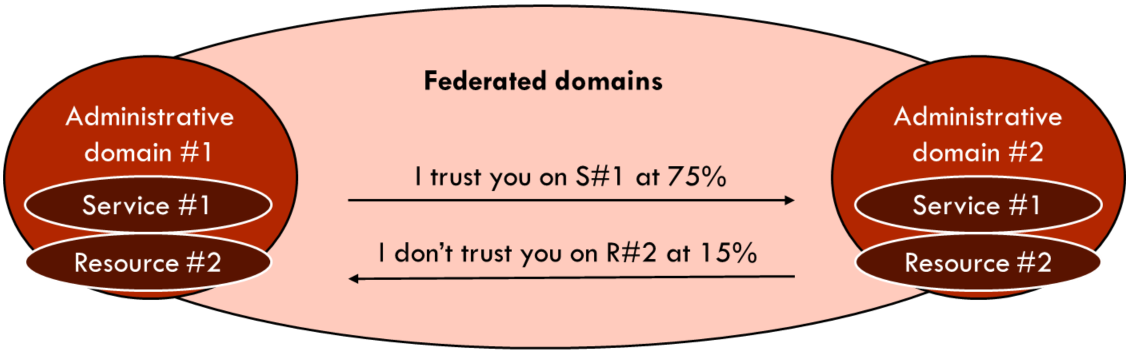


FIGURE 4: FEDERATION AS A DOMAIN UNIFIED BY MUTUAL TRUST (FROM [59])

4.3.1 Federation Levels

Each agreement defined between two administrative domains have different terms and conditions that both should follow to maintain their partnership or federation relationship. A single administrative domain can have multiple different peer-to-peer agreements with other administrative domains. Each agreement, according to agreed terms and conditions, can belong to a different category of federation relationship or different federation level. The federation levels indicate the mutual degree of trust among administrative domains and they can be defined as bronze, silver, gold, platinum, etc. In that sense, if an administrative domain has low-level of trust or intentions to share only limited resources and/or services with other administrative domain, both would agree to terms and conditions of a bronze federation level. On the contrary, if two administrative domains establish high degree of mutual trust and share more transparent view on their resources and/or services, the federation would belong to a platinum federation level. For higher federation levels, significantly broader options and specific information parameters about resources and/or services are exchanged between the administrative domains. For the lower levels, the offering of resources and/or services is limited and information for parameters is more abstract and descriptive. How these levels should be achieved and to what extent is for further study. However, throughout the document we would refer to the federation levels mentioned in this section such as bronze, silver, gold and platinum, referring from low trust levels (high abstractions or hiding lot of details) to higher trust levels (lower abstractions or revealing more details).

4.3.2 Federation in 5G-TRANSFORMER

In 5G-TRANSFORMER, the federation assumes already established business level and service level agreements. It is assumed that the complete relationship between external administrative domains are already defined. The federation levels are taken into account and the scope is on the technical level.

From an architectural perspective, the federation in 5G-TRANSFORMER occurs only on the service orchestrator level. More precisely all external connections are established by the 5GT-SO via EBI/WBI.

Once all business agreements are settled by administrative domains that implement the 5G-TRANSFORMER system, the federation itself is implemented as two categories of services: NFV-NSaaS and NFVIaaS.

Network Service as a Service (NFV-NSaaS) represents the technical part of federation of services (NFV network services).

NFVI as a Service (NFVIaaS) represents the technical part of federation of resources.

In the next sections, both the NFV-NSaaS and NFVIaaS are examined in detail.

4.3.3 Service Federation

Service federation is the overall process of establishing, using or providing NFV network services by an administrative domain from/to other (peering/partner) administrative domain. The administrative domain that requests service is referred as consumer domain while the peering administrative domain capable of providing service is a provider domain. The service federation is the combination of establishing business/service level agreements among the administrative domains and requesting/providing NFV network service defined as NFV-NSaaS. The first phase is already explained in the introduction section and it is required for enabling NFV-NSaaS. The details about the business/service agreements are out of scope of 5G-TRANSFORMER. The details of NFV-NSaaS are the scope of this section. Sub-section 4.3.3.1 focuses on the ETSI GS NFV-IFA 028 [26] solution for multi-domain orchestration. The rest of the sections are focusing on the federation of services in 5G-TRANSFORMER, specifically on the 5GT-SO.

4.3.3.1 NFVO Multi-domain orchestration

The use case of offering NFV-NS to other administrative domains is covered in ETSI GS NFV-IFA 028 [26]. In the reference document, it is described how composite NFV-NS can span onto separate administrative domains can be achieved. NFVO Multi-domain orchestration of NFV-NSs is done upon decomposition of NFV-NS received from the 5GT-VS by the 5GT-SO NFVO. The result of decomposing a NFV-NS is a single or set of multiple nested NFV-NS. The decomposition of the requested NFV-NS is for further study. However, a single nested NFV-NS can be instantiated on a different administrative domain as a federated NFV-NS and together with the constituent nested NFV-NSs form the requested composite NFV-NS. According to ETSI GS NFV-IFA 028 architectural approach [26], the NFVO Multi-domain orchestration of NFV-NSscan be established using Or-Or interface between the NFVO belonging to a consumer domain and the NFVO belonging to a provider domain. The consumer NFVO is requesting NFV-NS to the provider NFVO. Once the NFV--NS is instantiated, the consumer NFVO can request LCM operations to the provider NFVO. However, the provider domain is responsible for all NFVO functionalities and Lifecycle Management (LCM) operations over the established NFV-NS. The consumer NFVO is not aware of the resources and VNFs that are part of the provider NFVO.

4.3.3.2 Catalogue of services

The catalogue of services can be formed dynamically or in a static approach. The dynamic approach is when each administrative domain (5GT-SO) locally forms a list of NFV-NSs that can provide as federated NFV-NS to other peering administrative domains, based on the capabilities and on-boarded NSDs. The formed list is broadcasted to the peering administrative domains and stored in their Catalogue DB. Note that, at this point, the network of peering administrative domains is assumed to be already established (section 6.2.2). Optionally, the catalogue list can be dynamically extended due to on-boarding of a new NSDs or by a request from peering 5GT-SOs (other administrative domains). The dynamic approach is intended for further study.

The static approach is simpler. Each administrative domain implementing the 5G TRANSFORMER system estimates its capabilities and conducts a list of possible services that it can offer as provider of NFV-NSaaS to other partnering administrative domains. This list of services is later negotiated (e.g. on business meetings) with each partner and depending on the federation level and business/service level agreements, it is reduced or extended. The final and agreed modifications of the list of services represent catalogue of (offered) services that are delivered to the specific partnering administrative domain. The partnering administrative domain receives the catalogue of services and stores it in the NFV-NS/VNF Catalogue Database of the 5GT-SO. All available services that can be consumed from external partnering administrative domains are stored in this database. In this case the catalogue would contain more generic NFV-NS (e.g. firewalling, domain name service, collision avoidance in automotive case, etc.).

4.3.3.3 NFV-NSaaS in 5GT-SO

Federation of services in 5GT-SO is realized with a similar approach as in ETSI GS NFV-IFA 028 [26]. The 5GT-SO NFVO-NSO is the enabling point of NFV-NSaaS for consuming or providing external (federated) NFV network services. The 5GT-SO NFVO-NSO can simultaneously consume and provide the NFV-NSaaS. The external connection between 5GT-SO NFVO-NSOs (belonging to different administrative domains) is realized by using reference points of the EBI/WBI (Section 4.6): So-So-LCM, So-So-MON and So-So-MON.

The 5GT-SO receives a request for instantiation composite NFV-NS from 5GT-VS. The 5GT-SO NFVO-NSO first makes decomposition of the requested NFV-NS into several segments or several nested NFV-NSs. The decomposition of the services is for further study. The 5GT-SO NFVO-NSO can decide one (or more) of the decomposed NFV-NSs to be consumed through NFV-NSaaS or in other words to use federated NFV-NS. The reasons can be various such as: lack of resources provided by the local 5GT-MTP, different target location that is not supported by the administrative footprint, extension of services, etc.

Upon the decision for consuming NFV-NSaaS, the 5GT-SO NFVO-NSO would query peering administrative domain for availability of providing the requested nested NFV-NS. Note that the information of availability is previously stored in the local Catalogue Database (DB) and the network connections to peering 5GT-SOs are already established. Optionally, the consumer 5GT-SO can provide an NSD to the peering (provider) 5GT-SO for the requested NFV-NS for federation (e.g. for more specific deployment flavor). This optional case is for further study.

The workflow of establishing an NFV-NSaaS or service federation is covered in Section 5.6.

As in ETSI GS NFV-IFA 028 [26], the consumer 5GT-SO NFVO-NSO is not aware of the resources and VNFs that the provider 5GT-SO NFVO-NSO is using to provide the NFV network service. The consumer 5GT-SO NFVO-NSO is using the So-So-LCM to send requests for lifecycle operations of the provided NFV network service. The lifecycle operations are performed by the provider 5GT-SO NFVO-NSO. The So-So-MON allows exchange of limited information consisting of performance indicators and fault alarms. According to the already negotiated terms and conditions, some performance indicators may be hidden from the consumer domain. The 5GT-SO NFVO-NSO is using the So-So-CAT reference point to exchange catalogue updates, querying for NSDs or on-boarding MEC AppDs.

4.3.4 Resource Federation

Resource federation is the overall process of establishing and using NFVI resources by a consumer domain from a provider domain. The resource federation process is divided between establishing business/service level agreements and providing/consuming NFVI resources. As in the service federation, the business/service level agreement part is out of scope. Requesting/providing NFVI resources is defined as NFVI as a Service (NFVIaaS). The usage of the NFVI resources is limited to the agreed terms and conditions. The reasons for using federation of resources are covered in section 4.3.4.1. There are multiple options of how the federation of NFVI resources can be performed, however the realization of it is covered in section 4.3.4.2. The whole process of NFVIaaS includes two phases: 1) advertising phase and 2) allocation and management of federated resources. The advertisement phase is covered in section 4.3.4.3 and the allocation and management of federated resources is covered in 4.3.4.4.

4.3.4.1 Motivation

The 5GT-SO would consume NFVIaaS for similar reasons as in the case of NFV-NSaaS, however with significant difference. As in the case of NFV-NSaaS the 5GT-SO may lack of resources or has a need to extend the footprint. However, in the case of NFVIaaS, the consumer 5GT-SO is in charge of managing the consumed resources. It is important to note that the exposure of parameters or detail information about consumed resources, depends on the federation level established between SOs. The federated resources would be abstracted to a certain level by the provider 5GT-SO. The consumer 5GT-SO can include the federated resources in deploying NFV NSs or VNFs by combining them with local resources (belonging to local 5GT-MTP).

4.3.4.2 Single logical point of contact for federation (SLPOC-F)

The use case of offering NFVIaaS to external administrative domains is covered in ETSI GS NFV-IFA 028 [26]. In this document, multiple options are provided for realization of the NFVIaaS. The closest option to the 5G-TRANSFORMER approach is through using single logical point of contact between NFVO belonging to different administrative domains, in indirect mode. The consumer NFVO can request NFVI resources to the provider domain through a single logical point of contact (SLPOC-F) via Or-Vi interface (defined in ETSI GS NFV-IFA 005 [17]). The consumer NFVO is in charge of managing the lifecycle of the NFVI resources once granted by the provider SLPOC-F. In the 5G-TRANSFORMER case, the 5GT-SO NFVO-RO acts like SLPOC-F. The provider 5GT-SO NFVO-RO is the SLPOC-F and routes lifecycle messages received from the consumer 5GT-SO NFVO-RO that manages NFVI resource of the provider 5GT-MTP. There is no direct connection between the consumer 5GT-SO NFVO-RO and the provider domain 5GT-MTP.

4.3.4.3 Advertisement phase

From resource federation perspective, the business/service level agreements define only the relationship between two administrative domains. Only the general terms and conditions for sharing NFVI resources are defined. Since there is not a closed list of resources similar to the catalogue of services, the 5GT-SOs perform advertising phase where they exchange information about the NFVI resources that they can offer to other potential consumer 5GT-SOs. In this phase all 5GT-SOs have an equal role without being consumer or provider. The aim is each 5GT-SO to broadcast its potential for providing NFVIaaS to the peering 5GT-SOs and as well to discover back the potential capabilities of the peering 5GT-SOs, in case there is a need to consume NFVIaaS from the peering 5GT-SOs. The advertisement phase consists of two local operations and an outbound/inbound exchange of information:

- Calculating available resources - local operation performed by each 5GT-SO NFVO-RO. Extracting information from NFVI Database and performing the calculations of available NFVI resources.
- Generating resource abstractions - local operation performed by each SO NFVO-RO. The abstractions are generated according to federation levels, specified service level agreements, etc.
- Exchanging abstractions with peering 5GT-SOs - performed by SO-SO Resource Abstractions via the So-So-RAM reference point of the EBI/WBI. The exchange of information occurs in both directions. The generated resource

abstractions from the "Generation resource abstractions" operation are broadcasted in outbound direction to the peering 5GT-SOs. In inbound direction, the received abstractions from the peering 5GT-SOs are stored in the local database or NFVI Resource Repository.

The advertising phase occurs periodically or event-based. Upon NFV network service creation, modification (scaling up) performed by the 5GT-SO that directly changes the state of the available NFVI resources provided by the local 5GT-MTP, the advertisement phase should be repeated.

4.3.4.4 Allocation and management of federated resources

The 5GT-SO NFVO-RO SO-SO Resource Federation is responsible for deciding if there is a need for requesting external NFVI resources based on the calculations for available local 5GT-MTP resource abstractions in the NFVI Resource Repository. The decision can be affected by a requirement from the 5GT-SO NFVO-NSO. For example, a service may require footprint on a NFVI-PoP that does not belong to the local 5GT-MTP.

Establishing NFVIaaS or resource federation workflow is explained in section 5.7 of this document.

Once the federated NFVI resources are managed by a consumer 5GT-SO NFVO-RO, two reference points from the EBI/WBI are used. The So-So-RM is used for performing lifecycle management operations. The limitations of the operations directly depend of the federation level established with the provider domain. The So-So-RMM reference point is used for receiving monitoring information of the federated resources. The monitoring information (performance parameters and fault alarms) is limited and directly depending on the federation level.

4.4 5GT-SO NBI

The sequel briefly introduces the design of the 5GT-SO's northbound interface (NBI); additional details can be found in [12]. The NBI enables the interaction between the 5GT-VS and the local 5GT-SO, which is based on the following three reference points (see Figure 5):

- **Vs-So(-L**ife **C**ycle **M**anagement**)** is used for the operation of NFV network services. It offers primitives to instantiate, terminate, query, update or re-configure network services or receive notifications about their lifecycle;
- **Vs-So(-MONi**toring**)** is used for the monitoring of network services and VNFs through queries or subscriptions/notifications about performance metrics, VNF indicators and network services or VNFs failures;
- **Vs-So(-Catalogue)** is used for the management of Network Service Descriptors (NSDs), VNF packages, including their VNF Descriptors (VNFDs), and MEC Application Packages, including their Application Descriptors (AppDs). This reference point offers primitives for on-boarding, removal, updates, queries and enabling/disabling of descriptors and packages.

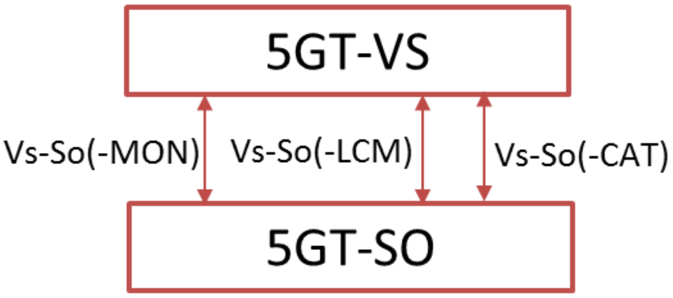


FIGURE 5: REFERENCE POINTS BETWEEN 5GT-VS AND 5GT-SO

The implementation of the 5GT-SO NBI is mostly based on ETSI GS NFV-IFA 013 specification [23], which defines the NBI of an NFVO. However, the 5GT-SO NBI implements further methods to allow the 5GT-SO to handle Network Services extended to include also MEC Applications. In terms of interfaces, this has an impact mostly on the management of the catalogues and on the queries of the NFV-NS instances, where both of them should support the information related to the MEC applications. In particular:

- The Vs-So(-LCM) reference point is implemented through the **ETSI GS NFV-IFA 013 [23] NFV-NS Lifecycle Management Interface**. This interface provides messages for creation and deletion of network service identifiers as well as instantiation, scaling, update, querying and termination of a network service instance;
- The Vs-So(-MON) reference point is implemented through the **ETSI GS NFV-IFA 013 [23] NFV-NS Performance Management Interface** and the **ETSI GS NFV-IFA 013 [23] NFV-NS Fault Management Interface**. The Performance Management Interface provides messages for creating, deleting and querying Performance Monitoring jobs (PMON jobs) as well as subscriptions and notifications to receive monitoring reports. The Fault Management Interface provides subscriptions, notifications and queries about alarms related to network services and VNFs;
- The Vs-So(-CAT) reference point is implemented through the **ETSI GS NFV-IFA 013 [23] NSD Management Interface**, the **ETSI GS NFV-IFA 013 [23] VNF Package Management Interface** and the **ETSI MEC 10-2 Application Package Management Interface** [27], where the 5GT-SO implements the role of the Mobile Edge Orchestrator (MEO). Each of them provides mechanisms to on-board, enable, disable, update, delete and query NSDs (ETSI GS NFV-IFA 013 [23]), PNFDs (ETSI GS NFV-IFA 013 [23]), VNF Packages (ETSI GS NFV-IFA 013 [23]) and MEC Application Packages (MEC 10-2), as well as subscriptions and notifications to notify new on-boarding actions or changes in existing descriptors.

NSD and VNF packages follow the information models defined in the ETSI GS NFV-IFA 014 [24] and ETSI GS NFV-IFA 011 [22] respectively, while the AppD follows the information model defined in ETSI MEC 10-2. However, in 5G-TRANSFORMER, the NSD is extended to include references to the AppDs, as shown in Figure 6 and Figure 7. Moreover, in order to specify the geographical location where the NFV network service should be placed, the instantiation request in the ETSI GS NFV-IFA 013 is extended with an additional “location constraint” attribute in the definition of the Service Access Points that connects the network service instance to the external networks.

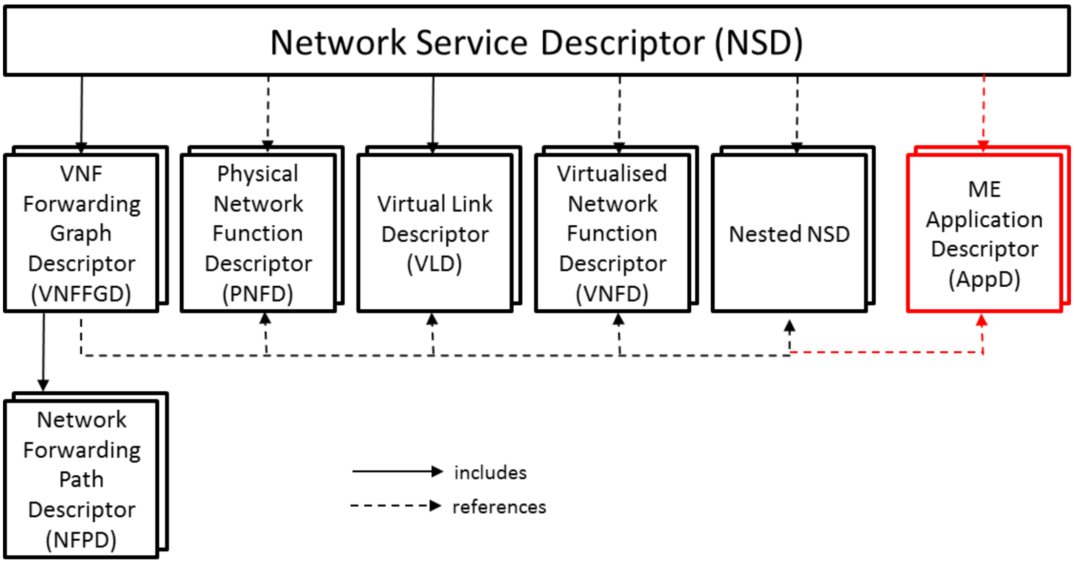

Network Service Descriptor (NSD)
VNF Forwarding Graph Descriptor (VNFFGD)
Physical Network Function Descriptor (PNFD)
Virtual Link Descriptor (VLD)
Virtualised Network Function Descriptor (VNFD)
Nested NSD
ME Application Descriptor (AppD)
Network Forwarding Path Descriptor (NFPD)
includes
references

FIGURE 6: NSD EXTENDED WITH APPD REFERENCES

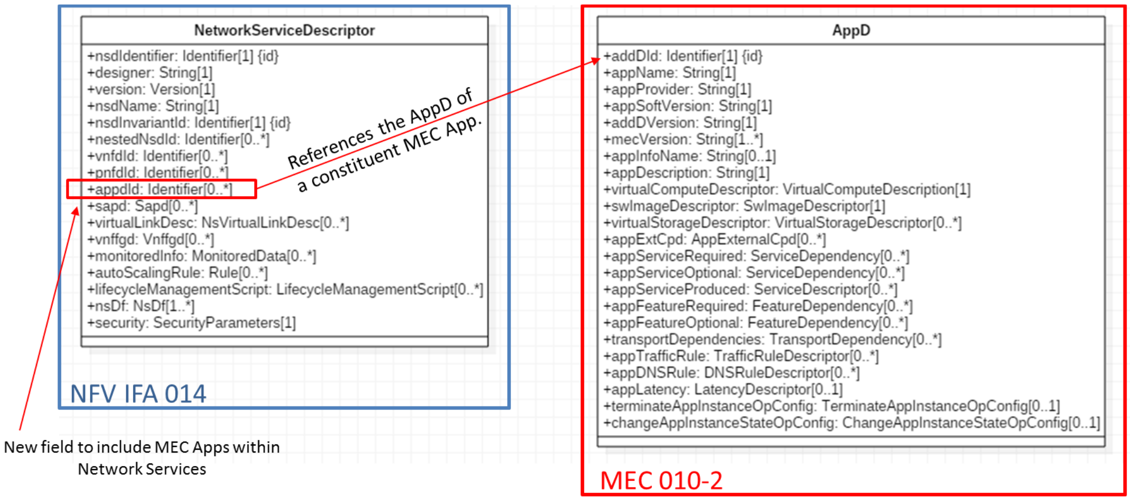

NetworkServiceDescriptor
+nsdIdentifier: Identifier[1] {id}
+designer: String[1]
+version: Version[1]
+nsdName: String[1]
+nsdInvariantId: Identifier[1] {id}
+nestedNsdId: Identifier[0..*]
+vnfdId: Identifier[0..*]
+pnfdId: Identifier[0..*]
+appdId: Identifier[0..*]
+sapd: Sapd[0..*]
+virtualLinkDesc: NsVirtualLinkDesc[0..*]
+vnffgd: Vnffgd[0..*]
+monitoredInfo: MonitoredData[0..*]
+autoScalingRule: Rule[0..*]
+lifecycleManagementScript: LifecycleManagementScript[0..*]
+nsDf: NsDf[1..*]
+security: SecurityParameters[1]
NFV IFA 014
References the AppD of a constituent MEC App.
New field to include MEC Apps within Network Services
AppD
+addDId: Identifier[1] {id}
+appName: String[1]
+appProvider: String[1]
+appSoftVersion: String[1]
+addDVersion: String[1]
+mecVersion: String[1..*]
+appInfoName: String[0..1]
+appDescription: String[1]
+virtualComputeDescriptor: VirtualComputeDescription[1]
+swImageDescriptor: SwImageDescriptor[1]
+virtualStorageDescriptor: VirtualStorageDescriptor[0..*]
+appExtCpd: AppExternalCpd[0..*]
+appServiceRequired: ServiceDependency[0..*]
+appServiceOptional: ServiceDependency[0..*]
+appServiceProduced: ServiceDescriptor[0..*]
+appFeatureRequired: FeatureDependency[0..*]
+appFeatureOptional: FeatureDependency[0..*]
+transportDependencies: TransportDependency[0..*]
+appTrafficRule: TrafficRuleDescriptor[0..*]
+appDNSRule: DNSRuleDescriptor[0..*]
+appLatency: LatencyDescriptor[0..1]
+terminateAppInstanceOpConfig: TerminateAppInstanceOpConfig[0..1]
+changeAppInstanceStateOpConfig: ChangeAppInstanceStateOpConfig[0..1]
MEC 010-2

FIGURE 7: NSD AND APPD INFORMATION MODELS

4.5 5GT-SO SBI

We next summarize the design of the 5GT-SO’s southbound interface (SBI); additional details can be found in [11]. The SBI addresses the interworking between the 5GT-SO and 5GT-MTP building blocks of the 5G-TRANSFORMER architecture [11]. It is worth mentioning that 5GT-SO and 5GT-MTP may follow a 1:N relationship. That is, a single 5GT-SO may interact via multiple SBI instances towards N 5GT-MTPs which handle the configuration and programmability of a number of domains including heterogeneous virtualized resources for compute, storage and networking. In the following we are also assuming that a 5GT-MTP is managed by a single 5GT-SO. Besides managing the utilization (i.e., de/allocation) of the virtualized resources, the 5GT-SO SBI also encompasses the required functionalities for deploying (updating and terminating) demanded VNFs by a given NFV-NS. In the 5G-TRANFORMER project all these operations are supported by the so-called So-Mtp interface.

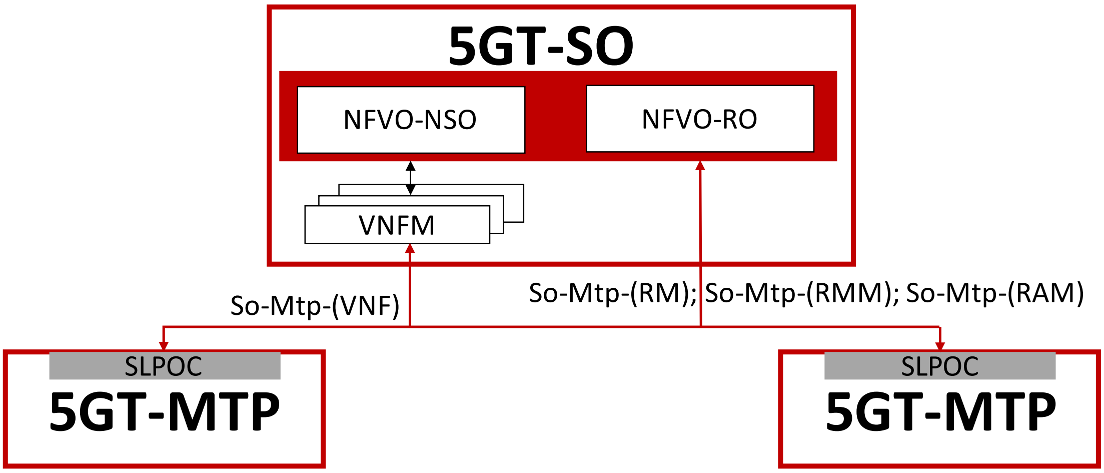


FIGURE 8: REFERENCE POINTS FOR 5GT-SO SBI (I.E., SO-MTP INTERFACE)

Figure 8 illustrates the targeted 5GT-SO SBI and its key reference points. Similar to the 5GT-SO NBI, the 5GT-SO SBI is mostly based on a set of standard documents being produced within the ETSI NFV framework, namely ETSI GS NFV-IFA 005 [17], ETSI GS NFV-IFA 006 [18] and ETSI GS NFV-IFA 008 [20]. In a nutshell, the 5GT-SO SBI shall provide the operations and functions, supported by a well-defined set of messages and workflows, for: (i), providing abstracted information (e.g., capacities, availability, connectivity, etc.) of the virtualized resources managed by each 5GT-MTP; (ii) managing (i.e., instantiation, reservation, allocation, scaling up/down and release) of the virtualized resources required to support an NFV-NS; (iii) enabling the fault management and performance monitoring aiming at recovering interrupted services or ensuring the targeted SLAs demanded by each NFV-NS; and (iv) supporting the lifecycle management (i.e., creation, configuration, modification and termination) along with related performance and fault management of the VNFs instantiated over the virtualized (compute and storage) resources. As shown in the figure, thus the So-Mtp interface enables communicating the specific entities of the 5GT-SO (VNFM and NFVO-RO) with a single logical point of contact (SLPOC) at each 5GT-MTP entity. Accordingly, four reference points for the 5GT-SO SBI are conceived:

- **So-Mtp(-Resource Advertising Management).** It provides the Resource Advertisement Management functions. That is, it allows feeding the 5GT-SO's NFVI repository with information regarding the virtualized resources that will accommodate requested NFV-NSs. Such information, modelled in 5G-TRANSFORMER D2.1 [11], can be delivered using different levels of details/abstraction. Thus, the adopted abstraction, and vision of the resources, will notably impact on the 5GT-SO NFVO-RO algorithms used for the VNF placement and/or networking computation. The mechanism used by the 5GT-MTP(s) to update the 5GT-SO's NFVI could be achieved also via different mechanisms such as immediate update when a change in any (abstracted) virtualized resource occurs (e.g., allocation or reservation), upon an explicitly demand sent by the 5GT-SO, or even applying predefined periodic updates.

- **So-Mtp(-R**esource **M**anagement**).** This encompasses the Resource Management operations over the virtualized resources. Basically, it contains the set of operations used for reserving, allocating, updating (in terms of scaling up or down) and terminating (i.e., release) the resources handled by each 5GT-MTP. In short, and according to the abstracted information managed by the 5GT-SO, the So-Mtp(-RM) coordinates the involved 5GT-MTPs to manage the utilization of a selected set of virtualized resources. This entails the VNF placement and triggering the reservation / allocation of the network resources constituting the demanded NFV-NS.
- **So-Mtp(-R**esource **M**onitoring **M**anagement**).** This provides the Resource Monitoring Management operations. Basically, it provides the required interworking procedures including the primitives and parameters for supporting the 5GT-SO Monitoring Service capability. This entails a twofold functionality: i) fault management to recover / restore the interrupted NFV-NS by a failure (e.g., link failure, VNF host crashes, etc.); ii) permanent performance monitoring crucial to ensure the demanded SLA for the existing NFV-NS. Obviously, the involved information needed to such Monitoring Service functions are also related with the adopted granularity and abstracted information (i.e., level of detail) within the 5GT-SO.
- **So-Mtp(-VNF).** This takes over of the general VNF lifecycle management (e.g., scaling up/down a particular VNF instance, fixing VNF malfunctions, etc.) commanded by the 5GT-SO VNFM. Moreover, the VNF configuration (i.e., specifying targeted parameters defining the targeted VNF behaviour) is also supported over this reference point. Last but not least, the So-Mtp-VNF reference point supports performance and fault management functionalities to monitor the VNF operations and adopt/apply (if needed) necessary actions to revert/solve VNF failures and performance degradations.

As anticipated above, the implementation of the 5GT-SO SBI operations and, particularly, those driven into the four reference points, leverage the procedures (i.e., interworking between entities, messages and basic contents) described in ETSI GS NFV-IFA 005, 006 and 008 [17], [18], and [20], resp., as much as possible. Note that other functions/deviations required to be covered within the 5G-TRANFORMER framework but not supported by those standard documents could be eventually added. Nonetheless, focusing exclusively on the operations currently supported in [17], [18], and [20], the following operations are identified as essential for encompassing the implementation of the 5GT-SO SBI reference points:

- For the **So-Mtp(-RAM),** a set of pairs of request/response messages is considered. This is divided into two main sets/groups, namely, Virtualized Resources Information Management and Virtualized Resources Capacity Management. These two subsets of messages (see [17] and [18]) grant 5GT-SO with multiple functionalities such as subscription to specific (filtered) resource information, query of and update (for changes) resource information, specific resource capacity, etc. Specifically, the resource capacity can be provided with respect to its current status (i.e., available, allocated or reserved) as well as specifying the amount of resources. Moreover, the resource information can be retrieved for a particular 5GT-MTP governing a certain Resource Zone (e.g., geographic NFVI-PoP specifying the reachable endpoints). With respect to the amount of resources, this clearly depends not only on the type of virtualized resource but also on the abstraction policy. An abstraction model is planned (for the sake of scalability and/or confidentiality purposes) to keep track of selected and relevant information about the virtualized resources (i.e., compute, storage and networking) without having a complete awareness of all the features and capabilities intrinsic to such resources. As described in [11], this abstraction may rely on basically knowing the maximum, available and allocated amount of each resource type, or having a summarized topology view enabling basic connectivity between remote network elements hiding details of the underlying transport infrastructure supporting such connectivity. In this regard, for instance, for the compute resources, it can be delivered the total amount of available or allocated virtual memory and virtual CPU; for storage, the information is related to the size of storage and type (e.g., volume, object) or the support of remote direct memory access; for networking resources, the provided attributes regarding the link type (e.g., VLAN or GRE), the supported link QoS parameters (e.g., latency), the total link bandwidth, the IP addressing for a specific (sub-)network, the port types connected to specific network elements (e.g., Layer 1, 2 or 3), etc.

- For the **So-Mtp(-RM)**, in ETSI GS NFV-IFA 005 [17], sets of request/response messages to allocate, query, update, migrate, and terminate virtualized resources are specified. Such defined sets are tailored to the operations to be made over a particular virtualized resource set. For instance, for compute resources the set named Virtualized Compute Management Interface is defined. This provides the specific allocation of compute resource over a particular Resource Zone, with a determined set of virtual CPUs and memory, as well as informing about the software image to be set on the Virtual Machine. For network resources, the set Virtualized Network Resource Management Interface takes over of all the operations to be made over the network resources. Non-exhaustively, this includes the allocation of selected bandwidth over a network entity (e.g., link) or the utilization of an entire data port. In addition, the operations to create Network Forwarding Paths (NFPs) to accommodate VNFFGD are supported. The request messages should include the list of virtual networks and ports forming the NFP. In addition, the top Virtualized Storage Resource Management Interface entails the set of operations (mapped to pair of messages) handling the storage resources. Similar to the compute and networking resources, these operations enable the selection of an amount of storage to be allocated over, e.g., a particular Resource Zone. Finally, ETSI GS NFV-IFA 005 [17] also defines a top interface to reserve virtualized resources referred to as Virtualized Resource Reservation Interfaces. This interface is used to (pre-)book virtualized resources which eventually may be needed and used.
- For the **So-Mtp(-RMM)**, ETSI GS NFV-IFA 005 [17] also specifies the interfaces (messages and contents) supporting fault management and performance monitoring. Specifically, the Virtualized Resource Fault Management Interface

defines the messages enabling the 5GT-SO to subscribe for notifications from 5GT-MTP about containers or virtual machines crashes, virtual network port errors or reserved resources unavailable or exhausted. To this end, such interface supports detailed alarms. The Virtualized Resources Performance Management Interface describes a set of messages used for collecting measurements within notifications that will feed the 5GT-SO's Monitoring Service. These messages include resource consumption, memory oversubscription, disk latency, etc. In general, the collection of such information is controlled by a Performance Monitoring (PMON) job. The interface is oriented on handling the management of PMON jobs (creation, subscribe, update, query, etc.). For a given PMON job, it can be specified the object to be monitored (e.g., CPU power consumption in VM), the performance metric, the frequency for capturing the measurements, threshold to send notifications, etc.

- For the **So-Mtp(-VNF)**, ETSI GS NFV-IFA 008 [20] describes the messages and contents supporting the operations for the creation / configuration / termination, scaling (up / down), monitoring and fault management of the VNFs being deployed in a specific NFVI-PoP and handled by the SO-MTP. In this context, [20] firstly addresses the necessary set of messages (as request/respond pairs) used for both initially configuring and modifying (e.g., deleting) a VNF (or Component). For the sake of completeness, this specific interworking is triggered by the 5GT-SO VNFM. In general, the messages providing a VNF operation must carry a unique identifier to unambiguously determine over which particular VNF (or Component) the action will be conducted. Moreover, configuration data or parameters are also included specifying the amount of required memory, CPU capacity, storage size, connection points (address and ports), software image of the VNF container, etc. That is, the set of parameters providing the description of the targeted VNF (Component) is referred to as VNF Descriptor (VNFD). In general, VNFDs are on-boarded in the so-called VNF package and they are assigned by an identifier which allows determining which descriptor is followed by a VNF instance being created. Exhaustive details about the VNFDs and the configurable parameters are provided in ETSI GS NFV-IFA 011 [22]. Another interworking operation supported by **So-Mtp(-VNF)** is the VNFM Indication actions. These indicators are used to notify the 5GT-SO VNFM about a VNF behaviour which can be eventually used by the VNFM to trigger auto-scaling operations. Finally, the performance and fault management interworking enable creating PMON jobs to impose the generation of notifications sending specific VNF parameters status (e.g. CPU) which are then gathered and processed. To this end, the interface entails the creation of thresholds to manage the notification message creation or the periodicity when such notifications are actually composed and sent. Last but not least, for the fault management purposes, the 5GT-SO VNFM is able to indicate the subscription demanding for specific alarms generated by the VNFs when, for instance, to react when a fault occurs.

In brief, the above reference points constitute and provide the basic and required operations and functions to be encompassed over the 5GT-SO-SBI. That said, it is important to outline that such an interface (in terms of design and supported capabilities) will be explored and assessed recurrently according to the (potential new) needs arising from the targeted use cases studied within the project.

4.6 5GT-SO EBI/WB

5GT-SO eastbound/westbound interface (EBI/WBI) is the only interface in the 5G-TRANSFORMER architecture that provides federation of services and federation of resources. 5GT-SO EBI/WBI enables interaction between a local 5GT-SO and other SOs that are part of other administrative domains. This So-So interface enables 5GT-SO to request/offer NFV-NSaaS and NFVIaaS. The approach of the So-So interface is adopted from the SLPOC for federation (SLPOC-F) solution of the ETSI GS NFV-IFA 028 [26]. The SLPOC-F solution offers two modes of operation: direct and indirect. The difference is on the NFVIaaS use case. In the SLPOC-F direct mode the consumer VNFM can directly invoke resource management operations on the provider 5GT-SO NFVO-RO. In SLPOC-F indirect mode, the consumer VNFM uses the consumer 5GT-SO NFVO-RO as a proxy (or forwarding point) to invoke resource management operations on the provider 5GT-SO NFVO-RO. This section mostly focuses on the direct approach without discarding the use of indirect mode in the future. In that sense, the 5GT-SO EBI/WBI is based on six reference points (see Figure 9):

- **So-So(-L**ife **C**ycle **M**anagement**)**, used for the operation of NFV network services. The reference point is used to instantiate, terminate, query, update or re-configure network services or receive notifications for federated NFV network services. 5GT-SO's NFVO-NSO is using this reference point;
- **So-So(-MON**itoring**)**, used for the monitoring of network services through queries or subscriptions/notifications about performance metrics, VNF indicators and network service failures. 5GT-SO's NFVO-NSO is using this reference point;
- **So-So(-Catalogue)**, used for the management of Network Service Descriptors (NSDs) flavors and MEC Application Packages, including their Application Descriptors (AppDs). This reference point offers primitives for on-boarding, removal, updates, queries and enabling/disabling of descriptors and packages. 5GT- SO's NFVO-NSO is using this reference point;
- **So-So(-R**esource **M**anagement**),** used for the operation of performing allocation, configuration, updates and release of resources. The Resource Management reference point offers operations such as configuration of the resources, configuration of the network paths for connectivity of VNFs. These operations mainly depend of the level of abstraction applied to the actual resources. 5GT-SO's NFVO-RO is using this reference point;
- **So-So(-R**esource **M**onitoring **M**anagement**),** used for monitoring of different resources, computing power, network bandwidth or latency, storage capacity, VMs, MEC hosts provided by the peering administrative domain. The details level depends on the agreed abstraction level. 5GT-SO's NFVO-RO is using this reference point.
- **So-So(-R**esource **A**dvertising **M**anagement**),** used for advertising available resource abstractions to/from other SOs. Broadcast of available resources or resource abstractions upon capability calculation and periodic updates for near real-time availability of resources. The SO-SO Resource Advertisement is using this reference point.

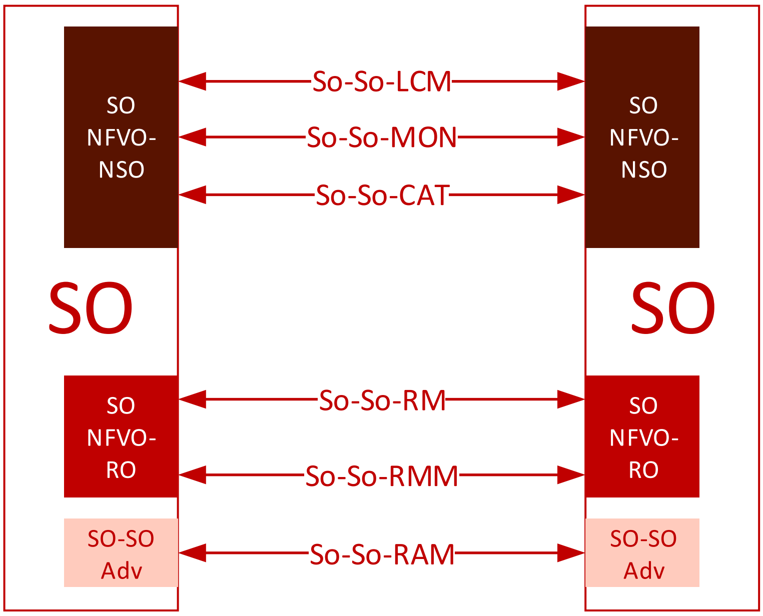


FIGURE 9: 5GT-SO EBI/WBI REFERENCE POINTS

The 5GT-SO EBI/WBI implementation is based on the ETSI GS NFV-IFA 013 [23], ETSI GS NFV-IFA 005 [17], ETSI GS NFV-IFA 006 [18] and ETSI GS NFV-IFA 008 [20]. NFV-NSaaS provides NFV network services through the reference points of the 5GT-SO NFVO-NSO mainly by implementing and extending interfaces of ETSI GS NFV-IFA 013 [23]. In particular:

- The So-So-LCM (Lifecycle Management) is implemented through the **ETSI GS NFV-IFA 013 [23] NFV-NS Lifecycle Management Interface**. The interface provides request messages for instantiation, creation, scaling, update, querying, termination and deletion of network services;
- The So-So-MON (Service Monitoring) is implemented through the **ETSI GS NFV-IFA 013 [23] NFV-NS Performance Management Interface** and the **ETSI GS NFV-IFA 013 NFV-NS Fault Management Interface**. They provide messages for monitoring of performances as well as notifications for alarms related to NFV-NSs;
- The So-So-CAT (Service Catalogue) is implemented through the **ETSI GS NFV-IFA 013 [23] NSD Management Interface** and the **ETSI MEC 10-2 Application Package Management Interface**. The reference point implements mechanisms for providing updates of flavors and querying NSDs (ETSI GS NFV-IFA 013 [23]) as well as on-boarding or applying changes of MEC Application Packages (MEC 10-2).

NFVIaaS provides resources or resource abstractions through the reference points of 5GT-SO NFVO-RO and SO-SO Resource Advertising block, generally by implementing interfaces of ETSI GS NFV-IFA 005/006/008 [17][18]. In particular:

- The So-So-RM (Resource Management) implementation is based on the **ETSI GS NFV-IFA 005/006 [17][18] Software Image Management Interface**, **ETSI GS NFV-IFA 005/006 [17][18] Virtualized Compute/ Network/Storage Management Interface**, **ETSI GS NFV-IFA 008 [20] VNF Lifecycle Management**, **ETSI GS NFV-IFA 008 [20] VNF Configuration Management**. The implementation is limited and directly dependable of the abstraction level and federation level. Operations for managing VNFs or VMs, software images as well as basic operation and management of allocated compute/network/storage resources;

- The So-So-RMM (Resource Monitoring Management) implementation is based on **ETSI GS NFV-IFA 005/006 [17][18] Virtualized Resource Performance Management Interface, ETSI GS NFV-IFA 005/006 [17][18] Virtualized Resource Fault Management Interface, ETSI GS NFV-IFA 008 [20] VNF Lifecycle Change Notification and ETSI GS NFV-IFA 008 [20] VNF Performance & Fault Management**. The limited implementation provides information about the performance metrics and monitoring reports of the allocated federated resources or abstractions as well as alarms or notifications of resource failures or VNFs;
- The So-So-RAM (Resource Advertising Management) implementation is based **on ETSI GS NFV-IFA 005/006 [17][18] Virtualized Resource Quota Interfaces**. The implementation is limited to the query requests and broadcasting updates with purpose of advertising available resources or resource abstractions.

It is important to note that each connection on the 5GT-SO EBI/WBI (between two peering SOs) has different limitations (e.g. advertising and/or monitoring storage capacity, without vendor indicator or type of storage parameters, etc.)on each reference point due to hiding details of the NFV-NSs or abstractions of federated NFVI resources. These limitations depend of the business agreement levels and policies applied by each administrative domain. 5GT-SO EBI/WBI contains similar functionalities enabled through NBI and SBI with expected limitations.

4.7 Service Monitoring

The 5GT-SO Monitoring Service is in charge of producing monitoring reports related to the performance or to failure events associated to the managed NFV network services and their VNFs. The generated monitoring reports can be used internally at the 5GT-SO, for example to validate SLAs or as triggers to auto-scaling procedures, according to the auto-scaling rules defined in the NSDs of the instantiated network services. Moreover, the monitoring reports produced by the 5GT-SO Monitoring Service can be also provided to the 5GT-VS through the Vs-So(-MON) reference point (see Section 4.4 for details).

The 5GT-SO Monitoring Service collects elementary monitoring data from different sources and correlates or aggregates them to global monitoring reports about the logical entities managed by the 5GT-SO, i.e. NFV network services and VNFs. The sources of elementary monitoring data for the 5GT-SO Monitoring Service are the following:

- Local 5GT-MTP Monitoring Service, for collecting performance metrics or failure events about the virtualised resources (i.e. virtual computes, virtual networks, virtual storages) that compose the managed VNFs and network services.
- 5GT-SO Monitoring Service of federated 5GT-SOs, for collecting performance metrics or failure events about (i) nested network services or VNFs or (ii) virtualized resources provided by the federated 5GT-SO, in the NFV-NSaaS or

in the NFVIaaS case respectively (see service and resource federation in section 4.3.3 and section 4.3.4).

- VNFM(s) or Physical Network Function Managers (PNFMs) for collecting VNF/PNF indicators produced by the managed VNFs/PNFs, or their Element Managers (EMs).

Examples of performance metrics that can be elaborated at the 5GT-SO Monitoring Service are the amounts of vCPU or RAM consumed in a time interval by a specific VNF or by an entire network service, the number of packets lost on a virtual link, or VNF-specific indicators. The rules to correlate the elementary data received by the monitoring sources (e.g. vCPU consumption in single VMs as received by the 5GT-MTP Monitoring Service) are embedded in the monitoring algorithms. These rules allow to obtain the performance record for entire VNFs or NFV network service instances, as specified for the different metrics encoded in the NSDs. Section 5.5 provides an example of a workflow to collect monitoring data from the 5GT-MTP and consolidate them in 5GT-SO performance metrics that can feed procedures for SLA validation.

In detail, the scope of the 5GT-SO Monitoring Service is focused on producing “aggregated” monitoring data that provides relevant information about the status and the performance of a number of logical entities, somehow correlated with the end-to-end NFV-NS instances managed by the 5GT-SO itself. Starting from these monitoring data, the 5GT-SO will take decisions about the lifecycle or the resource orchestration of the NFV-NS instances. In particular, the following logical entities have an impact on the end-to-end NFV-NS instances and, consequently, should be monitored by the 5GT-SO Monitoring Service:

- Virtual resources instantiated from the 5GT-SO in its local domain through the 5GT-MTP.
- Virtual resources from federated 5GT-SO (e.g. VMs, connection services between VMs, etc.) in the NFVIaaS case.
- VNFs, VAs and (nested) NFV Network Service instances managed by the 5GT-SO itself, as well as the ones requested to federated 5GT-SOs in the NFV-NSaaS case.
- PNFs used in scope of the NFV-NS instantiated by the 5GT-SO.
- As option, the 5GT-MTP may expose to the 5GT-SO a subset of monitoring data related to the physical resources, for example to enable more efficient resource allocation decisions at the 5GT-SO. This is typically the case where 5GT-SO and 5GT-MTP are managed by the same administrative entity.

The 5GT-SO Monitoring Service APIs should be compliant with the abstract messages defined in the ETSI GS NFV-IFA 013 reference point [23], with particular reference to the “NS Performance Management Interface” and the “NS Fault Management Interface”. As analyzed in Section 4.4, the Performance Management Interface provides messages for creating, deleting and querying Performance Monitoring jobs (PMON jobs) as well as subscriptions and notifications to receive monitoring reports. The Fault Management Interface provides subscriptions, notifications and queries about alarms related to NFV network services’ and VNFs’ failures. Subscriptions and notifications can be also based on thresholds, in order to receive events only when a specific condition occurs. This is particularly useful to efficiently trigger NFV network service lifecycle commands, like scaling actions, to automatically react in case of performance degradation or SLA violation. Moreover, the ETSI GS NFV-IFA 013 [23] messages includes several parameters to customize the desired behaviour of the monitoring platform, for example in terms of filtering and attributes selection in the queries, threshold specification, collection and reporting periods for monitoring threads.

It should be noted that the 5GT-SO Monitoring Service APIs are also used in the interaction between federated 5GT-SOs. As described in section 4.6, in this case the APIs are used not only for monitoring of NFV network services and VNFs (at the So-So-MON reference point), but also for the monitoring of virtual resources provided by the peering administrative domain (at the So-So-RMM reference point).

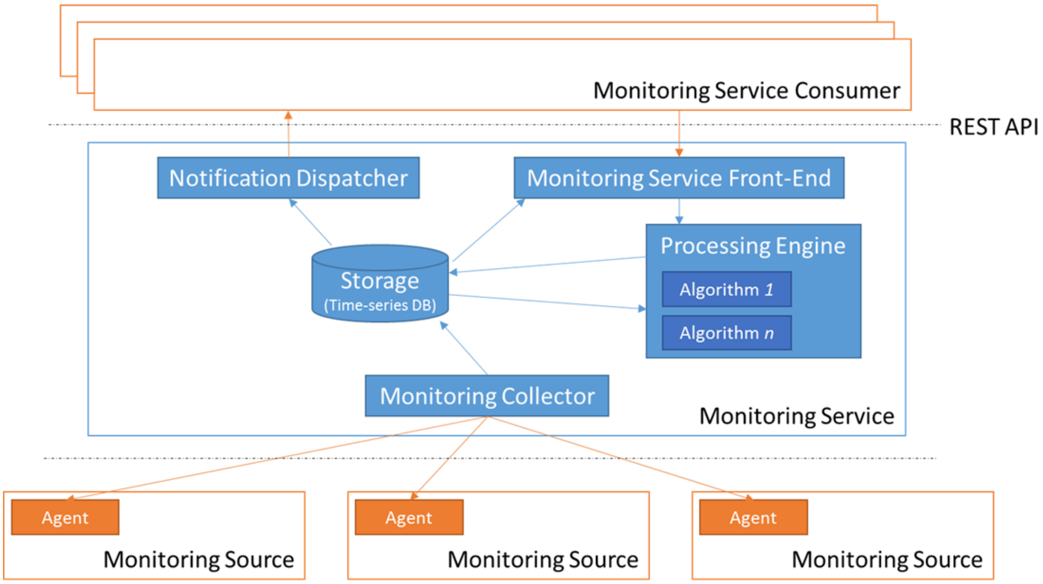


FIGURE 10: FUNCTIONAL ARCHITECTURE OF THE 5GT-SO MONITORING PLATFORM

Figure 10 shows the functional architecture of the 5GT-SO Monitoring Platform, which can be easily adapted also to the 5GT-VS and 5GT-MTP Monitoring Platforms. At the southbound, the monitoring collector retrieves monitoring data from different monitoring sources through dedicated agents that translates between different information models, monitoring mechanisms or message protocols. In the 5GT-SO case, a number of interfaces are adopted to collect information from the different monitoring sources, as follows:

- The 5GT-MTP monitoring platform exposes monitoring data from VIMs and WIMs about virtual resources and, optionally, physical resources. This interface is based on the ETSI GS NFV-IFA 005 specification [17], which defines the reference points between NFVO and VIM.
- The federated 5GT-SOs' monitoring platforms expose monitoring data about nested network services or VNFs and virtual resources, depending on the type of service. In the NFV-NSaaS case the interface is based on the ETSI GS NFV-IFA 013 specification [23], while in the NFVIaaS case the interface is based on the ETSI GS NFV-IFA 005 [17] specification.

- The VNFMs expose VNF indicators collected by the VNFs or their EMs or VNFs' performance metrics. In this case the specific interface may vary with the target VNF. For example, it might be based on the ETSI GS NFV-IFA 007 specification [19], which defines the reference points between NFVO and VNFM, or, in case of proprietary VNFMs, proprietary protocols may be adopted.

The elementary monitoring data collected by these sources is then stored in the internal storage, typically a time-series DB to enable more efficient queries. These data can be used as input for further elaboration at the processing engine, which aggregates and correlates elementary data using specialized algorithms, according to the rules and filters specified by the monitoring service consumers. The post-processing data are also stored in the internal DB, so that they are made available for the consumers, which can retrieve them through queries (through the Monitoring Service Front-End) or notifications (through the Notification Dispatcher).

5 5GT-SO Workflows

In this section we detail the workflows incurred by a selected set of procedures supported by 5G-SO, namely: (1) Service On-boarding, (2) Service Instantiation, (3) Service Modification, (4) Service Termination, (5) Service Assurance, (6) Service Federation, (7) Resource Federation, and (8) NFV-NS instantiation when including MEC applications.

5.1 Service On-boarding

Description: The workflow describes the on-boarding of VNFs (initiated by the 5GT-SO admin), VAs (initiated by the vertical) and network services (initiated by the 5GT-VS). The same procedure applies for on-boarding MEC applications.[4]

Prerequisites: None.

Assumptions: The vertical service can be deployed in one NFV-NS. Also, the service is deployed in one new NFV-NSI. We assume the mapping between vertical service and the NSD has been already performed.

Workflow:

This workflow includes three parts that are initiated at separate times by different actors, as set forth below.

The first part of the workflow describes the on-boarding of VNFs:

- The process is triggered by the 5GT-SO administrator or operator, requesting the on-board to the 5GT-SO NFVO-NSO (1) and providing the VNF package - including the VNFD as well as additional configuration files - as an input;
- The 5GT-SO NFVO-NSO requests the VNF package (2) from the software provider[5], which replies (3) by sending the VNF package;
- The 5GT-NFVO-NSO extracts the VNFD from the VNF package (4) and stores it in its VNFD/AppD catalogue (5-7);
- The 5GT-NFVO-NSO can confirm the successful on-boarding to the 5GT-SO admin[6] (8).

The second part of the workflow describes the on-boarding of VAs:

- The vertical sends to the 5GT-VS an on-board request (9), including the vertical application (VA) package (see [12] for more details);
- The 5GT-VS sends to the 5GT-SO NFVO-NSO a request to on-board the VA package, including the VNFD (10), to which the 5GT-SO NFVO-NSO replies with a request for the VA package itself (11). The 5GT-VS replies (12) by sending the VA package, including the VNFD/AppD as well as additional configuration files;

[4] MEC applications are described using the AppD as specified by ETSI MEC [43]. The AppD has similar information as the VNFD, but with specific fields reflecting the MEC applications requirements (e.g., traffic redirection, latency requirement, required MEC service, etc.). In 5G-TRANSFORMER we assume that the NSD is extended to integrate AppD(s).

[5] A software provider is an entity that assists 5G-TRANSFORMER's service provider with VNFs.

[6] The 5GT-SO admin is the entity in charge of the management of 5G-TRANSFORMER service orchestrator.

- The 5GT-SO NFVO-NSO extracts the AppD from the VA package (13), and then stores the AppD into the 5GT-SO VNFD/AppD catalogue (14-16);
- The 5GT-SO NFVO-NSO confirms (17) the successful uploading to the 5GT-VS. Then the 5GT-VS sends a response (18) to the original on-board request from the vertical.

The third part of the workflow describes the on-boarding of network services:

- The 5GT-VS sends an on-boarding request to the 5GT-SO NFVO-NSO (19), including the NSD info;
- The 5GT-SO NFVO-NSO checks with the 5GT-SO VNFD/AppD catalogue that all the VNFs referenced in the NSD are already correctly stored (20-22);
- The 5GT-SO NFVO-NSO stores the NSD in the 5GT-SO NSD catalogue (23-25);
- The 5GT-SO NFVO-NSO replies with a success message (26) to the original request from the 5GT-VS.

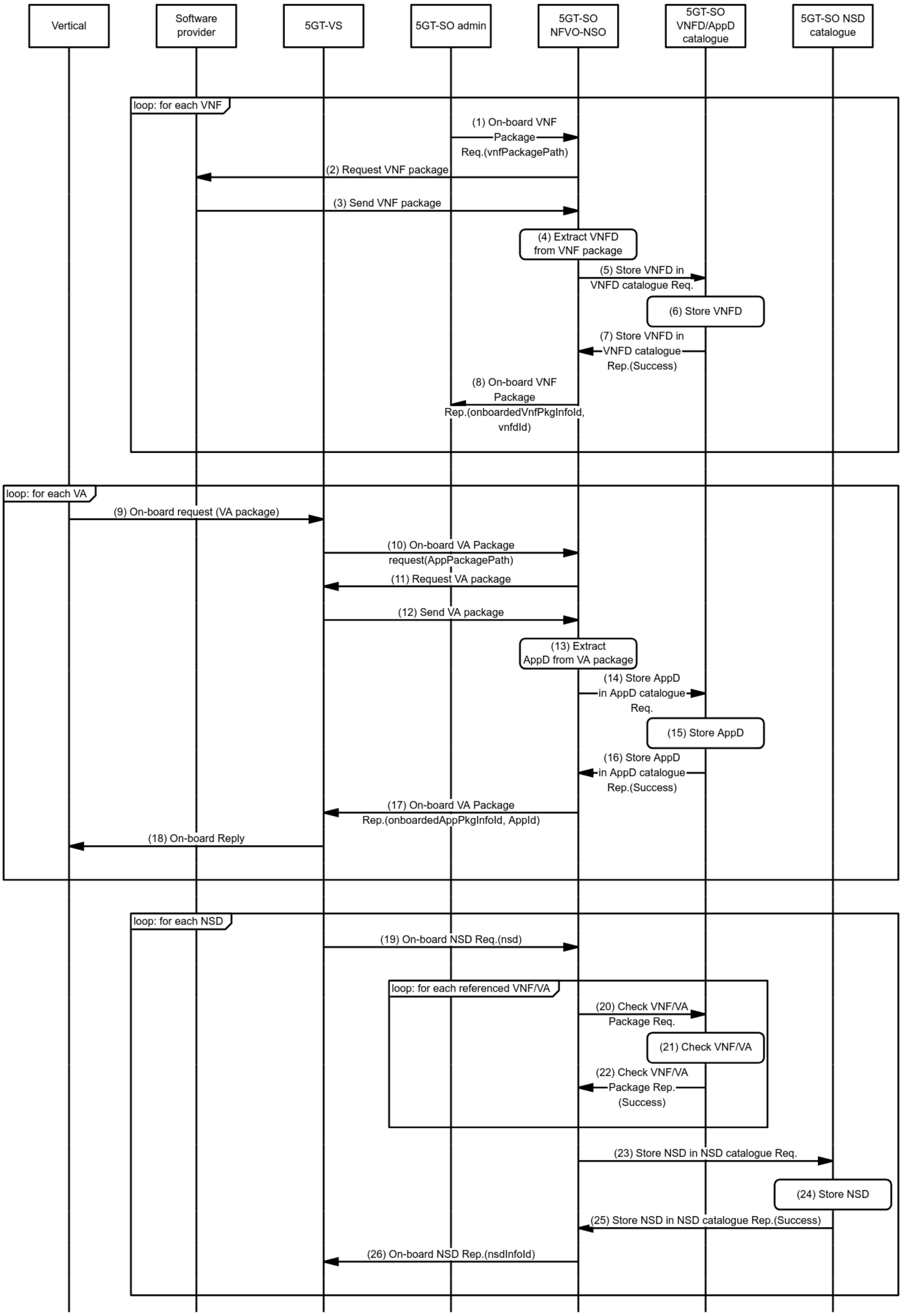

Vertical
Software provider
5GT-VS
5GT-SO admin
5GT-SO NFVO-NSO
5GT-SO VNFD/AppD catalogue
5GT-SO NSD catalogue
loop: for each VNF
(1) On-board VNF Package Req.(vnfPackagePath)
(2) Request VNF package
(3) Send VNF package
(4) Extract VNFD from VNF package
(5) Store VNFD in VNFD catalogue Req.
(6) Store VNFD
(7) Store VNFD in VNFD catalogue Rep.(Success)
(8) On-board VNF Package Rep.(onboardedVnfPkgInfoId, vnfdId)
loop: for each VA
(9) On-board request (VA package)
(10) On-board VA Package request(AppPackagePath)
(11) Request VA package
(12) Send VA package
(13) Extract AppD from VA package
(14) Store AppD in AppD catalogue Req.
(15) Store AppD
(16) Store AppD in AppD catalogue Rep.(Success)
(17) On-board VA Package Rep.(onboardedAppPkgInfoId, AppId)
(18) On-board Reply
loop: for each NSD
(19) On-board NSD Req.(nsd)
loop: for each referenced VNF/VA
(20) Check VNF/VA Package Req.
(21) Check VNF/VA
(22) Check VNF/VA Package Rep. (Success)
(23) Store NSD in NSD catalogue Req.
(24) Store NSD
(25) Store NSD in NSD catalogue Rep.(Success)
(26) On-board NSD Rep.(nsdInfoId)

FIGURE 11: SERVICE ON-BOARDING WORKFLOW

5.2 Service instantiation

Description: The workflow describes the instantiation of a vertical service, triggered by the vertical.

Prerequisites: The vertical has prepared a vertical service description and the 5GT-VS sends a service instantiation requests to the SO.

Assumptions: The vertical service is a simple one, meaning it can be deployed on one NFV network service instance. We also assume that this service is deployed in a new NFV network service instance. We assume that (1) VSD has been on-boarded into the 5GT-VS; (2) and the NSD including its associated VNFDs/AppDs, to which the vertical service is mapped to and which describes the network service, has been previously on-boarded to the 5GT-SO.

Workflow:

- The vertical triggers the workflow by requesting the instantiation of a vertical service, identified by an identifier for the VSD (01).
- Before the workflow proceeds further, the vertical is authenticated and its authorization is verified by the system (i.e., 5GT-VS, 5GT-SO and 5GT-MTP) (02).
- Assuming the authentication process succeeds, the 5GT-VS creates an entry with an ID for the specific instance of this vertical service (VSI) and stores it in its repositories (03). Note, this is internal bookkeeping of the 5GT-VS, the vertical service instance is not deployed yet.
- The vertical is informed about the ID of this vertical service instance (04).
- At next, the 5GT-VS maps the VSD to an NSD[7] and checks whether a new instance of it would be possible according to the resource budget of the vertical (05).
- If the resource budget has not been already exhausted, the 5GT-VS requests a new NFV network service identifier from the 5GT-SO (06). Note, we assume that the vertical service instance is mapped to a network slice instance, which is deployed as a NFV network service instance described by a NSD already known or on-boarded to the 5GT-SO[8].
- Upon receiving “Create NFV NFV-NS Id Request” from the 5GT-VS, the NFVO-NSO running in the 5GT-SO firstly performs a formal check (e.g., validating the integrity) of received NSD within the NFV-NS/VNF Catalogue DB (07 and 08).
- If the formal check succeeds, the NFVO-NSO creates a new ID for the requested NFV NFV-NS instance with the associated instance of an *NsInfo* information element (09), and then it will add a new entry in the NFV-NS instance repository (10).
- The NFVO-NSO responds to the 5GT-VS with the created *nfv_nsinstanceId* (11).

[7] We assume a 1:1 relationship in this example though 5G-TRANSFORMER will support different relationships.

[8] In more complex cases, a new network service descriptor could be created and would have to be onboarded to the 5GT-SO first, or an existing network service descriptor would have to be modified and the 5GT-SO would have to be updated at the 5GT-SO.

- The 5GT-VS requests to the 5GT-SO instantiating this NFV-NS (i.e. instantiation of the vertical slice instance) (12).
- The NFVO-NSO performs a formal check of the requested NFV-NS Instance (NFV-NSI) (13 and 14), i.e., validating the created NFV-NSI within the NFV-NS/VNF instance repository.
- If the formal check succeeds, the NFVO-NSO seeks for available resources stored in the NFVI repository within the 5GT-SO entity through the NFVO-RO (15, 16, 17 and 18).
- Based on the returned NFVI resources information and requirements of the NFV NSI (according to NSD and its deployment flavor), the NFVO-NSO and NFVO-RO make optimized decision on the orchestration of NFV-NS, the placement of involved VNFs and their virtual resource allocation (19).
- Next, for each VNF, the NFVO-RO entity inside the 5GT-SO sends “resource allocation request” messages to the 5GT-MTP to ask for the actual allocation of resources inside the 5GT-MTP. Such requests include the information for the resources (i.e., storage, compute and virtual network resources) to be allocated as well as the ID of the VNF to be instantiated. After the successful allocation of the resources and VNFs, the 5GT-MTP will notify that to the NFVO-RO inside the 5GT-SO with their IDs (20, 21, 22). Afterwards, the 5GT-SO updates the NFVI resource repository accordingly (23). Alternatively, the 5GT-SO may also request an update of the resource abstraction from the 5GT-MTP with certain mechanisms, e.g., via periodical polling.
- The NFVO-NSO requests the VNFM to instantiate all corresponding VNFs included in this NSD. For each VFN in the NSD, the VNFM needs to firstly create a VNF instance Identifier (*vnfInstanceId*) followed by adding a new entry in the VNF instance repository (24, 25 and 26). After that, the VNFM instantiates the VNF and updates its status in the VNF instance repository (27, 28 and 29). Finally, the VNFM returns the identifier of the VNF lifecycle operation occurrence (VNF Lcid) (30).
- Once all VNFs are instantiated, the NFV-NSI is also instantiated. Then, the NFVO-NSO notifies the 5GT-VS about the identifier of the NFV-NS lifecycle operation occurrence (NS Lcid) (31).
- Once the NFV network service instance has been instantiated on the actual infrastructure, the 5GT-SO (NFVO-NSO) updates its records of instantiated NFV network service instances in the NFV-NS instance repository (32).
- The NFVO-SO informs the 5GT-VS (33) about the NFV NFV-NS instantiation. After that, the 5GT-VS updates its record of instantiated vertical services and network slice instances (34).
- Eventually, the vertical is notified[9] about the instantiation of the requested vertical service (35).

[9] It will be decided in the implementation phase whether notification is an operation initiated by 5GT-VS and consumed by the vertical or whether the vertical periodically polls the 5GT-VS. In case the interface between vertical and 5GT-VS is implemented as a GUI, this notification might just be a graphical indication in the GUI.

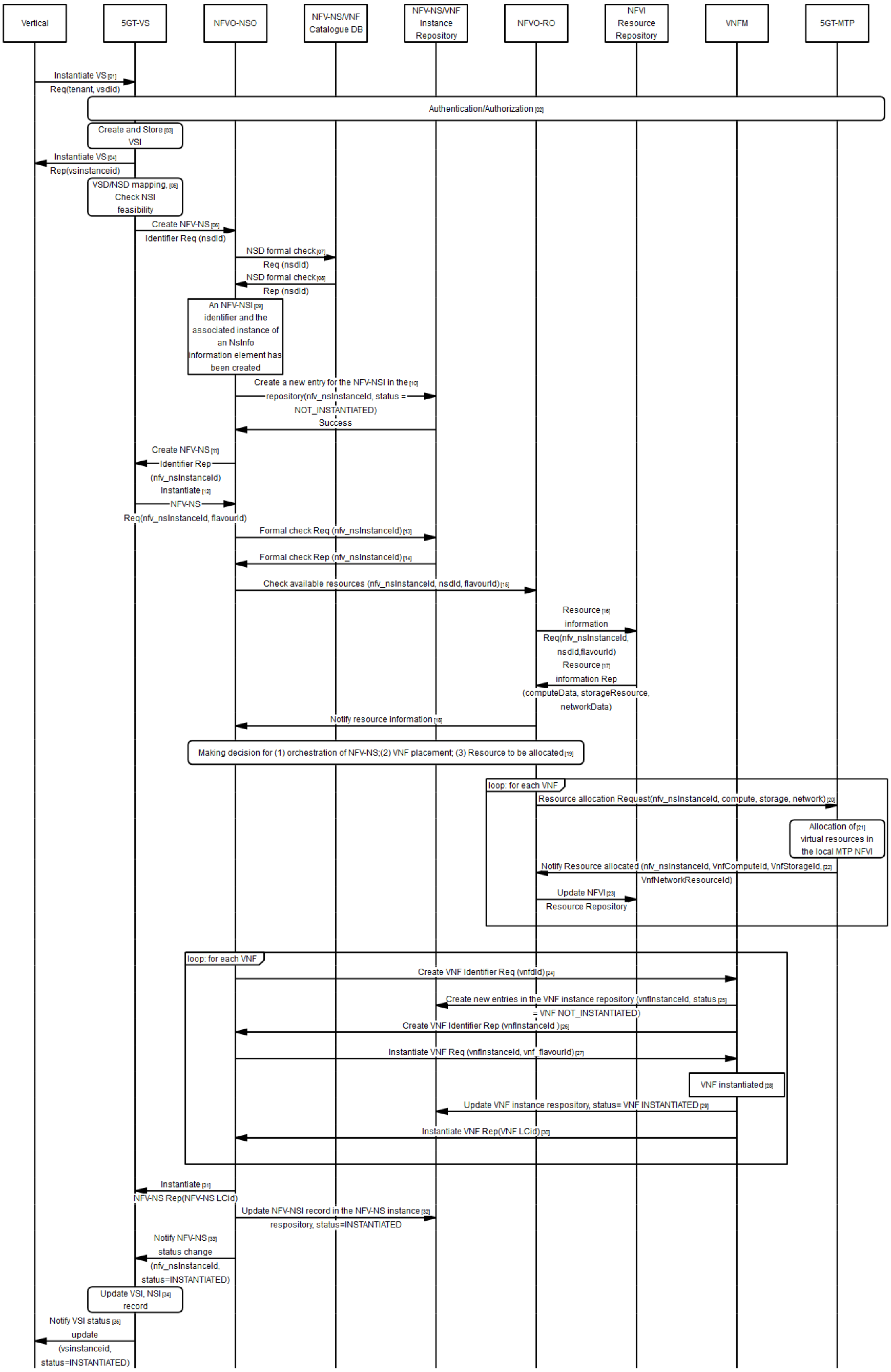

Vertical
5GT-VS
NFVO-NSO
NFV-NS/VNF Catalogue DB
NFV-NS/VNF Instance Repository
NFVO-RO
NFVI Resource Repository
VNFM
5GT-MTP
Instantiate VS [01] Req(tenant, vsdid)
Authentication/Authorization [02]
Create and Store [03] VSI
Instantiate VS [04] Rep(vsinstanceid)
VSD/NSD mapping, [05] Check NSI feasibility
Create NFV-NS [06] Identifier Req (nsdId)
NSD formal check [07] Req (nsdId)
NSD formal check [08] Rep (nsdId)
An NFV-NSI [09] identifier and the associated instance of an NsInfo information element has been created
Create a new entry for the NFV-NSI in the [10] repository(nfv_nsInstanceId, status = NOT_INSTANTIATED)
Success
Create NFV-NS [11] Identifier Rep (nfv_nsInstanceId)
Instantiate [12] NFV-NS Req(nfv_nsInstanceId, flavourId)
Formal check Req (nfv_nsInstanceId) [13]
Formal check Rep (nfv_nsInstanceId) [14]
Check available resources (nfv_nsInstanceId, nsdId, flavourId) [15]
Resource [16] information Req(nfv_nsInstanceId, nsdId,flavourId)
Resource [17] information Rep (computeData, storageResource, networkData)
Notify resource information [18]
Making decision for (1) orchestration of NFV-NS;(2) VNF placement; (3) Resource to be allocated [19]
loop: for each VNF
Resource allocation Request(nfv_nsInstanceId, compute, storage, network) [20]
Allocation of [21] virtual resources in the local MTP NFVI
Notify Resource allocated (nfv_nsInstanceId, VnfComputeId, VnfStorageId, [22] VnfNetworkResourceId)
Update NFVI [23] Resource Repository
loop: for each VNF
Create VNF Identifier Req (vnfdId) [24]
Create new entries in the VNF instance repository (vnfInstanceId, status [25] = VNF NOT_INSTANTIATED)
Create VNF Identifier Rep (vnfInstanceId) [26]
Instantiate VNF Req (vnfInstanceId, vnf_flavourId) [27]
VNF instantiated [28]
Update VNF instance respository, status= VNF INSTANTIATED [29]
Instantiate VNF Rep(VNF LCid) [30]
Instantiate [31] NFV-NS Rep(NFV-NS LCid)
Update NFV-NSI record in the NFV-NS instance [32] respository, status=INSTANTIATED
Notify NFV-NS [33] status change (nfv_nsInstanceId, status=INSTANTIATED)
Update VSI, NSI [34] record
Notify VSI status [35] update (vsinstanceid, status=INSTANTIATED)

FIGURE 12: SERVICE INSTANTIATION WORKFLOW

5.3 Service Modification

Description: The workflow describes the modification of a vertical service, triggered by the vertical. The modification is a simple one, changing e.g., the amount of supported devices.

Prerequisites: The vertical service instance has been instantiated.

Assumptions: The vertical service is a simple one, meaning it can be deployed in one network slice. We also assume that this service is deployed in its own network slice, the service does not share the slice with another slice.

Workflow:

1. NFV-NS update request arrives to the 5GT-SO via the NBI.
2. The SO-NBI component checks authentication and authorization data included in the update request. If the authentication data is valid, the request body (contents) is checked for the semantic validity and consistency, e.g., whether all required entries are present in the request.
3. If the request is valid, it is forwarded to the NFVO-NSO component. To do so, the *nfv_nsInstanceId* is included in this request as an identifier of the service instance to be updated.
4. Assuming asynchronous communication between the involved components, NFVO-NSO checks in the local NFV-NS DB for the requested *nfv_nsInstanceId* to be updated. If found, it sets the instance state to “Updating” status and sends the reply immediately back.
5. NFVO-NSO component resolves all resources currently deployed for the particular networking service instance.
6. NFVO-NSO component calculates the differences between the deployed NFV-NSD instances and the received modified NFV-NSD.
7. As a result of the such a calculation it is possible that:
    a. Some new resources have to be added to the deployed NFV-NS
    b. Some deployed resources have to be removed from the NFV-NS
    c. Some deployed resources have to be modified in the NFV-NS.
8. The set of operations described below are triggered for each (virtualized) resource appearing in the calculated differences.
9. NFVO-NSO sends the corresponding Add/Remove/Update (depending on the required action) request to the NFVO-RO component. For the existing resources *resourceId* is included in such a request.
10. Assuming asynchronous communication between components, the NFVO-RO first looks in the local resource DB for the resource related to the *resourceId.* Then, depending on the targeted action, the resource state is set to “Terminating/Updating”. Next the NFVO-RO sends a reply immediately back. For the new resource to be added the state “Adding” is set.
11. For VNF resources, the RO sends an “Add/Remove/Update” request to the appropriate VNFM. For already deployed resources to be either modified or removed, the *vnfId* should be also included in the request.

12. Assuming asynchronous communication, the VNFM checks internally for the requested *vnfId* to be modified, and sends a reply message immediately back with the *vnfId* state set to “Adding/Terminating/Updating”.
13. VNFM triggers the request to the 5GT-MTP, specifying *vnfId* for “Remove/Update” operations. For “Add” operation only the resource flavor is specified.
14. Assuming asynchronous communication between VNFM and 5G-MTP, it is expected that the 5G-MTP will return a reply message carrying the *vnfId* state “Adding/Terminating/Updating” depending on a nature of the request.
15. For non-VNF related resource, the RO sends an “Add/Remove/Update” request to the 5GT-MTP and includes the *resourceId* if already deployed resources need to be modified or removed. For the new resource to be added, the “Adding” state is set.
16. Assuming asynchronous communication between both the RO and the 5GT-MTP, the latter will return a reply with the *resourceId* state “Adding/Terminating/Updating” depending on a nature of the request.
17. When an operation for a particular resource is completed, the NFVO-RO updates local resource DB and sets it state to “Active” for added or updated resources or “Terminated” for removed resources
18. When all resource operation included in the calculated differences are completed, the NFVO-NSO updates the local NFV-NS DB and sets the service instance status to “Active”.

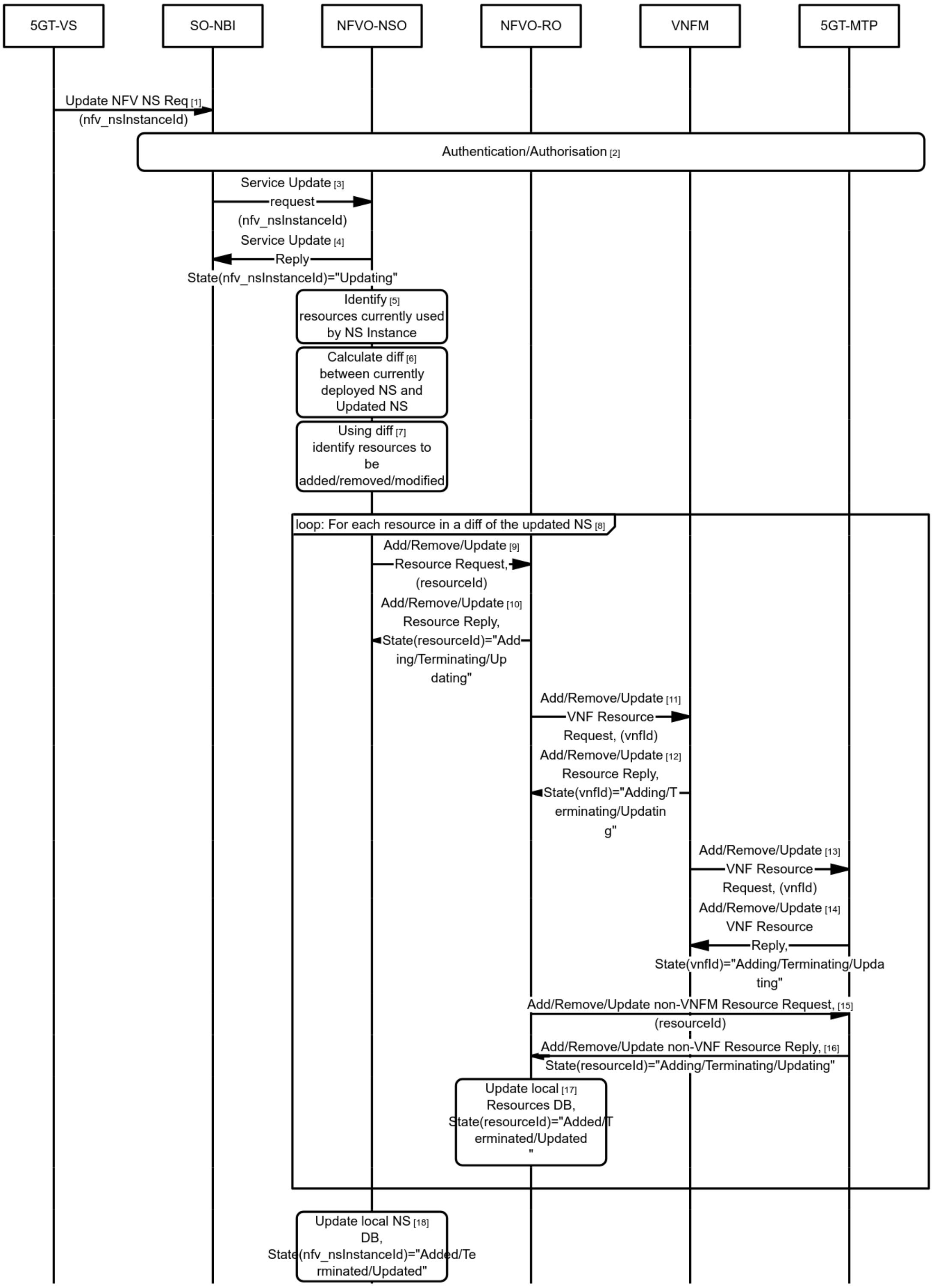

5GT-VS
SO-NBI
NFVO-NSO
NFVO-RO
VNFM
5GT-MTP
Update NFV NS Req [1] (nfv_nsInstanceId)
Authentication/Authorisation [2]
Service Update [3] request (nfv_nsInstanceId)
Service Update [4] Reply
State(nfv_nsInstanceId)="Updating"
Identify [5] resources currently used by NS Instance
Calculate diff [6] between currently deployed NS and Updated NS
Using diff [7] identify resources to be added/removed/modified
loop: For each resource in a diff of the updated NS [8]
Add/Remove/Update [9] Resource Request, (resourceId)
Add/Remove/Update [10] Resource Reply, State(resourceId)="Adding/Terminating/Updating"
Add/Remove/Update [11] VNF Resource Request, (vnfId)
Add/Remove/Update [12] Resource Reply, State(vnfId)="Adding/Terminating/Updating"
Add/Remove/Update [13] VNF Resource Request, (vnfId)
Add/Remove/Update [14] VNF Resource Reply, State(vnfId)="Adding/Terminating/Updating"
Add/Remove/Update non-VNFM Resource Request, [15] (resourceId)
Add/Remove/Update non-VNF Resource Reply, [16] State(resourceId)="Adding/Terminating/Updating"
Update local [17] Resources DB, State(resourceId)="Added/Terminated/Updated"
Update local NS [18] DB, State(nfv_nsInstanceId)="Added/Terminated/Updated"

FIGURE 13: VERTICAL SERVICE MODIFICATION WORKFLOW

5.4 Service Termination

Description: The workflow describes the termination of a vertical service, triggered by the vertical.

Prerequisites: The vertical service instance has been instantiated.

Assumptions: The vertical service is a simple one, meaning it can be deployed in one network slice. We also assume, that this service is deployed in its own network slice, the service does not share the slice with another slice.

Workflow:

1. A NFV-NS termination request arrives via the NBI of the 5GT-SO
2. The SO-NBI component checks authentication and authorization data included into the termination request. In case the authorization data is valid, the request body is checked for the semantic validity, e.g., whether all the required entries are present in the request.
3. If the request is valid, it is forwarded to the NFVO-NSO component. The *nfv_nsInstanceId* is included in this request as an identifier of the service to be terminated.
4. Assuming asynchronous communication between the involved components, the NFVO-NSO checks in the local NFV-NS DB for the *nfv_nsInstanceId* requested for termination: If it is found, the NFVO-NSO sets the related service instance state to “Terminating” and sends the reply immediately back.
5. The NFVO-NSO component resolves all resources deployed for the particular NFV-NS instance.
6. For each resource allocated in the targeted NFV-NS instance to be terminated, the NFVO-NSO launches termination workflow.
7. The NFVO-NSO sends the termination request to the NFVO-RO component where the *resourceId* is included in this request specifying the particular resource to be terminated.
8. Assuming asynchronous communication between the components, the NFVO-RO checks in the local resource DB for the *resourceId* requested, sets resource state to “Terminating” and sends the reply immediately back.
9. For VNF resources, the NFVO-RO sends the termination request to the appropriate VNFM including the *vnfId*.
10. Assuming asynchronous communication, the VNFM checks internally for the requested *vnfId* to be terminated. Then it sends a reply immediately back with the *vnfId* state set to “Terminating”.
11. The VNFM triggers the termination request to the SO-MTP specifying the *vnfId* for termination in this request.
12. Assuming asynchronous communication between the VNFM and the SO-MTP, it is expected that SO-MTP will return a reply with the *vnfId* state “Terminating”.
13. For non-VNF resources, the NFVO-RO sends the termination request to the SO-MTP including the *resourceId*.
14. Assuming asynchronous communication between the NFVO-RO and the SO-MTP, the SO-MTP will return a reply with the *resourceId* specifying the state “Terminating”.
15. When a particular resource is terminated, the NFVO-RO updates the local resource DB and sets it state to “Terminated”
16. When all resources included in the *nfv_nsInstanceId* are terminated, the NFVO-NSO updates the local NFV-NS DB and sets the service instance status to “Terminated”.

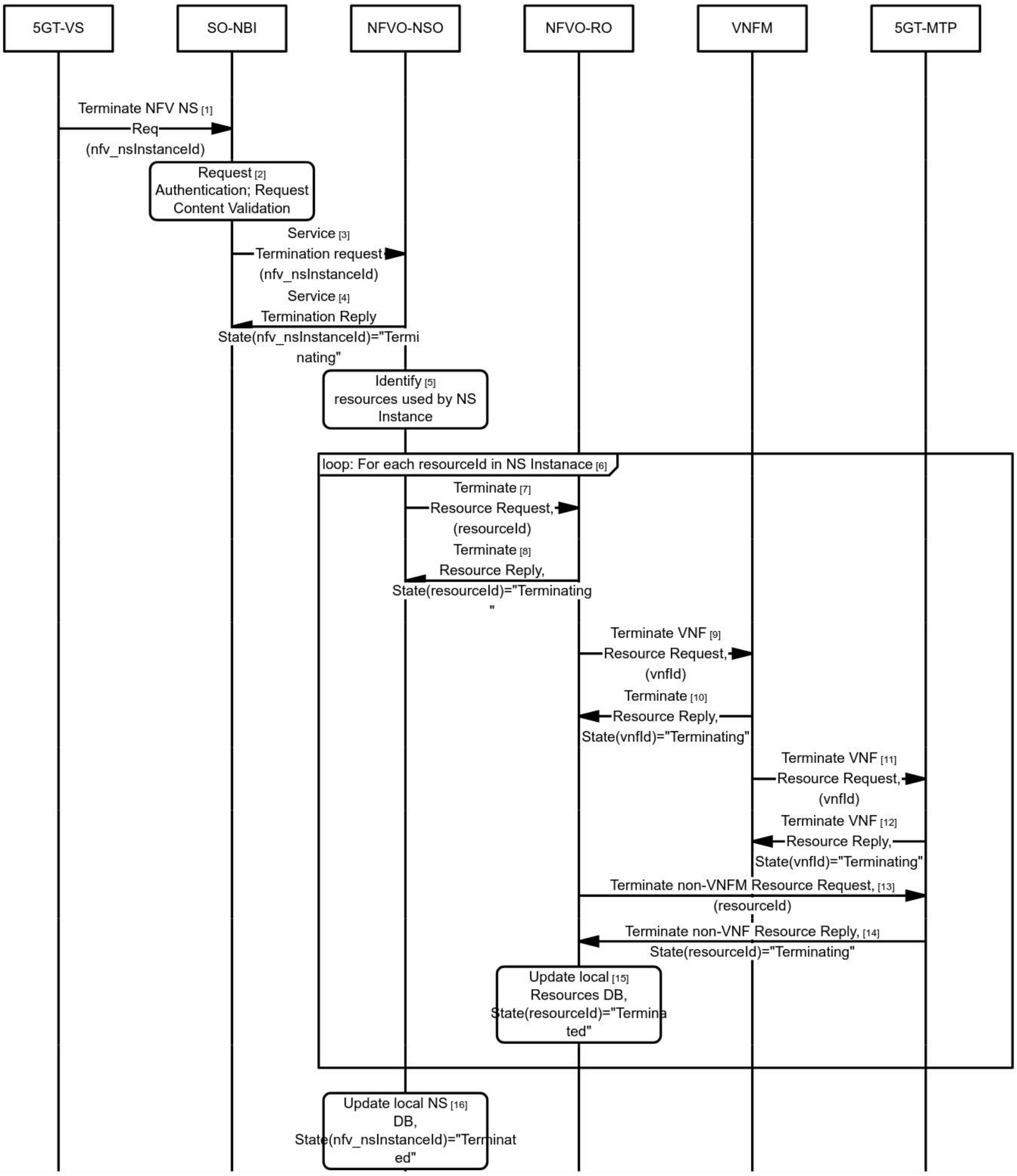

5GT-VS
SO-NBI
NFVO-NSO
NFVO-RO
VNFM
5GT-MTP
Terminate NFV NS [1] Req (nfv_nsInstanceId)
Request [2] Authentication; Request Content Validation
Service [3] Termination request (nfv_nsInstanceId)
Service [4] Termination Reply State(nfv_nsInstanceId)="Terminating"
Identify [5] resources used by NS Instance
loop: For each resourceId in NS Instanace [6]
Terminate [7] Resource Request, (resourceId)
Terminate [8] Resource Reply, State(resourceId)="Terminating"
Terminate VNF [9] Resource Request, (vnfId)
Terminate [10] Resource Reply, State(vnfId)="Terminating"
Terminate VNF [11] Resource Request, (vnfId)
Terminate VNF [12] Resource Reply, State(vnfId)="Terminating"
Terminate non-VNFM Resource Request, [13] (resourceId)
Terminate non-VNF Resource Reply, [14] State(resourceId)="Terminating"
Update local [15] Resources DB, State(resourceId)="Terminated"
Update local NS [16] DB, State(nfv_nsInstanceId)="Terminated"

FIGURE 14: SERVICE TERMINATION WORKFLOW

5.5 Service assurance (including service monitoring) for the local 5GT-SO domain

Description:

The workflow describes the assurance phase as part of the lifecycle management of an NFV-NS instance. The service assurance relies on the collection and elaboration of monitoring data through a distributed monitoring platform.

Prerequisites: The NFV-NS instance has been instantiated.

Assumptions: The NFV-NS instance is correctly deployed and the monitoring data collection correctly starts at the time the NFV-NS instance deployment is accomplished.

Workflow:

1. The 5GT-NFVO completes the deployment of a Network Service (NFV-NS).
2. The 5GT-NFVO sends a request for the activation of a data monitoring collection to the *Service Monitoring Data Consumer* in order to obtain performance metrics related to the instantiated NFV-NS or its component VNFs. The monitoring parameters that need to be collected are those specified in the NSD *monitoredInfo* field of the instantiated service, referred to as 5GT-SO *performance metrics* in the following.
3. The *Service Monitoring Data Consumer* sends back a response about the activation request. If the request is accepted and the activation process can start correctly then a response with “status=IN PROGRESS” is sent back and the following actions are performed for each 5GT-SO *performance metric* to be collected.
4. The 5GT-SO *Monitoring Data Consumer* starts the procedure to create a Performance Management (PM) job and to subscribe with the local 5GT-SO *Monitoring Service* to receive the metrics it is interested in. Thus, 5GT-SO *Monitoring Data Consumer* sends a request to create a PM job to the local 5GT-SO *Monitoring Service*.
5. The 5GT-SO *Monitoring Service* translates (i.e., maps) each 5GT-SO *performance metric* into one or more *5GT-MTP performance metric(s)*; this is attained relying on the information about the virtual resources composing the NFV-NS instance stored in the 5GT-SO NFV-NS*/VNF Instance Repository.* For each 5GT-MTP *performance metric,* a PM job creation workflow starts into the 5GT-MTP *Monitoring Service* (step 6 to step 11).
6. The 5GT-SO *Monitoring Service* sends a request to create a PM job to the local 5GT-MTP *Monitoring Service*.
7. The 5GT-MTP *Monitoring Service* sends back a response with the identifier of the job just created (i.e., *pmJobId*)
8. The 5GT-SO *Monitoring Service* creates a new entry in the *PM job repository*
9. The *5GT-SO Monitoring Service* subscribes with the local 5GT-MTP *Monitoring Service* to receive the metrics it is interested in.
10. The 5GT-MTP *Monitoring Service* sends back a response with the identifier of the subscription (i.e., *subscriptionId*)
11. The 5GT-SO *Monitoring Service* creates a new entry in the *monitoring subscription repository.* At this point, the 5GT-SO *Monitoring Service* is able to receive all the required monitoring data from the 5GT-MTP *Monitoring Service*.
12. The 5GT-SO *Monitoring Service* instantiates a new internal job (e.g., a new thread) to process and elaborate such data to eventually generate the targeted 5GT-SO *performance metrics*. This step concludes the creation of a PM job in the 5GT-SO *Monitoring Service* started at step 4.
13. The 5GT-SO *Monitoring Service* sends back to the 5GT-SO *Monitoring Data Consumer* a response with the identifier of the job just created (i.e., *pmJobId*)
14. To receive notifications about the 5GT-SO *performance metrics*, the 5GT-SO *Monitoring Data Consumer* subscribes with the 5GT-SO *Monitoring Service* to receive the metrics it is interested in.
15. The 5GT-SO *Monitoring Service* sends back a response with the identifier of the subscription (i.e., *subscriptionId*)

16. The 5GT-SO *Monitoring Data Consumer* creates a new entry in the *monitoring subscription repository.* At this point, the 5GT-SO *Monitoring Data Consumer* is able to receive all the required monitoring data from the 5GT-SO *Monitoring Service*.
17. The 5GT-SO *Monitoring Data Consumer* reports the NFVO about the conclusion of the monitoring data activation phase (i.e., "status=COMPLETED").
18. Following the subscription/notification and report queries approaches, at service runtime, as the 5GT-MTP *Monitoring Service* produces new monitoring data for which the 5GT-SO *Monitoring Service* has an active subscription, it sends a notification.
19. The 5GT-SO *Monitoring Service* sends a request to retrieve the available monitoring data.
20. The 5GT-MTP *Monitoring Service* sends back the requested monitoring data.
21. If needed, the 5GT-SO *Monitoring Service* aggregates the monitoring data to obtain the needed 5GT-SO performance metric value (e.g., it aggregates the vCPUs consumed by all the VMs composing the NFV-NS instance).
22. The 5GT-SO *Monitoring Service* stores the result as a new 5GT-SO performance metric value into its repository.
23. The 5GT-SO *Monitoring Data Consumer* is in turn notified about the availability of a new performance report related to the given NFV-NS instance.
24. The 5GT-SO *Monitoring Data Consumer* sends a request to the 5GT-SO *Monitoring Service* to retrieve the new performance report.
25. The 5GT-SO *Monitoring Service* responds to the 5GT-SO *Monitoring Data Consumer* with the required performance report.
26. In case of SLA assurance, the performance data are sent to the *SLA Manager*.
27. The *SLA Manager* start a process to validate SLA and to check whether the SLAs are met or not.
28. If the request's SLA parameters are not met, the *SLA Manager* triggers a lifecycle action at the NFVO (e.g., scaling-up) to recover from the SLA violation.

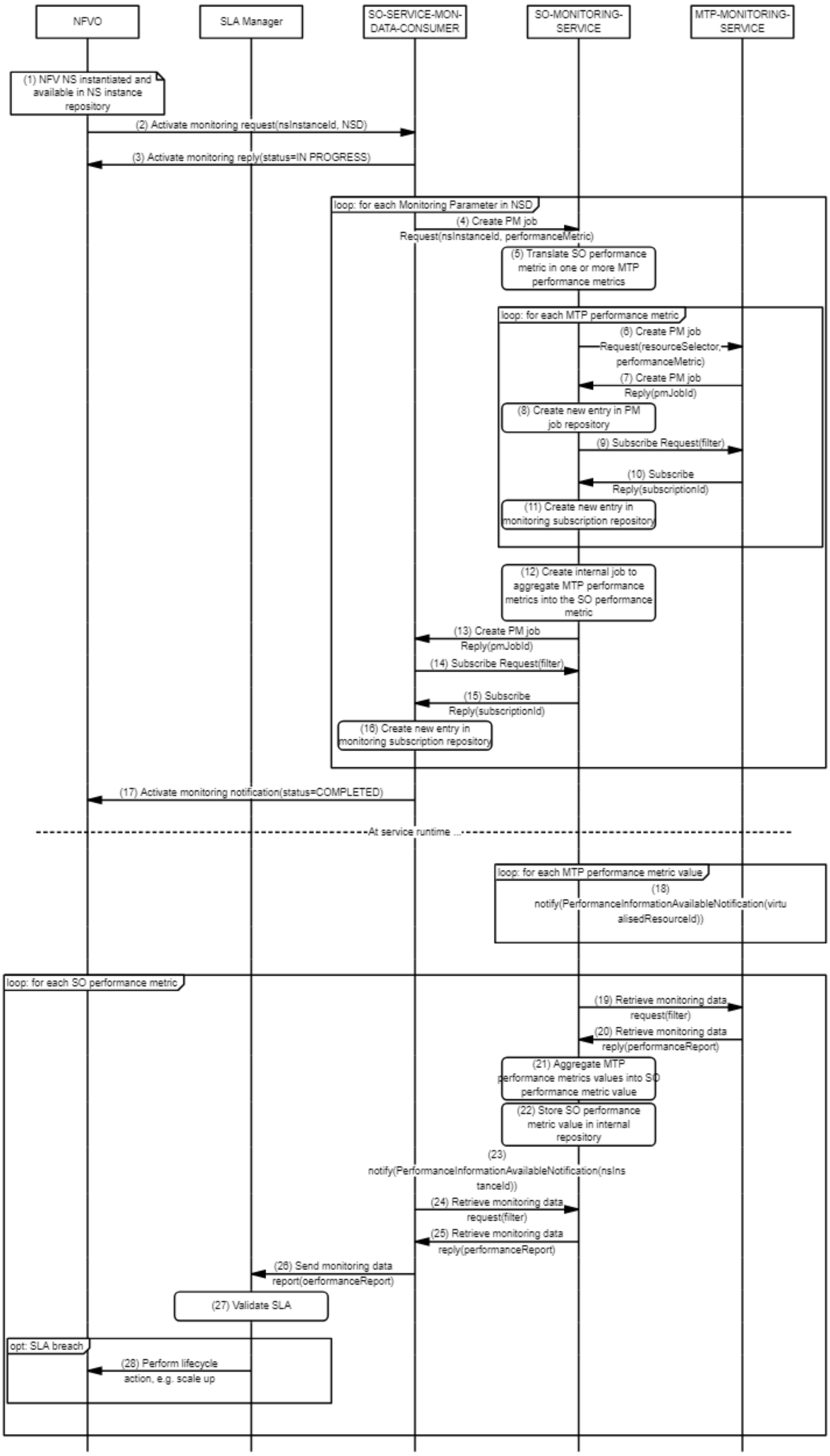

NFVO
SLA Manager
SO-SERVICE-MON-DATA-CONSUMER
SO-MONITORING-SERVICE
MTP-MONITORING-SERVICE
(1) NFV NS instantiated and available in NS instance repository
(2) Activate monitoring request(nsInstanceId, NSD)
(3) Activate monitoring reply(status=IN PROGRESS)
loop: for each Monitoring Parameter in NSD
(4) Create PM job Request(nsInstanceId, performanceMetric)
(5) Translate SO performance metric in one or more MTP performance metrics
loop: for each MTP performance metric
(6) Create PM job Request(resourceSelector, performanceMetric)
(7) Create PM job Reply(pmJobId)
(8) Create new entry in PM job repository
(9) Subscribe Request(filter)
(10) Subscribe Reply(subscriptionId)
(11) Create new entry in monitoring subscription repository
(12) Create internal job to aggregate MTP performance metrics into the SO performance metric
(13) Create PM job Reply(pmJobId)
(14) Subscribe Request(filter)
(15) Subscribe Reply(subscriptionId)
(16) Create new entry in monitoring subscription repository
(17) Activate monitoring notification(status=COMPLETED)
At service runtime ...
loop: for each MTP performance metric value
(18) notify(PerformanceInformationAvailableNotification(virtualisedResourceId))
loop: for each SO performance metric
(19) Retrieve monitoring data request(filter)
(20) Retrieve monitoring data reply(performanceReport)
(21) Aggregate MTP performance metrics values into SO performance metric value
(22) Store SO performance metric value in internal repository
(23) notify(PerformanceInformationAvailableNotification(nsInstanceId))
(24) Retrieve monitoring data request(filter)
(25) Retrieve monitoring data reply(performanceReport)
(26) Send monitoring data report(oerformanceReport)
(27) Validate SLA
opt: SLA breach
(28) Perform lifecycle action, e.g. scale up

FIGURE 15: SERVICE ASSURANCE WORKFLOW

5.6 Service federation

Description: The workflow describes federation of NFV-NSs (NSaaS) between different administrative domains

Prerequisites: Different administrative domains need to have some business relations.

Assumptions: 5GT-SO receives a request for instantiation of a certain NFV-NS (with the NSD) that the 5GT-SO has no sufficient resource capabilities or capacity, it can exploit the service federation for accommodating the requested service.

Workflow:

1. The agreed administrative domains (white and grey in Figure 16) create a set of NFV-NSs that each domain can offer to each peering administrative domain. A set of agreed and offered NFV-NSs create a *white-list* of services which is referred to as catalogue of services. The pre-phase is concluded when the catalogues from each peering administrative domain is stored into the Catalogue DB.
2. NFV-NS instantiation request received from the 5GT-VS.
3. The 5GT-SO NFVO-NSO performs decomposition of the requested NFV-NS. The decomposition of requested NFV-NS can result in a single or set of multiple nested NFV-NSs. In this workflow, a decision is made to consume one of the set of decomposed nested NFV-NSs through federation of NFV-NS or by consuming NFV-NSaaS. The decomposition of NFV-NS is for further study.
4. NFVO-NSO checks the NFV-NS Catalogue for the desired decomposed NFV-NS if it can be consumed from another administrative domain. The selection of the potential peering 5GT-SOs is based on either the parameters of the NSD or best-matching capabilities to enable the requested NFV-NS instance. The 5GT-SO takes the role of consumer 5GT-SO (white on Figure 16).
5. Consumer 5GT-SO NFVO-NSO sends the "check for availability" requests to multiple peering 5GT-SOs (or their NFVO-NSOs) that potentially could enable the NFV-NS.
6. The peering 5GT-SO NFVO-NSOs check their availability of the requested service and make decision if it can provide a service instance to the consumer 5GT-SO NFVO-NSO.
7. The peering 5GT-SO NFVO-NSOs send feedback to the consumer 5GT-SO NFVO-NSO.
8. Depending on the received results, the consumer 5GT-SO NFVO-NSO decides for the provider 5GT-SO NFVO-NSO or the procedure is terminated if all received responses are negative. The consumer 5GT-SO NFVO-NSO decides for the best-matching provider 5GT-SO NFVO-NSO(i.e. provides lowest latency for a low-latency NFV-NS and/or at minimal cost).
9. The consumer 5GT-SO NFVO-NSO sends a request for NFV-NS instantiation to the selected provider 5GT-SO NFVO-NSO.
10. The selected provider 5GT-SO NFVO-NSO initiates the service creation and consequently the instantiation procedure involving its constituent 5GT-MTP.
11. Once the requested federated NFV-NS is instantiated, a positive feedback is sent back to the 5GT-SO NFVO-NSO which becomes the consumer of the federated NFV-NS instance.

12. The selected provider 5GT-SO NFVO-NSO updates its constituent Catologue DB (grey)
13. The consumer 5GT-SO NFVO-NSO consumes the federated NFV-NS by adding it to the set of nested services.

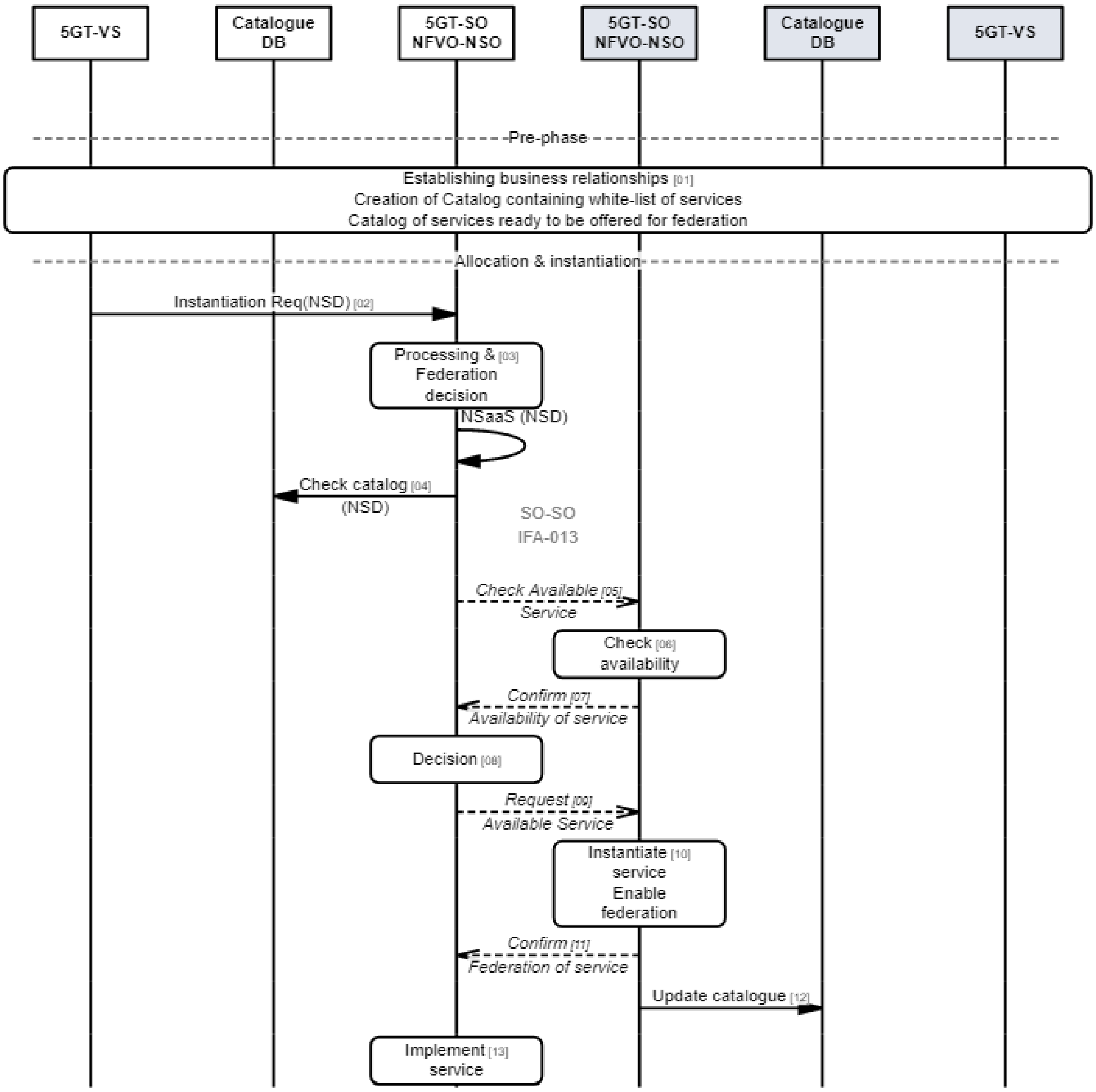

5GT-VS
Catalogue DB
5GT-SO NFVO-NSO
5GT-SO NFVO-NSO
Catalogue DB
5GT-VS
Pre-phase
Establishing business relationships [01]
Creation of Catalog containing white-list of services
Catalog of services ready to be offered for federation
Allocation & instantiation
Instantiation Req(NSD) [02]
Processing & [03] Federation decision
NSaaS (NSD)
Check catalog [04] (NSD)
SO-SO IFA-013
Check Available [05] Service
Check [06] availability
Confirm [07] Availability of service
Decision [08]
Request [09] Available Service
Instantiate [10] service Enable federation
Confirm [11] Federation of service
Update catalogue [12]
Implement [13] service

FIGURE 16: SERVICE FEDERATION WORKFLOW

5.7 Resource federation

Description: The workflow describes federation of NFVI resources or resource abstractions (NFVIaaS) between different administrative domains

Prerequisites: Different administrative domains need to have already established business relations.

Assumptions: There are lack of NFVI resources at the local (constituent) 5GT-MTP.

Workflow:

1. The **pre-phase** usually is part of the same business agreements between administrative domains, held “offline” as in the NFV-NSaaS. All the terms and conditions are agreed about how the interactions between the agreed administrative domains will proceed.
2. The 5GT-MTP SLPOC sends/advertises resource abstractions to the NFVI Resource Repository (Advertising resource abstractions).
3. The NFVI Resource Repository stores the received resource abstractions.
4. The SO NFVO-RO reads the NFVI Resource Repository for all available resources, which are previously filled by the 5GT-MTP SLPOC.
5. The 5GT-SO NFVO-RO calculates the available resource abstractions (ready to be offered) for federation to potential consumer 5GT-SOs.
6. The calculated resource abstractions are assigned to the 5GT-SO Resource Advertisement.
7. The 5GT-SO Resource Advertisement block is responsible to broadcast/advertise the available resource abstractions. A process of exchanging (sending and receiving) calculated resource abstractions among different domains.
8. When an advertisement from other peering 5GT-SO is received, the carried information is passed to the 5GT-SO NFVO-RO.
9. Such information is then filtered and stored into the NFVI Resource Repository.
10. In the 5GT-SO NFVO-RO, federation of resources is triggered for instantiating a NFV-NS by using resources from external 5GT-SO.
11. The 5GT-SO NFVO-RO reads the NFVI Resource Repository.
12. Available resource abstractions for federation from peering 5GT-SOs are provided to the 5GT-SO NFVO-RO.
13. The 5GT-SO NFVO-RO issues a request for the actual allocation of the available resource abstractions at the peering 5GT-SO NFVO-RO.
14. The peering (provider) 5GT-SO NFVO-RO processes the request.
15. The provider 5GT-SO NFVO-RO sends it to its constituent 5GT-MTP.
16. The provider 5GT-MTP allocates and instantiates the resources that correspond to the requested resource abstractions.
17. Once the resources are allocated and instantiated, the provider 5GT-MTP sends back a positive response to the provider 5GT-SO NFVO-RO.
18. At the same time, the provider 5GT-MTP updates the provider NFVI Resource Repository with the newly abstracted view of the resources as result of the performed actions.
19. The provider 5GT-SO NFVO-RO provides a positive response to the consumer 5GT-SO NFVO-RO with some details of the newly allocated resources.
20. The consumer 5GT-SO NFVO-RO includes the allocated federated resources in a NFV-NS. The connection from the consumer 5GT-SO NFVO-RO to the federated resources will pass through the provider 5GT-SO NFVO-RO (as a forwarding point) up to the peering provider 5G-MTP as an end-point.
21. The provider 5GT-SO NFVO-RO re-calculates its abstraction capabilities and triggers the advertisement phase by reading the NFVI Resource Repository.
22. New resource abstractions are calculated.
23. The provider 5GT-SO NFVO-RO sends the new resource abstractions to the 5GT-SO Resource Advertisement.

24. The 5GT-SO Resource Advertisement broadcasts the new resource abstractions to all peering 5GT-SOs.
25. The consumer 5GT-SO Resource Advertisement receives the new resource abstractions and passes to the 5GT-SO NFVO-RO.
26. The 5GT-SO NFVO-RO stores the newly received resource abstractions in its NFVI Resource Repository.

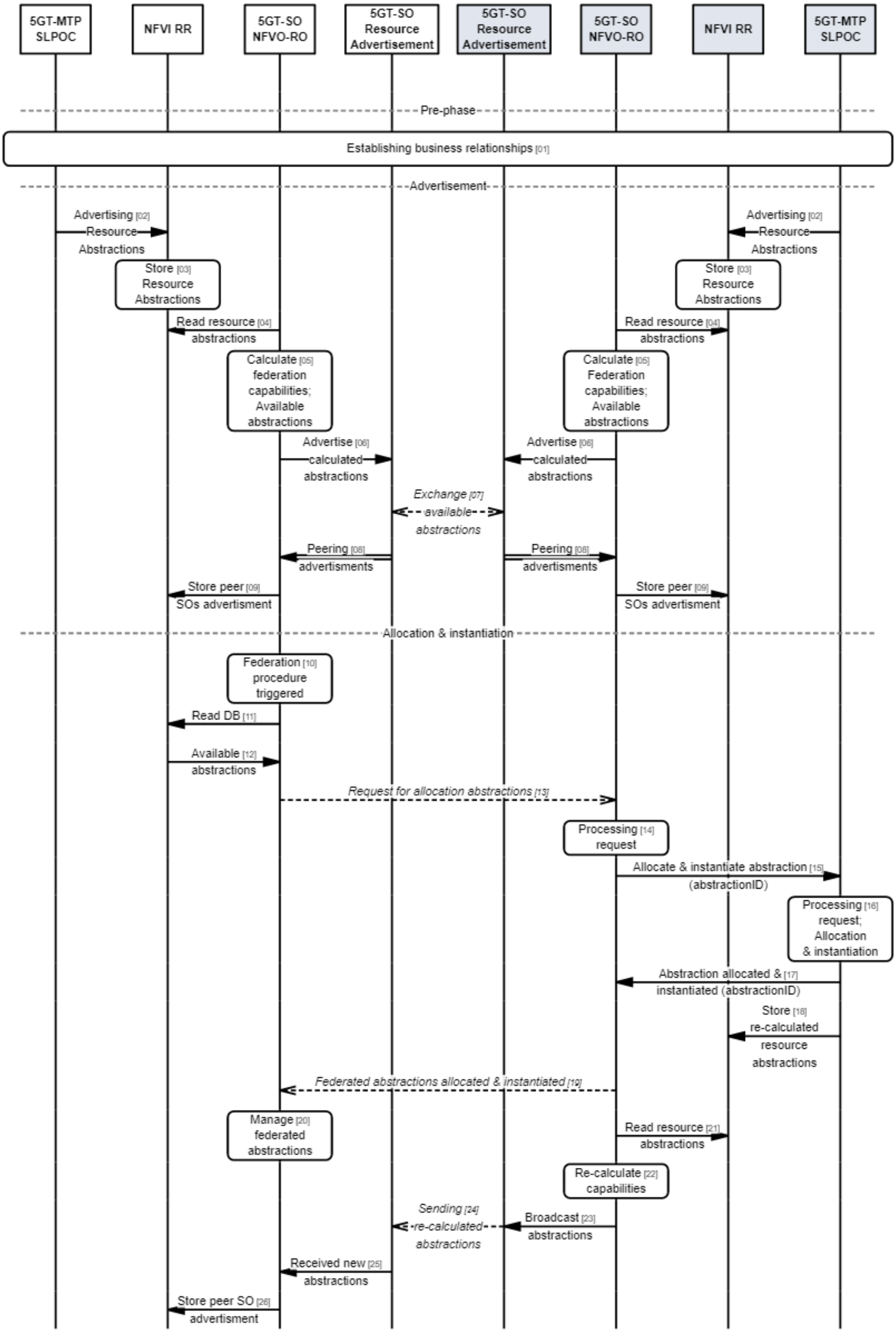

5GT-MTP SLPOC
NFVI RR
5GT-SO NFVO-RO
5GT-SO Resource Advertisement
5GT-SO Resource Advertisement
5GT-SO NFVO-RO
NFVI RR
5GT-MTP SLPOC
Pre-phase
Establishing business relationships [01]
Advertisement
Advertising [02] Resource Abstractions
Advertising [02] Resource Abstractions
Store [03] Resource Abstractions
Store [03] Resource Abstractions
Read resource [04] abstractions
Read resource [04] abstractions
Calculate [05] federation capabilities; Available abstractions
Calculate [05] Federation capabilities; Available abstractions
Advertise [06] calculated abstractions
Advertise [06] calculated abstractions
Exchange [07] available abstractions
Peering [08] advertisments
Peering [08] advertisments
Store peer [09] SOs advertisment
Store peer [09] SOs advertisment
Allocation & instantiation
Federation [10] procedure triggered
Read DB [11]
Available [12] abstractions
Request for allocation abstractions [13]
Processing [14] request
Allocate & instantiate abstraction [15] (abstractionID)
Processing [16] request; Allocation & instantiation
Abstraction allocated & [17] instantiated (abstractionID)
Store [18] re-calculated resource abstractions
Federated abstractions allocated & instantiated [19]
Manage [20] federated abstractions
Read resource [21] abstractions
Re-calculate [22] capabilities
Broadcast [23] abstractions
Sending [24] re-calculated abstractions
Received new [25] abstractions
Store peer SO [26] advertisment

FIGURE 17: RESOURCE FEDERATION WORKLOW

5.8 NFV-NS instantiation including MEC applications

5G-TRANSFORMER considers three scenarios to deploy a MEC application: (i) a MEC application requiring traffic redirection and consuming MEC services; (ii) a MEC application requiring only MEC services; (iii) MEC application requiring only traffic redirection. For the first scenario, the MEC application requires that also an EPC (i.e., MME, HSS, SPGW-C), SPGW-U, eNodeB(s) and Mobile Edge Platform (MEP) are deployed. Specifically, the SPGW-U and the MEP should be run at the edge with the MEC application. For the second scenario, the MEC application needs eNodeB(s) and MEP. For the third scenario, the MEC application can be deployed without the need of other components.

Description: The workflow describes the instantiation of a NFV-NS instance including AppD (i.e. MEC applications)

Prerequisites: None.

Assumptions: The NSD includes AppD.

Workflow:

The workflow of *NFV_nsinstantion* when an AppD is included in the NSD, highly depends on the adopted MEC scenario. One common feature is the placement decision since all the applications described by AppD shall typically be run at the edge. The following workflow shows the instantiation of a NFV-NS, which reflects the case of having AppD in the NSD.

1. The procedures (from 01 to 22, and 41) are common for the three scenarios of MEC deployment, and similar to the workflow described in section 5.2.
2. The 5GT-MTP will place, as requested by the SO, the applications defined via AppD in the Edge Cloud to ensure latency requirements.
3. The difference between scenario 1 and the other two are the procedures labelled by 31, 32, 33 and 34. These procedures are needed to enforce the traffic rule as requested by the AppD of the MEC applications.
4. Once the 5GT-SO receives the confirmation from the 5GT-MTP about the Edge resource allocation, it sends to the VNFM (31) a request to modify the configuration of SGW-U[10] instance to request the traffic redirection to the MEC application. The messages between the 5GT-SO and VNFM are using some of the interfaces described in ETSI GS NFV-IFA 007 [19] (section 7.6.2).
5. The VNFM sends a message (32, 33) to the VNF running the SGW-U to update its configuration with the traffic redirection requested by the MEC application (information encoded in the AppD). The messages between the VNFM and EM/SGW-U are using some of the interfaces described in ETSI GS NFV-IFA 008 [20] (section 6), while the *vnfConfigurationData* is 5G-TRANSFORMER specific, and thus require a devised 5G-TRANSFORMER specific Element Manager (EM) to reflect the traffic redirection at the SGW-U. Indeed, as described in the MECinNFV [28] the traffic redirection mechanism is an open issue, and no standard solution has been proposed yet. However, for the scenarios 2 and 3, these messages are not needed since the 5GT-MTP will

[10] The equivalent entity of SGW-U in 5G is User Plane Function (UPF)

implicitly enforce the traffic redirection rules (scenario 3), and the mandatory communications links (e.g., eNodeB to MEP) using the Virtual Links (VLs), NFP of the VNFFG. A more detailed description can be found in [11].

6. Once the VNFM has received the acknowledgement of the SGW-U that the new rules have been applied, it informs the 5GT-SO.
7. The 5GT-SO informs the 5GT-VS about the NFV-NS instantiation status

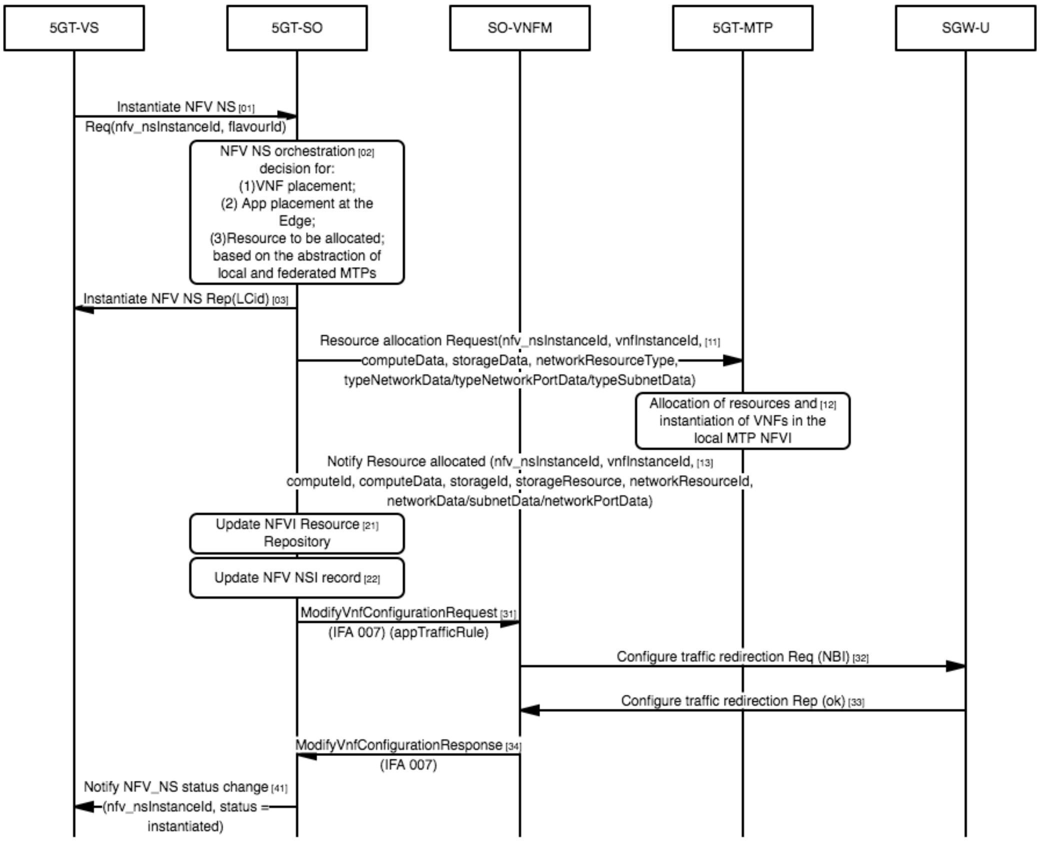

5GT-VS
5GT-SO
SO-VNFM
5GT-MTP
SGW-U
Instantiate NFV NS [01]
Req(nfv_nsInstanceId, flavourId)
NFV NS orchestration [02] decision for: (1)VNF placement; (2) App placement at the Edge; (3)Resource to be allocated; based on the abstraction of local and federated MTPs
Instantiate NFV NS Rep(LCId) [03]
Resource allocation Request(nfv_nsInstanceId, vnfInstanceId, [11] computeData, storageData, networkResourceType, typeNetworkData/typeNetworkPortData/typeSubnetData)
Allocation of resources and [12] instantiation of VNFs in the local MTP NFVI
Notify Resource allocated (nfv_nsInstanceId, vnfInstanceId, [13] computeId, computeData, storageId, storageResource, networkResourceId, networkData/subnetData/networkPortData)
Update NFVI Resource [21] Repository
Update NFV NSI record [22]
ModifyVnfConfigurationRequest [31] (IFA 007) (appTrafficRule)
Configure traffic redirection Req (NBI) [32]
Configure traffic redirection Rep (ok) [33]
ModifyVnfConfigurationResponse [34] (IFA 007)
Notify NFV_NS status change [41] (nfv_nsInstanceId, status = instantiated)

FIGURE 18: CASE OF MEC APPLICATIONS INSTANTIATION WORKFLOW

6 Service Orchestration and Federation Algorithms

As summarized in Section 5.2, orchestration decisions are triggered whenever the 5GT-VS requests the instantiation of a network service or modifications of an existing network service. In both cases, the 5GT-SO NFVO has two main decisions to make, namely:

1. How to decompose the network service graph (i.e., NSDs) into several network service segments and, consequently, where to implement each segment, whether in the local 5GT-MTP domain or by leveraging neighbor SOs (federation);
2. How to map a service segment into a set of virtual resources, i.e., where to run the component VNFs in the virtual infrastructure based on specified resource demand.

The first decision takes place when the federation case comes into play and it is addressed in Section 6.2. The second decision is further developed in Section 6.1 next.

6.1 Service Orchestration Algorithms

In the following, we examine the decisions that service orchestration algorithms have to make, the input values they can rely upon, and the computational strategies they can be based upon.

6.1.1 Objectives and decisions to make

Service orchestration decisions actually consist of three steps, namely:

- VNF[11] placement, i.e., where in the virtual infrastructure exposed by the 5GT-MTP each VNF shall be hosted;
- Resource assignment, i.e., how much virtual resources (computational power, memory, storage…) each VNF shall be assigned;
- Traffic routing, i.e., which virtual links (VL)[12] should be used to transfer data between different VNFs, and bandwidth assignment.

The three decisions above impact one another. As an example, placing multiple VNFs at the same host reduces the amount of computational power each can be assigned; placing them at different hosts will lead to a higher bandwidth consumption. It follows that making decisions separately, e.g., first deciding the placement of all VNFs and then assigning them, e.g., computational resources, can yield decisions that are not only suboptimal but also infeasible.

Within the context of the 5GT-SO architecture described in Figure 3, the entities in charge of these decisions are the NFVO-NSO and the NFVO-RO. However, owing to the interdependence between decisions, a convenient approach is therefore to make *joint* placement, assignment and routing decisions, thus accounting for all the aspects of the orchestration problem.

[11] For simplicity, we refer to VNFs only in the description of our algorithms. Notice however that they can be easily extended to VNFCs.
[12] In this context, VLs are the links exposed by the 5GT-MTP to the 5GT-SO. As such, they are distinct from IFA013 virtual links.

6.1.2 Input Information

The 5GT-SO interacts with two main entities, namely, the 5GT-VS and the 5G-MTP. Both provide information that is used to make orchestration decisions, as summarized in Table 6.

TABLE 6: MAIN INPUT DATA TO THE ORCHESTRATION PROBLEM

Piece of information	Entity providing the information	Moment at which it is provided	Remarks
VNFs each NFV-NS requires and connectivity among them	5GT-VS	Service on-boarding	Provided as NSDs. Stored at the SO's NSD repository.
Resources needed by each VNF instance	5GT-VS	Service on-boardinstantiation	Provided as a deployment flavors. Stored at the SO's NSD repository.
Available resources	5G-MTP	Periodic or on-demand updates from the 5GT-MTP.	Include hosts (and their capabilities) and VLs (and their capacity/delay). Stored within the SO's resource repository.

Intuitively, the 5GT-VS asks the 5GT-SO to instantiate a certain network service, and it is the 5GT-SO's task to decide how to provide it, i.e., which of the virtual resources exposed by the 5GT-MTP shall be used.

6.1.3 System Model

As a preliminary step, the 5GT-SO has to translate the input data described in Table 6 into an *instance-level system model*, synthetically representing all the elements to account for. These elements can be categorized in:

- Related to the VNFFG - VNFs to place, capabilities they require, amount of data they exchange;
- Related to the virtual infrastructure - hosts able to run VNFs and their capabilities, virtual links and their capacity/delay.

Both the VNF-related and the infrastructure-related elements of the system model can be represented as graphs, namely *the VNFFG* and a virtual *infrastructure* graph.

The 5GT-VS communicates to 5GT-SO, through deployment flavors, the minimum and maximum amount of each type of resource (e.g., CPU, memory, storage...) each VNF should be assigned. This information translates into constraints on the placement of VNFs across hosts.

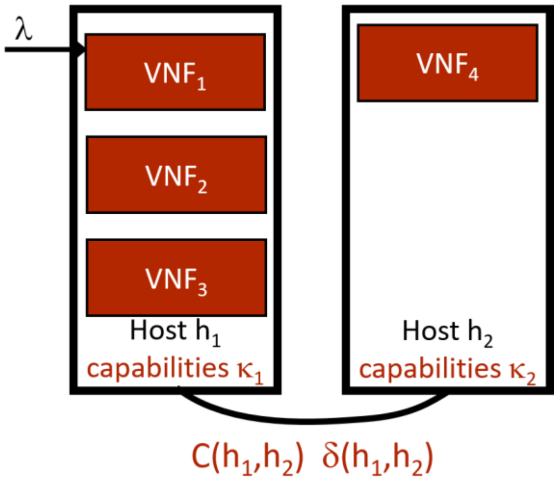


FIGURE 19: EXAMPLE OF VIRTUAL INFRASTRUCTURE GRAPH

Figure 19 shows an example of the virtual infrastructure graph. Its vertices are the hosts in charge of running the VNFs; edges correspond to virtual links connecting the hosts. Nodes are labeled with the capabilities $\kappa(h)$ of the host h_x they are associated with, e.g., its CPU; edges between hosts h_1 and h_2 are labeled with the capacity $C(h_1,h_2)$ and delay $\delta(h_1,h_2)$ of the virtual link they represent.

6.1.4 Decision Variables

The first decision the 5GT-SO is in charge of is VNF placement, i.e., assigning to each VNF one host to run on. Notice that the mapping is many-to-one, i.e., each VNF is assigned to one host while one host will, in general, be assigned multiple VNFs. This is also exemplified in Figure 19, where host h_1 runs three VNFs. VNF placement decisions represented through binary variables A(h,v), each expressing whether a given VNF v is assigned to a certain host h. Using a binary variable reflects the fact that VNFs cannot be split across multiple hosts.

The second decision the 5GT-SO has to make is resource assignment, i.e., how the resources (CPU, memory, disk…) available at each host are shared among the VNFs it runs. For many resources, e.g., memory, this decision is trivial: VNFs shall get exactly the amount they need, since giving less would result in malfunctioning, and giving more would yield no benefit.

Other resources, most notably, CPU, can be allocated in a *flexible* fashion. VNFs have a minimum requirement that must be satisfied but assigning a VNF more CPU than the minimum may result in faster processing. Capturing flexible resource allocation is a major goal of our model, as it allows the 5GT-SO to discriminate among the different decisions that satisfy the target KPIs, most notably, the maximum service time. As an example, an allocation strategy assigning to each VNF the minimum amount of CPU it requires could allow service time requirements to be satisfied while using fewer hosts than another, resulting in a lower power consumption and therefore lower costs.

The third decision, traffic routing, only concerns the 5GT-SO to a limited extent. Indeed, the 5GT-MTP exposes to it only the *virtual* links and their characteristics. The 5GT-SO will therefore be confined to checking that the traffic on virtual links does not exceed their capacity, and that the delays they introduce does not jeopardize service time requirements.

6.1.5 Constraints

There are four main constraints the 5GT-SO has to honor, summarized in Table 7 along with the decisions they have an impact on. The first three are fairly straightforward, and amount to ensuring that the capabilities of hosts are not exceeded, that each VNF is assigned the resources it needs to properly work, and that links between hosts are not overloaded.

The fourth constraint deals with the service time, an important KPI for most services. This constraint is more complex to check and satisfy, as the request service time includes two main components, namely:

- The processing time, i.e., the time that requests spend being processed at one of the VNFs;
- The network delay, i.e., the sum of the delays of the virtual links traversed by the request.

TABLE 7: MAIN CONSTRAINTS THE 5GT-SO HAS TO HONOR

Constraint	Decisions it affects	Description
Host capabilities	Placement	The sum of the resources assigned to each VNF must not exceed the capabilities of the host running them. Must be checked for each type of resource (memory, CPU…) separately.
Minimum VNF resources	Placement, assignment	Each VNF must be assigned at least the minimum amount of resources it requires. Must be checked for each type of resource (memory, CPU…) separately.
Link capacity	Placement, routing	The total traffic going from any VNFs hosted at host h_1 to any VNF hosted at host h_2 must not exceed the capacity of the link between h_1 and h_2.
Target service time	Placement, assignment, routing	The *total* service time requests incur must be lower than the maximum delay specified by the KPIs.
Other KPIs	Potentially: placement, assignment, routing	Any additional, service-specific KPIs must be satisfied

As pointed out in Table 7, all the decisions the 5GT-SO has to make impact one of the two components of the service time, and therefore whether the service time constraint is honored or not.

Finally, the last row of Table 7 refers to additional, service-specific KPIs; such KPIs also translate into additional constraints.

6.1.6 Objective

Any combination of decisions about VNF placement, CPU assignment and routing satisfying the constraints described in 6.1.5 will result in a service instance conforming to the requirements of the vertical. However, not all such combinations are equal from the viewpoint of the operator, as they will use different parts of its network to a different extent. Among these combinations, the 5GT-SO is therefore free to choose the one that optimizes such goals as network usage, power consumption, or cost.

A straightforward option is to minimize the number of *active* hosts, i.e., the number of hosts that are assigned at least one VNF. This leaves some of the resources exposed by the 5GT-MTP unused, with a twofold benefit: first, a lower power consumption and lower costs; second, the possibility of using those unused resources to serve a subsequent network service instantiation request. On the negative side, this could make scaling decisions more complex to make and enact.

6.1.7 Service Orchestration Algorithms

The task of the 5GT-SO can be viewed as setting the decision variables in 6.1.4 in order to optimize the objective in 6.1.6, subject to the constraints in 6.1.5. The most straightforward approach to perform such task is formulating an optimization problem and solving it through one of many solvers available, both commercial (e.g., Gurobi and CPLEX) and open-source (e.g., GLPK) [62][63][64].

An optimization formulation of the orchestration problem, accounting at the same time for the VNF placement, resource assignment and traffic routing decisions, would have:

- Convex constraints (if, as it is commonly done, VNFs are modeled as queues, the expression of the processing time at VNFs is of type $\frac{1}{1-x}$, which is convex but not linear);
- A mixture of binary and real variables (in particular, placement decisions are expressed through binary variables, as discussed in 6.1.4).

Problems of this type are prohibitively complex to solve. It is well known that MILP (mixed-integer linear programming) problems are NP-hard, i.e., their worst-case time complexity grows exponentially with the instance size. The orchestration problem is harder than MILP optimization, since its constraints are convex and not linear.

Besides its implications on worst-case complexity, the fact that orchestration problems are not even MILP also means that the many popular solution strategies aimed at MILP problems (most notably, branch-and-cut and cutting-plane approaches) are ruled out. In summary, directly solving the orchestration problem through optimization is not a viable approach.

6.1.7.1 Relaxation

Relaxation is a solution strategy used when dealing with binary optimization problems, based on replacing binary variables with continuous (real) ones. Specifically:

1. The binary variables are replaced by real variables, bounded between 0 and 1;
2. The resulting problem is solved;
3. The optimal values of the relaxed variables are used to assign a value to the original (binary) variables.

In the orchestration case, the relaxed problem in step 2 above would be convex, and therefore feasible to solve in a short time. However, using the relaxed variables to assign a value to the original variables (step 3) can result in solutions that are not only suboptimal, but indeed infeasible.

Suppose, for example, that we decide to set to one the binary variables A(h,v) whose relaxed counterpart a(h,v) is above 0.5, and to zero the others. However, there is nothing guaranteeing that there will be an a(h,v)-value greater than 0.5 for every VNF v, which would result in necessary VNFs not being present.

6.1.7.2 Decoupled strategy

As straightforward optimization approaches are not an option, we devise a different solution strategy, based on three pillars:

- *Decoupling* VNF placement decisions from resource assignment and traffic routing ones;
- Making placement decisions *sequentially*, one VNF per iteration, without deciding a priori the order in which VNFs are placed;
- Within each iteration, solve a relaxed (convex) problem to obtain *guidance* about the placement decisions, rather than to directly set all decision variables.

The idea of decoupling VNF placement decisions from the others comes from the observation that, if placement decisions were given, then the problem of resource assignment and traffic routing would become much simpler, namely, convex. Therefore, we employ a placement heuristic, described in 6.1.7.3, to make placement decisions. Given those, we simply solve a convex problem to find the optimal resource allocation and traffic routing.

6.1.7.3 Placement heuristic

Our VNF placement heuristic is summarized in Figure 20. At the first iteration, it solves a convex problem where all placement variables A(h,v) are replaced with their relaxed counterparts a(h,v). Then, we identify the VNF v* and the host h* with the highest value of such variable, and place VNF v* at host h*. If there are more VNFs to place, the heuristic continues with a new iteration, solving a new relaxed problem where:

- The a-variables corresponding to VNFs that have not been placed are relaxed;
- The a-variables corresponding to VNFs that have been placed are fixed, either to one (for the host the VNF is placed at) or to zero (for all other hosts).

After all VNFs are placed, the heuristic terminates.

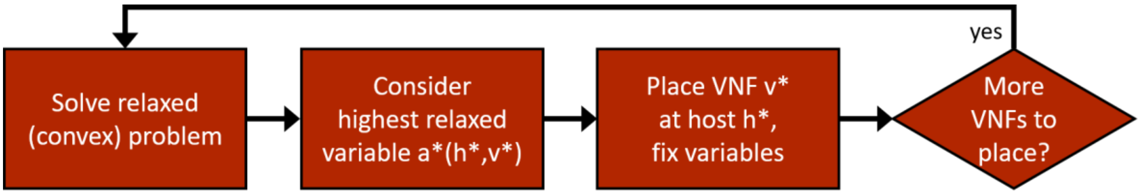

yes
Solve relaxed (convex) problem
Consider highest relaxed variable a*(h*,v*)
Place VNF v* at host h*, fix variables
More VNFs to place?

FIGURE 20: DECOUPLED VNF PLACEMENT HEURISTIC

There are two aspects of our placement heuristics that make it especially effective. The first is usage of relaxed problems: relaxed variables are not interpreted as proxies of their binary counterparts, but rather as guidance for placement decisions. The intuition is that higher values, i.e., closer to one, correspond to a higher confidence that a particular VNF shall be placed at a particular host.

Perhaps more important is how decisions are made across iterations. The order in which VNFs should be placed is not decided a priori, which allows us to place first the VNFs for which the relaxed problem provides the strongest indications, as described earlier. Furthermore, by fixing the variables corresponding to previously-made decisions, we are able to account for their effect, e.g., the resources VNFs consume at the host they are placed at. This results in decisions that are guaranteed to be feasible, unlike the traditional relaxation approaches discussed in 6.1.7.1.

As far as complexity is concerned, our placement heuristic runs for as many iterations as there are VNFs to place, and during each iteration exactly one convex optimization problem is solved. Considering that convex optimization problems have polynomial (namely, cubic) time complexity in the number of variables, we can conclude that our placement heuristic - and our decoupled solution strategy as a whole - have the potential to be used in real-world 5GT-SO implementations. Other approaches are currently being investigated, based on creating clusters on both the VNF and the host graphs, and then matching such clusters together.

6.2 Federation Algorithms

In the following, we present a brief overview of service/resource abstractions and high-level procedures that should be handled by 5G-SO's algorithmic framework. We note that this is work in progress; the complete design of such framework will be included in the next deliverable.

6.2.1 Service and resource abstractions

The federation levels play a key role in the decision on the amount of information that can be shared among different administrative domains. The higher the mutual trust or federation level, administrative domains tend to share more detailed information about their services or resources. Therefore, to hide some information (parameters regarding vendor, location, connections, hardware configurations, etc.) the 5G-TRANSFORMER system applies resource and service abstraction.

For the services, abstractions (i.e., lower level of details) are applied already in the pre-phase, during the business negotiations. All information is hidden from a peering administrative domain. Once the provider 5GT-SO NFVO-NSO enables a federation of NFV network service to a consumer SO, during the active phase it provides monitoring information with abstracted parameters and indicators. These abstractions do not reveal the underlying NFVI infrastructure, VNFs, PNFs, etc., that are enabling the federated NFV network service. For example, if a peering domain has a higher federation level (e.g. platinum) the offered NFV-NS would hide only some very detailed parameters (like number of VNFs, number of CPUs, etc.), but expose more monitoring parameters such as CPU utilization, storage capacity, etc. Whereas in terms of lower federation level (e.g. bronze), the level of revealed details is quite low, providing monitoring only on a key parameter e.g. bandwidth, latency, number of users, etc. The parameters exchanged between peering administrative domains are those already settled in the business/level agreements.

For the federation of resources, the abstraction is applied twice. First, the 5G-MTP applies resource abstraction on the underlying NFVI infrastructure [11]. The abstraction applied in the 5GT-MTP Resource Abstraction block is applied to hide some details of NFVI resources from the local SO. The second abstraction is applied upon calculation of federation capabilities in the 5GT-SO NFVO-RO. The 5GT-SO NFVO-RO reads information of available resources in the NFVI database (which are already generated and stored abstractions from the 5GT-MTP) and performs calculation. The outcomes of the calculations are categorized in federation levels. The categorized information is sent to the SO-SO Advertisement block. This block broadcasts the categorized resource abstractions to peering SOs according to the established federation level.

6.2.2 Pre-configured and dynamic (centralized or distributed) federation

In this section, it is assumed pre-configured federation between administrative domains. Each administrative domain first establishes business/service agreements with peering administrative domain. Then all the network connections are manually created. The catalogues of services are established on business face-to-face meetings and well-defined before any networking connection is established. In terms of the resource, the process is more dynamic. The general terms and conditions are agreed and the agreement between peering administrative domains is to maintain certain federation level when federation of NFVI resources is enabled. The offering or the list of available NFVI resources for federation is dynamically generated and regularly updated.

Another case of federation is when the whole process is dynamically created. The complete dynamical federation would consist of sharing service level agreements, establishing connections, initiating federation services or resources, lifecycle of federated services or resources. The realization of the dynamic approach can be centralized or distributed.

For centralized dynamic federation, all the 5GT-SOs of each administrative domain have to maintain secure connection to a central authority or central registry. This secure connection could be established through reference points So-So-CAT. The central registry is an aggregator of offered list of NFV network services, list of NFVI resources and list of service agreements. Each 5GT-SO upon establishing connection to the central registry would be responsible of providing the three kinds of information to the registry and regularly updating them. For requesting federated services or resources, the consumer 5GT-SO sends requests directly to the central register. The central register can be responsible for establishing the rest of the federation procedure or it can just deliver information to the provider 5GT-SO and the consumer 5GT-SO allowing them to proceed with the federation in a peer-to-peer manner. The advantage of this approach is that the connections are dynamically created and fast deployment can be established by having a single point where the global view of available resources is generated. However, the drawback is that the central registry is a single point of failure (the decentralized solution can solve this by replication), there is a higher cost for maintaining these information (storage, connections and data centers) as well as applying additional security protocols (note that each peer could decide on its own security level) hence higher communication overhead.

In a distributed dynamic approach, all the 5GT-SOs of each administrative domain create peer-to-peer network in which they begin by establishing network connections with 5GT-SOs that have already established business relationship. Upon establishing the pre-defined network connections, the 5GT-SOs can dynamically exchange list of peering 5GT-SOs that are participants in the network and allowed to share their information. Each 5GT-SO would maintain global view of the available resources and regularly would sync the global view to the rest of the network. The updates related to the availability of the resources or services are broadcasted by each administrative domain/SO. Establishing connection and negotiating terms is dynamically established as in the centralized case. The difference is that each 5GT-SO should handle distribution of the information that circulates throughout the network and contribute by broadcasting neighboring advertisements to all connections in the network. Another solution that can be adopted is using Blockchain technology [60], where the whole eco-system can be easily maintained and managed. The updates of available services and resources can be well ordered and the global view can be easily maintained. Although this approach has the same benefits as in the centralized case and less cost referring maintenance of data storage and security, still it brings high overhead and complex federation scenarios which is not easy to adopt. This solution will be addressed in more detail in future deliverables.

7 Conclusions

This deliverable has introduced both business and functional requirements to design 5GT-SO. As a result, we presented the internal architecture of the 5GT-SO, in addition to its interfaces to the remaining components of the 5G-TRANSFORMER system, namely the northbound interface to the 5GT-VS, the southbound interface to the 5GT-MTP and the east/westbound interface to service orchestration from different domains.

This deliverable also introduces a monitoring service that 5GT-SO exposes to all the components in the 5GT-TRANSFORMER architecture. In the particular case of 5GT-SO, monitoring reports from this platform can be used by the 5GT-SO to, e.g., validate SLAs or as triggers to auto-scaling procedures. In case of 5GT-VS, the monitoring reports produced by the 5GT-SO Monitoring Service can be also provided to the through the Vs-So(-MON) reference point.

In order to illustrate the internal operation of 5GT-SO we have selected a set of representative procedures and we have described the internal workflow in 5GT-SO to execute them. Namely, the selected cases are (1) Service On-boarding, (2) Service Instantiation, (3) Service Modification, (4) Service Termination, (5) Service Assurance, (6) Service Federation, (7) Resource Federation, and (8) two MEC-related procedures: AppD on-boarding and NFV-NS instantiation when including MEC applications.

At the core 5GT-SO, a framework of optimization and decision algorithms will assist in making appropriate decisions in each of the operations supported by 5GT-SO, including service orchestration and federation operations. An overview of this framework has been introduced in this deliverable.

In the course of the project, the most important operations will be implemented [14], [15] and eventually demonstrated [13]. In particular, we aim to demonstrate that the 5GT-SO creates, instantiates and properly manages network services requested by 5GT-VS. To this aim, an appropriate utilization of different NFVIs by means of 5GT-MTPs will also be demonstrated.

8 References

[1] A. de la Oliva *et al.*, "5G-TRANSFORMER: Slicing and Orchestrating Transport Networks for Industry Verticals", IEEE Wireless Communications, to appear.

[2] X. Li *et al.* "Service Orchestration and Federation for Verticals", The First Workshop on Control and management of Vertical slicing including the Edge and Fog Systems (Co-located with IEEE WCNC 2018), April 2018.

[3] L. Valcarengui *et al.*, "Orchestration and Federation of Multi-Domain 5G Transport Networks", European Conference on Networks and Communications (EuCNC) 2018, June 2018.

[4] C. J. Bernardos, A. De La Oliva, F. Giust, JC. Zuniga, A. Mourad, "Proxy Mobile IPv6 extensions for Distributed Mobility Management", September 2018, IETF draft (work-in-progress), draft-bernardos-dmm-pmipv6-dlif-01.

[5] C. J. Bernardos, *et al.*, "Multi-domain Network Virtualization", September 2018, IETF draft (work-in-progress), draft-bernardos-nfvrg-multidomain-04.

[6] L. Geng, *et al.*, "COMS Architecture", September 2018, IETF draft (work-in-progress), draft-geng-coms-architecture-02.

[7] L. Geng, *et al.*, "Problem Statement of Common Operation and Management of Network Slicing", September 2018, IETF draft (work-in-progress), draft-geng-coms-problem-statement-03.

[8] 5G-PPP, "5G: Serving vertical industries", https://5g-ppp.eu/2nd-5g-vertical-workshop/.

[9] 5G-TRANSFORMER, D1.1, Report on vertical requirements and use cases, November 2017.

[10] 5G-TRANSFORMER, D1.2, 5G-TRANSFORMER initial system design, May 2017.

[11] 5G-TRANSFORMER, D2.1, Definition of the Mobile Transport and Computing Platform, March 2018.

[12] 5G-TRANSFORMER, D3.1, Definition of vertical service descriptors and SO NBI, March 2018.

[13] 5G-TRANSFORMER, D5.1, Definition of vertical testbeds and initial integration plans, May 2018.

[14] 5G-TRANSFORMER, D4.2, Initial Service Orchestrator Reference Implementation, November 2019.

[15] 5G-TRANSFORMER, D4.3, Final design and implementation report on service orchestration, federation and monitoring platform (including reference implementation), May 2019.

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

[17] ETSI GS NFV-IFA 005, "Network Function Virtualisation (NFV); Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification", v2.1.1, 2016.

[18] ETSI GS NFV-IFA 006, "Network Function Virtualisation (NFV); Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification", v2.1.1, 2016.

[19] ETSI GS NFV-IFA 007, "Network Function Virtualisation (NFV); Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification" v2.1.1, 2016.

[20] ETSI GS NFV-IFA 008, "Network Functions Virtualisation (NFV); Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification", v2.1.1, 2016.

[21] ETSI GS NFV-IFA 010, Management and Orchestration; Functional requirements specification", v2.4.1, 2018.

[22] ETSI GS NFV IFA 11, "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; VNF Packaging Specification" v2.3.1, August 2017.

[23] ETSI GS NFV IFA 13, "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification" v2.3.1, August 2017.

[24] ETSI GS NFV IFA 14, "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Network Service Templates Specification" v2.3.1, August 2017.

[25] ETSI GR NFV-IFA 022, Management and Orchestration; Report on Management and Connectivity for Multi-Site Services", v0.8.2, 2018.

[26] ETSI GS NFV-IFA 028, Management and Orchestration; Report on architecture options to support multiple administrative domains", v3.1.1, 2018

[27] ETSI GS MEC 010-2, "Mobile Edge Computing (MEC); Mobile Edge Management; Part 2: Application lifecycle, rules and requirements management", v1.1.1, July 2017.

[28] ETSI GR MEC 017, "Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV Environment", v1.1.1, February 2018.

[29] ETSI GR NFV-EVE 012 V3.1.1, Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework, 2017.

[30] ETSI GS NFV-INF 003, Infrastructure; Compute Domain", v1.1.1, 2014.

[31] ETSI GS NFV-INF 004, Infrastructure; Hypervisor Domain", v1.1.1, 2015.

[32] ETSI GS NFV-INF 005, Infrastructure: Network Domain, V1.1.1, 2014.

[33] 5GEx deliverable D2.1, public version "Initial System Requirements and Architecture, available at https://drive.google.com/file/d/0B4O_JVjsvab9VXItUDJqMUR6d28/view

[34] B. Chatras, U. S. Tsang Kwong and N. Bihannic, "NFV enabling network slicing for 5G," 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, 2017, pp. 219-225.

[35] TeleManagement Forum. "TeleManagement Forum Information Framework (SID): GB922_Addendum_4SO_Service_Overview_R9-5_v9-7".

[36] 3GPP TR28.801. Telecommunication management; Study on management and orchestration of network slicing for next generation network.

[37] 3GPP TS23.501, V15.0.0, System Architecture for the 5G System; Stage 2, 2017.

[38] 3GPP TS 38.300, "NR; NR and NG-RAN Overall Description; Stage 2; (Release 15)", v15.0.0, December 2017.

[39] 3GPP TS28.530, V0.4.0, Telecommunication management; Management of 5G networks and network slicing; Concepts, use cases and requirements, 2017.

[40] 3GPP TR 28.531, "Management and orchestration of networks and network slicing; Provisioning; Stage 1 (Release 15)", v0.3.0, February 2018.

[41] Xi Li *et.al.,* "Service Orchestration and Federation for Verticals", 1st Workshop on Control and management of Vertical slicing including the Edge and Fog Systems (COMPASS), Barcelona, IEEE, 2018.

[42] P. Iovanna *et.al.*, "5G Mobile Transport and Computing Platform for verticals", 1st Workshop on Control and management of Vertical slicing including the Edge and Fog Systems (COMPASS), Barcelona, IEEE, 2018.

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

[44] Costa-Pérez, Xavier; García-Saavedra, Andrés; Li, Xi; Deiss, Thomas; Oliva, Antonio de la; Di Giglio, Andrea; Iovanna, Paola, and Mourad, Alain. 5G-Crosshaul: An SDN/NFV Integrated Fronthaul/Backhaul Transport Network Architecture. IEEE Wireless Communications, 24(1), pp. 38-45, February 2017

[45] Vision on Software Networks and 5G - https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP_SoftNets_WG_whitepaper_v20.pdf

[46] Open Source MANO. https://osm.etsi.org/

[47] RIFT.ware. https://www.riftio.com/

[48] Juju. https://www.ubuntu.com/cloud/juju

[49] ONAP Architecture overview - https://www.onap.org/wp-content/uploads/sites/20/2017/12/ONAP_CaseSolution_Architecture_120817_FNL.pdf

[50] TOSCA DSL specification - https://www.oasis-open.org/committees/tosca

[51] Apache Aria project - http://ariatosca.incubator.apache.org/

[52] Cloudify project repository - https://github.com/cloudify-cosmo

[53] OpenBaton project - https://openbaton.github.io/

[54] https://spring.io/

[55] https://www.postgresql.org/

[56] https://www.rabbitmq.com/

[57] SONATA. Deliverable 2.2: Architecture and Design, 2015.

[58] 5G-TANGO. Deliverable 2.2: Architecture and Design, 2017.

[59] L. Cominardi, "Multi-domain federation: scope, challenges, and opportunities," Workshop in 3rd IEEE Conference on Network Softwarization, Bologna, Italy, July 2017.

[60] T. Swanson, “Consensus-as-a-service: A brief report on the emergence of permissioned, distributed ledger systems,” Technical Report, Apr. 2015. [Online]. http://www.ofnumbers.com/wp-content/uploads/2015/04/. Permissioned-distributed-ledgers.pdf. [Accessed: 06 Mar 2018].

[61] Luis M. Contreras and Diego R. López, “A Network Service Provider Perspective on Network Slicing”, IEEE Softwarization, January 2018. Available online: https://sdn.ieee.org/newsletter/january-2018/a-network-service-provider-perspective-on-network-slicing

[62] The Gurobi Solver, https://gurobi.com

[63] ILOG CPLEX Optimization Studio, https://www.ibm.com/products/ilog-cplex-optimization-studio

[64] GNU Linear Programming Kit, https://www.gnu.org/software/glpk/

[65] B. Sonkoly, *et.al., “*Multi-Domain Service Orchestration Over Networks and Clouds: A Unified Approach”, SIGCOMM Comput. Commun. Rev. 45, 4 pp 377-378. August 2015.

[66] Guerzoni *et al.*, “Analysis of end-to-end multi-domain management and orchestration frameworks for software defined infrastructures: an architectural survey”. Trans. Emerging Tel. Tech. April 2017.

9 Annex I. Vertical Services

The 5G-TRANSFORMER consortium includes partners from several of the Vertical Industries identified in the market portfolio of the 5G-PPP [1], namely: automotive, entertainment, healthcare and manufacturing and also representatives from the Mobile (Virtual) Network Operator (MNO/MVNO) industry. The following sections summarize the use cases (UC) established in D1.1 [9], and detail the expectations within the 5G-TRANSFORMER project in order to ease the reading of the rest of the document.

9.1 Automotive

The automotive industry is currently undergoing key technological transformations, as more and more vehicles are connected to the Internet and to each other, and advances toward higher automation levels. In order to deal with increasingly complex road situations, automated vehicles will have to rely not only on their own sensors, but also on those of other vehicles, and will need to cooperate with each other, rather than make decisions on their own.

These trends pose significant challenges to the underlying communication system, as information must reach its destination reliably within an exceedingly short time frame - beyond what current wireless technologies can provide. 5G, the next generation of mobile communication technology, holds promise of improved performance in terms of reduced latency, increased reliability and higher throughput under higher mobility and connectivity density.

Vehicle domain features differ across the target operative scenarios which are strongly characterized by their own peculiarities. In order to better analyse the needs of the automotive domain versus the incoming communication technology, we considered four main scenarios (urban, rural, highway and transversal) and several use cases quite different for their peculiar features outlining the key aspects that mostly impacts on 5G.

Typical automotive UCs are various and can address heterogeneous domains. In D1.1 [9] more than 25 UCs from those most popular in the literature have been described; the identified UCs are grouped in 6 domains: safety, mobility, entertainment, e-road, digitalized Vehicles and automated vehicle.

In the 5G-TRANSFORMER project, we focus in the safety domain where, thanks to 5G capabilities, the vehicle can outline/foresee dangerous situations and properly react on time. In particular, two use cases have been selected and proposed for implementation:

TABLE 8 AUTOMOTIVE USE CASES

ID	Goal In context	General description
UC A.01, UC A.02	Avoid possible collision crossing intersection	The purpose of the Intersection Collision Avoidance (ICA) system is to alert drivers about the existence of any possible obstacles and eventually activate the emergency braking system. The communication infrastructure facilitates a real-time exchange of data between the involved entities.

UC A.04	Vehicles are able to see through obstacles, thanks to cooperation among them achieving bilateral awareness of road conditions	Thanks to the cooperation between vehicles, streaming information is provided to all the vehicles that want/need to access to it. This information can be used to identify potential obstacles that cannot be detected through on-board sensors.

9.2 Entertainment

The Media and Entertainment (M&E) industry is one of the industries most affected by the deep changes in terms of user habits and expectations that the society has been experiencing with the explosion of Internet. The amount of users grows daily and the users demand progressively media-rich contents and a better quality of experience.

While all these changes provide a great economical fuel for the industry, they also impose challenges to the network infrastructures (in terms of data rates, number of connections, quality of experience, etc.) not present before. The 5G PPP already identifies the entertainment industry as one of the key interested parties. This is because one of the key objectives of the 5G is to open operators' networks for new services, and this is the key enabler to support the data rates and the latency required to give an immersive experience. Furthermore, the 5G also aims to provide the services with network information not available before (i.e. packet losses, signal level, etc.) to better adapt the service to the network conditions.

The 5G-TRANSFORMER project focuses in the M&E services particularly targeting sports events. The aim is to encompass these services to the FAN ENGAGEMENT trend, which envisions smarter venue services by means of providing targeted and high-quality content and following fans along the journey with contextualized information. This trend also envisions fans as content producers (i.e. to share videos, photos, emotions, opinions, comments, etc.), and captures the explosion of IoT devices by including them as additional content producers. The final goal is to give the fans a more interactive, immersive and participative experience like never before.

The following use cases are considered in the project, in order to address the needs for the different actors and scenarios identified in [9] for the Entertainment vertical industry:

TABLE 9: ENTERTAINMENT USE CASES

ID	Goal In context	General description
UC.E01	To provide a better fan experience to users attending (on-site) an event	Large-scale event sites, such as stadiums are more and more being connected in order to give a better experience to their customers (replay, choose a specific camera, language, augmented reality to bring additional information, etc.)

9.3 eHealth

The eHealth use case is one of the most critical verticals we have in the 5G-TRANSFORMER project. This industry can effectively take advantage of the future 5G networks to improve the quality of life and medical assistance of people in emergency situations. It aims to be able to detect and assist people in emergencies in the minimum possible time in order to assure the maximum probability of people passing the emergency and recover from it.

5G networks will be able to support high demands of traffic with low delay requirements. Thus it is very helpful for the eHealth use case because that allows discovering and attending emergencies in short time. Hence, there are two main targets for this use case: e-Infrastructure and eHealth application.

On the one hand, the e-Infrastructure use case focuses on how the current municipality infrastructure based on Terrestrial Trunked Radio (TETRA) can be replaced based on the 5G features. This will allow emergency alarms to be received with smaller delay and thus, be processed in a small amount of time to send an ambulance to the place of the emergency. This will also allow to access in real time the clinical history of the patient from the place of the incident to give the patient a better medical attention. In addition, to have a better e-Infrastructure, the eHealth use case will need a high-priority and low latency service in the 5G-TRANSFORMER system. To address that, the 5G-TRANSFORMER system will allow to have access to the resources of the eHealth slice in extreme cases where the network is overloaded by users like in big events.

On the other hand, the eHealth application aims to study how new technologies such as Multi-access Edge Computing (MEC) can help improving the speed of response. This application tries to reduce the response time and automate processes of communicating between the patient and the medical personnel and among the medical personnel. The idea is to have an application based on MEC for automating the collection of data from wearables, detection of problems and automatic calling the ambulance, which requires mechanisms for patient feedback (call back). If possible, it is important to provide video feed between emergency team and doctor at the hospital because the personnel in the ambulance is not specialized to deal with in some emergencies such as an urgent surgery in the most extreme case. In that case, they need to contact a doctor that monitors and guides the process over real-time 4K video.

TABLE 10: EHEALTH USE CASES

ID	Goal In context	General description
UC.H01	To provide a better medical assistance in emergency cases	Large-scale event sites where a lot of groups are deployed to cover the emergencies and have to communicate between them in real time. Emergencies that requires real time communication between the ambulances and doctors. Improvement of the current infrastructure to guarantee the real time exchange of information to detect early the emergencies.

9.4 eIndustry

The production and manufacturing industry is currently undergoing important changes mainly driven by the ongoing introduction of new emerging technologies, including mobile network, cloud computing, robotics, machine intelligence and big data. Nowadays we are facing a new industrial revolution, commonly referred to as Industry 4.0, whose aim is to provide mass customization with costs comparable to those of mass production. This can be achieved leveraging on full digitalization and automation of industrial processes.

The major ingredient to ensure full digitalization and automation is the virtualization of control, allowing to centralize all the intelligence of the operations in order to increase flexibility and facilitate the changes of the manufacturing plants. Moreover, it is essential to monitor all the elements of an industrial manufacturing plant through wireless connectivity (in order to avoid cabling that further increases complexity) and information processing (including big data and analytics technologies). These enhanced functionalities introduce strict requirements on data rates, latency, reliability, etc., all of which are addressed in the 5G mobile transport and computing platform.

The role of 5G in Industry 4.0 extends to large area logistics (i.e. in the optimization of maritime, ground, air transportations, as well as to optimize port operations and goods production processes), where there is a similar need to increase the productivity and the efficiency of the processes to cut production costs and become more and more competitive.

In [9] several use cases have been identified for the e-Industry vertical, namely monitoring in production line, cloud robotics, automated logistics, electric power generation, electric power transmission and electric power distribution. Several uses cases can coexist in different scenarios. For example, in an automated factory both monitoring application and cloud robotics solutions can be in use. All use cases presented in [9] involve enabling more efficient manufacturing and lean production which poses severe requirements on the underlying communication network making it essential that the industrial environment be equipped with 5G solutions.

Among all the e-Industry use cases, the cloud robotics has been selected as candidate for implementation in the final demonstrators of 5G-TRANSFORMER project.

TABLE 11: EINDUSTRY USE CASES

ID	Goal In context	General description
UC I.02	Highly automation of the factory plant is provided moving the control of the production processes and of the robots functionalities in cloud, exploiting wireless connectivity to minimize infrastructure, optimize processes, implement lean manufacturing.	The controlling functionality of the robots is moved to the cloud, in order to utilize its massive computing power. Huge amounts of information will have to be transferred instantaneously. With lower latency and higher bandwidth than other forms of wireless connectivity, 5G is the optimal choice.

9.5 MNO/MVNO

Increasing the capacity and the elasticity of mobile network operators' networks is one of the most important challenges foreseen in 5G networks, as it will allow opening MNOs business toward new markets and a large variety of tailored services. This evolution is especially brought through the convergence of mobile networks and cloud infrastructures, which provides the capability for mobile operators to use network function virtualization (NFV) concepts and cloud-based infrastructures in order to virtualize and decentralize their network entities. Hence, the MVNO business model emerges from this evolution through the creation of a new business model that disrupts the traditional mobile value chain. In the MVNO model, new players can participate in the mobile value chain and extract value to leverage their valuable assets.

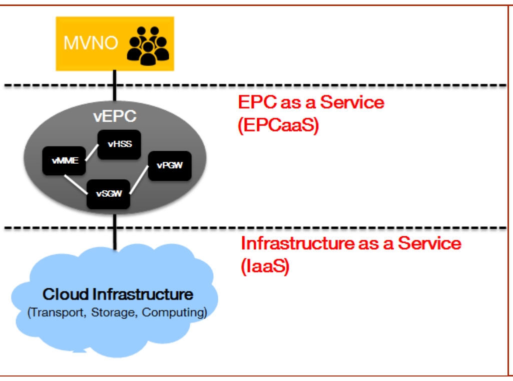
In 5G-TRANSFORMER, the MNO/MVNO industry is especially relevant and interesting because of this new role model it injects into the mobile value chain and also because of the nature of services it is offering compared to the other studied vertical industries in the project. For instance, offering the Network as a Service (NaaS) or the Infrastructure as a Service (IaaS) are types of services that are challenging for MNOs and MVNOs in order to reach real on demand and finely tailored services for their customers.

Thus, the MNO/MVNO player has a different role compared to the other verticals. In fact, the relation between the MNO/MVNO and the 5G-TRANSFORMER system depends on the chosen MVNO business model. For instance, in the case of a Full MVNO or a Mobile Virtual Network Enabler (MVNE) business model as described in [9], the role of the MVNO exceeds that of a simple vertical service provider and has almost the same role as an MNO acting as a Network service provider. Likewise, the role of the MNO hosting an MVNO is built on the offering of a Network as a Service (NaaS); for instance, the MNO would rely on network slicing combined to services like EPCaaS and IaaS in order to set up an MVNO network and provide Network services like connectivity to the MVNO. In addition, verticals can be seen as customers of an MNO or an MVNO.

In [9], several use cases have been identified as relevant for the MNO/MVNO domain in 5G-TRANSFORMER. We chose to focus here on the use case UC M.01 vEPCaaS. This use case especially describes how the MNO/MVNO can offer Network as a Service (NaaS) for its customers by offering a dedicated and on-demand core network. In the same context of offering Network as a Service, we also investigate the particular case of NFVIaaS as an example of IaaS. In this particular case of IaaS, the challenge resides in the fact that the correspondent MVNO business model may imply the ownership by the MNO/MVNO of an OSS/BSS that provides him with the capability to create and configure its own network slice instances for its customers and the non-ownership of an NFV infrastructure. In this case, 5G-TRANSFORMER will offer NFVIaaS for the MNO/MVNO. One possible question in this case is whether it is possible for an MNO/MVNO to request a network slice with particular Network Functions Virtualisation Infrastructure (NFVI) resources and through which interface or service catalogue would this be possible.

TABLE 12: MNO/MVNO USE CASES

ID	Goal in context	General description
UC M.01	Build of an MVNO service through the deployment and operation of a network slice with a vEPC in “as a Service” mode.	The vEPC can be instantiated as a virtualized Control plane only or as a complete virtualized Control and User planes core network. The vEPC is supposed to provide the same implementation and performances of a real EPC that is deployed on a real infrastructure. The use of a vEPC should be totally transparent and should not impact services’ end

	 MVNO vEPC vHSS vMME vPGW vSGW EPC as a Service (EPCaaS) Infrastructure as a Service (IaaS) Cloud Infrastructure (Transport, Storage, Computing)	to end latency.

10 Annex II. Monitoring Requirements

The objective of the 5G-TRANSFORMER monitoring services is to generate monitoring data in support of the management of the vertical services at the run-time. Therefore the 5GT-SO Monitoring Platform is in charge of elaborating performance metrics and failure events with an end-to-end scope and aggregating data at different hierarchical levels, from virtual resources and infrastructures up to VNFs and network services. The resulting monitoring reports can be used internally at the 5GT-SO, e.g. for SLA validation or to evolve the lifecycle of the network services, or can be provided to the 5GT-VS for further elaboration at the network slice or vertical service level.

In general, in a typical vertical service delivered by the 5G-TRANSFORMER system, we can distinguish two main types of monitoring data:

- The **vertical application monitoring data**, generated by and used by the application(s) running on the vertical service resources. An example could be the cars' monitoring data exchanged from the vehicles to the collision avoidance application running in the MEC, for the automotive use case. This kind of data is usually managed through vertical applications instantiated for and managed directly by the vertical, adopting proprietary algorithms (e.g. the collision avoidance algorithm) and dealing with private data (e.g. speed, position, etc.). It is important to highlight that this type of monitoring is handled directly by the vertical and it is fully transparent for the 5G-TRANSFORMER system. As such, it is considered out of scope and it will not be considered in this deliverable.
- The **monitoring data related to NFV network services, network slices and vertical services**, originally related to monitoring parameters concerning the physical infrastructures and the virtual resources where the vertical service is running, and describing their condition, usage, availability, load, etc. Starting from these elementary data, aggregation and correlation procedures at the 5GT-SO Monitoring Platform can generate performance metrics or failure alarms that are associated to the VNFs and the NFV network services implementing the network slice(s) where the vertical service is instantiated. As such, they can be used as input to take decisions about the service lifecycle management at the 5GT-SO or about the arbitration among multiple services at the 5GT-VS. The elaboration of this kind of monitoring metrics is driven by monitored data and auto-scaling rules specification in VNFDs or NSDs and it typically operates on raw monitoring data collected by the 5GT-MTP Monitoring Service from the VIMs or retrieved directly from the VNFs through the VNFMs. In the following, we will consider this kind of monitoring, with VNFs and network services monitoring as specific target of the 5GT-SO Monitoring Platform and network slice and vertical service monitoring as target of the 5GT-VS Monitoring Platform.

Since the 5G-TRANSFORMER Monitoring Platform has the objective of supporting the management of the vertical services at runtime, the project has designed the 5G-TRANSFORMER monitoring framework starting from the monitoring requirements of the use cases analysed in WP1, which are described in the following sub-sections. Starting from this analysis, we have generalized the requirements for the entire monitoring platform and we have derived a high level design of the monitoring framework compatible with the main architectural principles and business roles defined for the whole 5G-TRANSFORMER system. This monitoring framework is described in section 4.7.

In order to define the functionalities, features and target metrics of the 5G-TRANSFORMER Monitoring Platform, we have adopted a common methodology to identify the monitoring-related requirements for all the 5G-TRANSFORMER use cases, with particular focus on "which kind of monitoring data" and "which kind of processing" are needed. In details, the analysis has concerned the following aspects:

- Which "raw" monitoring data are required for the use case?
- Which is the source of the raw monitoring data?
- Which format, amount, rate?
- How is the monitoring data collected? E.g. through polling, periodical notifications, event-based notifications, etc.
- How can we aggregate and process the raw monitoring data and in which time intervals?
- How long the monitoring data must be stored?
- Which are the consumers of the monitoring data? E.g. internal decisions at the 5GT-SO or at the 5GT-VS or at the vertical service? How will the post-processing monitoring data be provided by the 5GT-SO to the 5GT-VS?

The answers to these questions will allow to identify not only the functional requirements of the monitoring platform (e.g. which parameters must be monitored and how), but also its non-functional requirements (e.g. how much storage is required? What is the frequency for notifications or polling mechanisms?). This is important to select the right technologies and provide guidelines for the system dimensioning in real deployments.

In the following, we present the result of the analysis for the different use cases.

10.1 Monitoring requirements for the automotive use case

In this section we analyze the monitoring requirements for the Extended Intersection Collision Avoidance (EXT-ICA) service. A preliminary, functional decomposition of the EXT-ICA service is shown in Figure 21, where we can identify a set of VNFs and Vas to be placed in MEC or cloud computing resources and exchanging application messages (e.g. Cooperative Information Messages - CIM - or Decentralized Environmental Notification Messages - DENM) with the vehicles or running specialized algorithms to detect possible collisions between cars.

Starting from this functional view, we have derived the NFV-based schema of the end-to-end Network Service model (Figure 22), highlighting a set of nested Network Services, their VNFs, PNFs and Vas, together with the virtual links among them and the Service Access Points (SAPs) between the nested network services and towards external networks. The 5GT-SO Monitoring Platform must be able to collect performance metrics for some of these entities and, based on their values, the 5GT-SO will need to automatically react scaling some characteristics of the VNFs or the virtual links, so that the end-to-end service can absorb the dynamicity of the car traffic in a seamless manner. Therefore, the key point is to understand the dynamicity of the service and how the processing and network load changes depend on the external conditions (e.g. the number of vehicles in a given area).

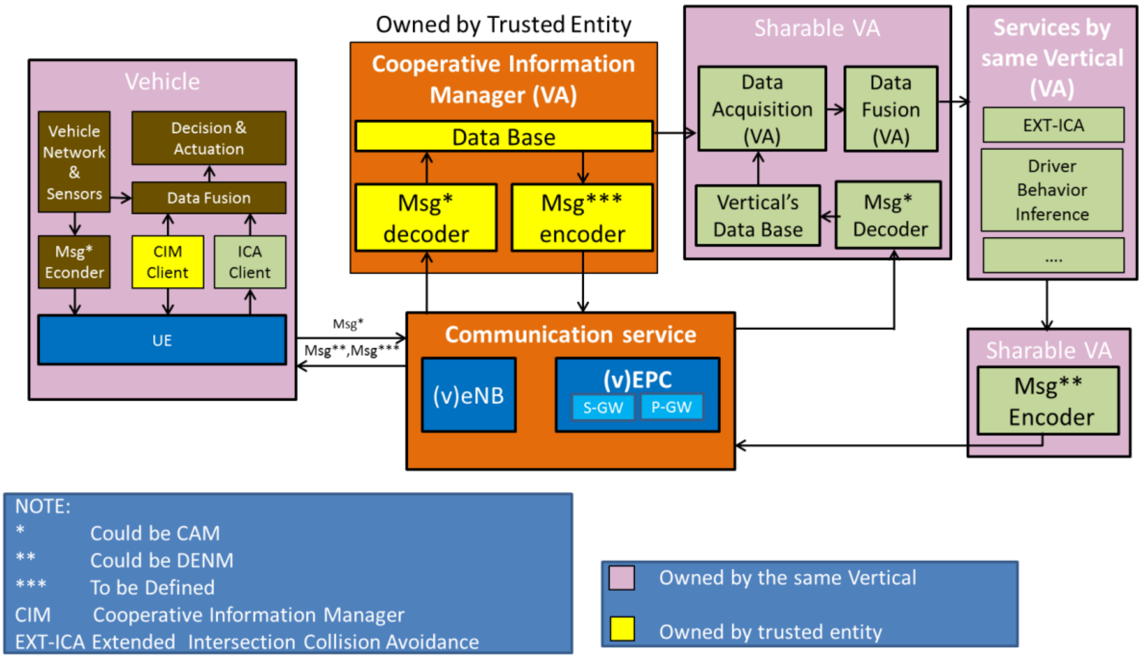


FIGURE 21: EXT-ICA SERVICE: FUNCTIONAL DECOMPOSITION

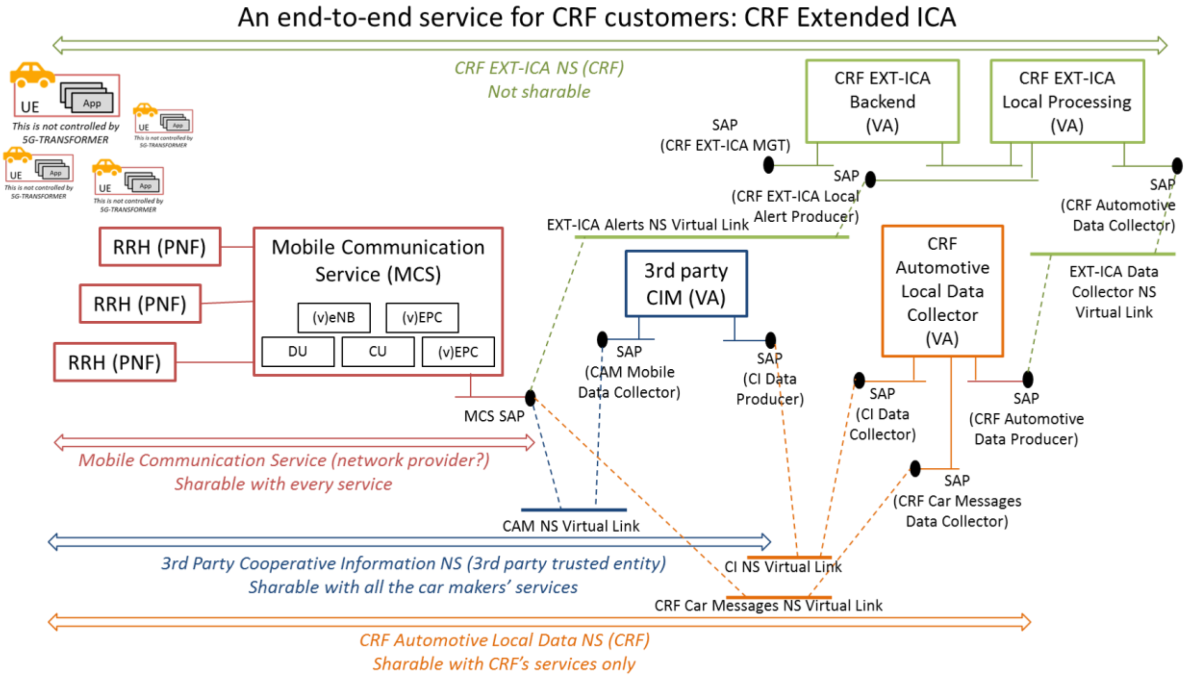

An end-to-end service for CRF customers: CRF Extended ICA
CRF EXT-ICA NS (CRF)
Not sharable
UE
App
This is not controlled by 5G-TRANSFORMER
CRF EXT-ICA Backend (VA)
CRF EXT-ICA Local Processing (VA)
SAP (CRF EXT-ICA MGT)
SAP (CRF EXT-ICA Local Alert Producer)
SAP (CRF Automotive Data Collector)
EXT-ICA Alerts NS Virtual Link
RRH (PNF)
RRH (PNF)
RRH (PNF)
Mobile Communication Service (MCS)
(v)eNB
(v)EPC
DU
CU
(v)EPC
3rd party CIM (VA)
CRF Automotive Local Data Collector (VA)
EXT-ICA Data Collector NS Virtual Link
SAP (CAM Mobile Data Collector)
SAP (CI Data Producer)
SAP (CI Data Collector)
SAP (CRF Automotive Data Producer)
SAP (CRF Car Messages Data Collector)
MCS SAP
Mobile Communication Service (network provider?)
Sharable with every service
CAM NS Virtual Link
3rd Party Cooperative Information NS (3rd party trusted entity)
Sharable with all the car makers' services
CI NS Virtual Link
CRF Car Messages NS Virtual Link
CRF Automotive Local Data NS (CRF)
Sharable with CRF's services only

FIGURE 22: EXT-ICA SERVICE: NETWORK SERVICE MODELLING

Figure 23 shows a simple deployment of the EXT-ICA service, covering five intersections under two geographical areas. The cars generate CAM messages with a frequency of 10 Hz, providing information about the vehicle's position, direction, speed, etc. Each car manufacturer can extend these messages with proprietary fields, so their size can be variable. These messages are collected at the closest MEC server, where they are stored at the CIM VA, a virtual application usually managed by a third party that collects and provides standard information from cars of all the car makers, in order to offer a full coverage for the basic service. Additional vendor specific information is instead stored in the CRF Automotive Local Data Collector VA, still running in the closest MEC server, but managed by the specific car maker. Finally, the CRF EXT-ICA algorithm elaborates the data from both the sources at the CRF EXT-ISA Local Processing VA and, in case of anomalous events, it produces DENM messages that are sent back to the CRF vehicles. Moreover, a centralized CRF EXT-ICA backend VA runs in the cloud and collects aggregated reports on alarms and accidents from all the distributed CRF's Vas.

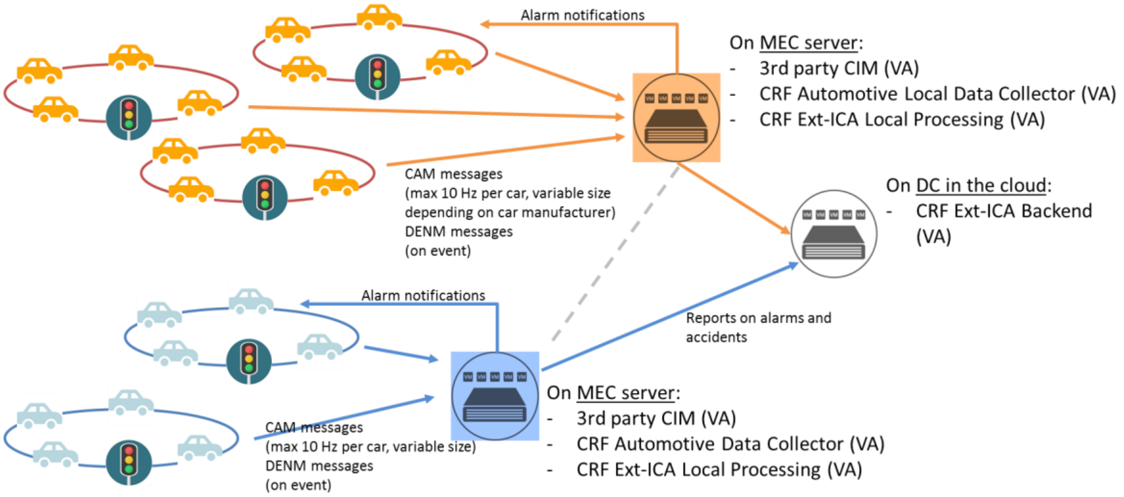


FIGURE 23: A SIMPLE DEPLOYMENT OF THE EXT-ICA SERVICE

Starting from this basic scenario, it is clear that the number of vehicles under the coverage of a MEC area has an impact on the load of the service and, in order to maintain the desired level of end-to-end performance, some of the service components need to scale up and down dynamically. In particular, an increasing number of vehicles has the following effects:

- More CAM traffic produced by the cars, with increasing bandwidth usage in uplink from RRHs to MEC server.
- More information to be processed by the EXT-ICA algorithm. This has an impact on the storage size for the 3rd party CIM and for the CRF Automotive Local Data Collector, but also on vCPU and RAM usage for the EXT-ICA Local Processing.
- More alarms may be distributed to vehicles through the DENM messages in the local or in the neighboring areas and reported to the CRF EXT-ICA backend. However, due to the asynchronous nature of these messages, the impact on bandwidth requirements should be limited.

Following this analysis, we can assume that the following monitoring metrics should be considered for the automotive use case:

- Monitoring data requested by the vertical service:

  - Number of vehicles in each MEC area covered by the service, per RRH
- Monitoring data for internal decisions at the 5GT-SO or 5GT-VS:
  - Network load in the connection from RRHs to MEC hosts
  - Storage usage in 3rd party CIM and CRF Automotive Local Data Collector Vas at MEC server
  - vCPU and RAM usage in CRF EXT-ICA Local Processing VA at MEC server
  - (Optional) Network load in the connection from MEC servers to RRHs, between MEC servers and from MEC servers to the cloud data center where the CRF EXT-ICA Backend is deployed.
  - (Optional) Storage usage in the CRF EXT-ICA Backend at the cloud data center.
  - Any failure event at the virtual infrastructure level.

The details of the monitoring metrics to be collected for the EXT-ICA service are summarized in Table 13, while a preliminary analysis about how their storage, processing, post-elaboration actions and consumers is presented in Table 14. This analysis has taken into account the requirements of the automotive use case and the characteristics and behavior of the related application, which determine the evolution of its lifecycle. This has an impact on the type, frequency and storage duration for the monitoring information that are used to trigger scaling actions. On the other hand, reporting information like the records about the alarms are mostly used for management issues and manual processing.

However, it should be noted that this analysis is highly dependent on the implementation of the EXT-ICA service applications and it should be refined at a later stage when the implementation of the applications is more mature.

TABLE 13: MONITORING METRICS FOR EXT-ICA SERVICE

Metric	Format Amount Rate	Source	Collection method	Consumer
Network load RRH→MEC host	JSON msg with int value	MTP (WIM)	Notifications on threshold + polling after alert	5GT-SO
Storage usage «3rd party CIM» & «CRF Automotive Local Data Collector»	JSON msg with % value	MTP (VIM) or VA	Notifications on threshold + polling after alert	5GT-SO
vCPU usage in «CRF EXT-ICA Local Processing» VM	JSON msg with % value	VA	Notifications on threshold + polling after alert	5GT-SO
RAM usage in «CRF EXT-ICA Local	JSON msg with int value	VA	Notifications on threshold +	5GT-SO

Processing» VM			polling after alert	
Network load MEC host → RRH	JSON msg with int value	MTP (WIM)	Notifications on threshold + polling after alert	5GT-SO
Network load MEC ←→ MEC	JSON msg with int value	MTP (WIM)	Notifications on threshold + polling after alert	5GT-SO
Network load MEC →Data Centre	N.A. (MEC to DC segment out of control)			
Storage usage in «CRF EXT-ICA Backend» VM	JSON msg with % value	MTP (VIM) or VA	Notifications on threshold + polling after alert	5GT-SO
# Vehicles in area	JSON msg from RNIS	MEC service	Polling	5GT-VS

TABLE 14: ELABORATION OF MONITORING METRICS FOR EXT-ICA SERVICE

Metric	How long to be stored	Aggregation/ Processing + time interval	Post-elaboration output	Consumer and how to retrieve
Network load RRH→MEC host	Alerts: 1 day Polling data: 1 min	Min on consecutive samples arrived in the last *X=1* minute	If (min > thr) then scale UL bw	5GT-SO, via internal event notification
Storage usage «3rd party CIM» & «CRF Automotive Local Data Collector»	Alerts: 1 day Polling data: 1 min	Min on consecutive samples arrived in the last *X=1* minute	If (min > thr) then scale storage	5GT-SO, via internal event notification
vCPU usage in «CRF EXT-ICA Local Processing» VM	Alerts: 1 day Polling data: 1 min	Min on consecutive samples arrived in the last *X=1* minute	If (min > thr) then scale vCPU	5GT-SO, via internal event notification
RAM usage in «CRF EXT-ICA Local Processing» VM	Alerts: 1 day Polling data: 1 min	Min on consecutive samples arrived in the last *X=1* minute	If (min > thr) then scale RAM	5GT-SO, via internal event notification

Network load MEC host → RRH	Alerts: 1 day Polling data: 1 min	Min on consecutive samples arrived in the last *X=1* minute	If (min > thr) then scale DL bw	5GT-SO, via internal event notification
Network load MEC ←→ MEC	Alerts: 1 day Polling data: 1 min	Min on consecutive samples arrived in the last *X=1* minute	If (min > thr) then scale bw between MEC hosts	5GT-SO, via internal event notification
Network load MEC →DC	N.A. (MEC to DC segment out of control)			
Storage usage in «CRF EXT-ICA Backend» VM	Alerts: 1 day Polling data: 1 min	Min on consecutive samples arrived in the last *X=1* minute	If (min > thr) then scale storage	5GT-SO, via internal event notification
# Vehicles in area	1 hr	Nothing	Nothing	5GT-VS – polling

10.2 Monitoring requirements for the entertainment use case

In this section we analyze the requirements for the entertainment use case, with reference to the Ucs E0.1 and E0.2 On-site live event experience.

The functional elements for E0.1 are shown in Figure 24, while the corresponding network service is depicted in Figure 25.

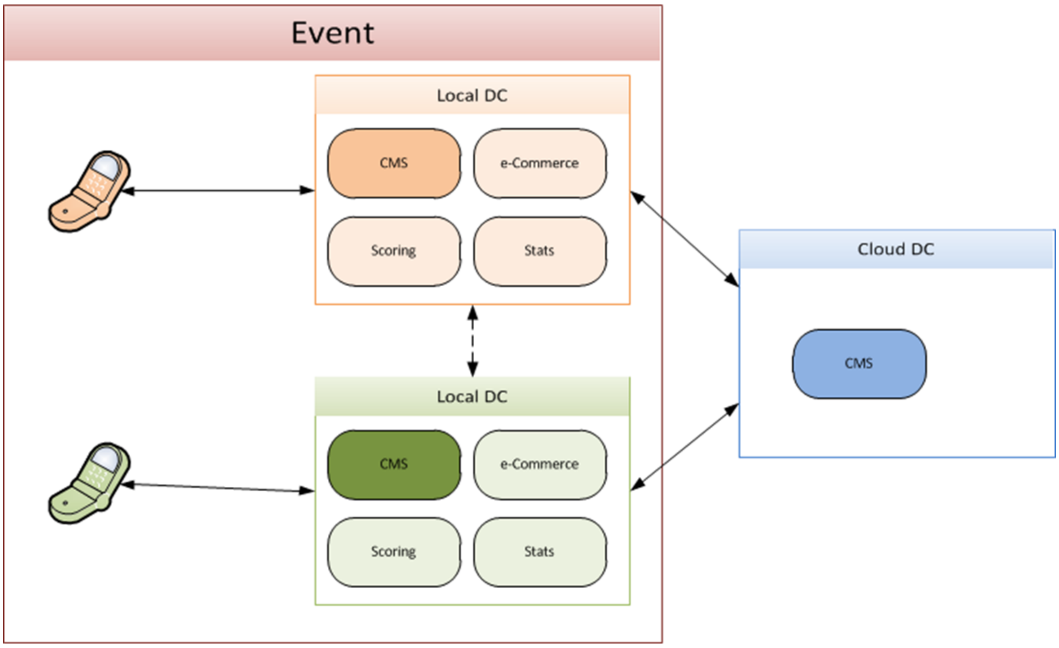


FIGURE 24: ON-SITE LIVE EVENT EXPERIENCE SERVICE (UC E0.1): FUNCTIONAL DECOMPOSITION

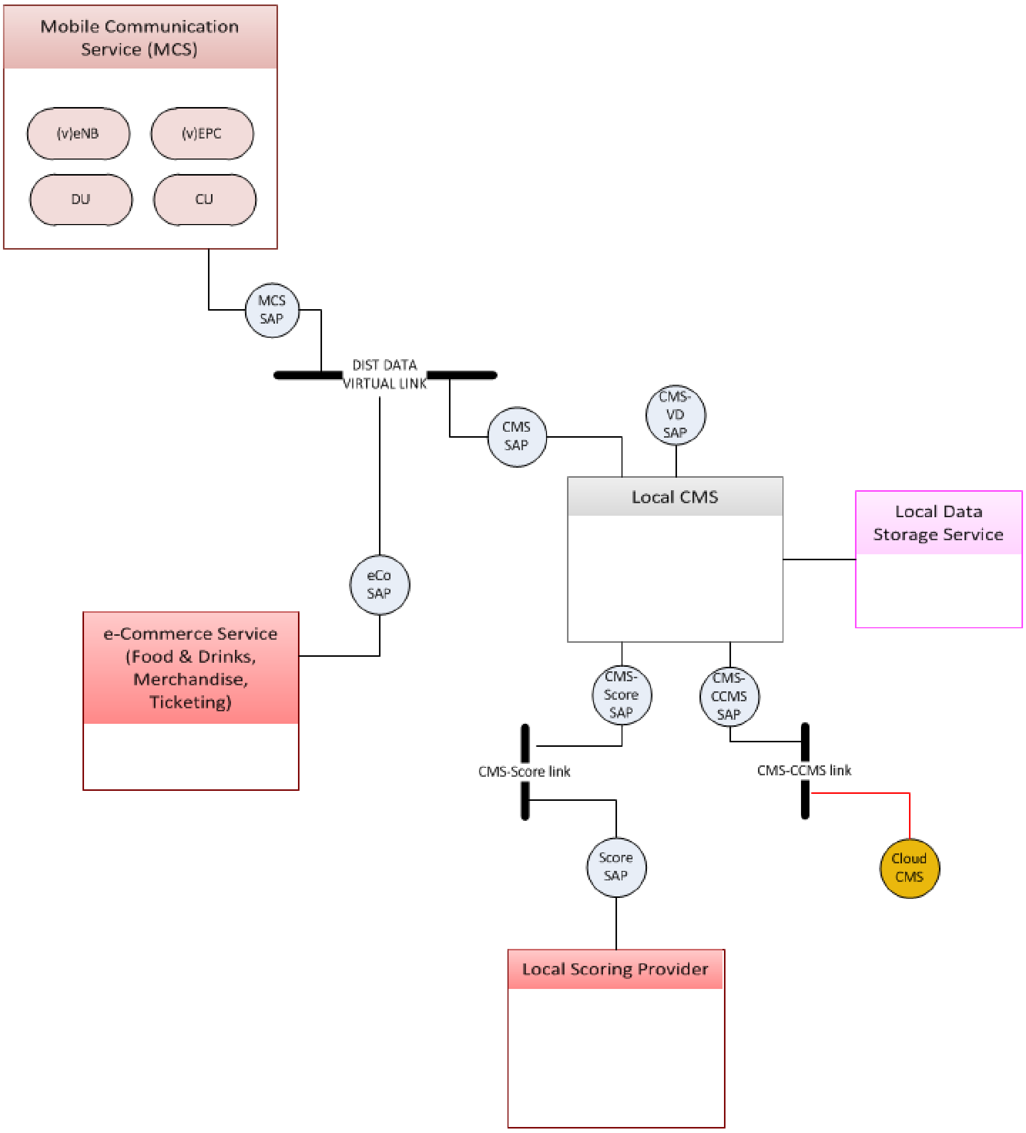


FIGURE 25: ON-SITE LIVE EVENT EXPERIENCE SERVICE (UC E0.1): NETWORK SERVICE MODELLING

The main external variable that affects the dynamicity of this service is the amount of people attending the event, which impacts the following requirements:

- Bandwidth capacity of the connectivity service from the Content Management System (CMS) to the data centre and between the CMS and the UEs.
- vCPU, RAM and storage utilization of the local CMS and the local storage service.

In particular, the following monitoring metrics should be considered for this use case:

- Monitoring data requested by the vertical service (produced by the 5GT-VS Monitoring Platform, out of scope for the 5GT-SO Monitoring Platform):
  - Validation of the SLAs, with reference to the following KPIs:
    - Ultra-low latency
    - Enhanced data rate
    - High reliability and availability (99.999%)

  - Monitoring of the correct working of the application
- Monitoring data for internal decisions at the 5GT-SO or 5GT-VS:
  - Network load in the connection UE - CMS, CMS-DC
  - Network latency in the connection UE - CMS, CMS-DC
  - vCPU, RAM, storage usage in CMS and local storage
  - Any failure event at the virtual infrastructure level.

The details of the monitoring metrics to be collected for this use case are summarized in Table 15.

TABLE 15: MONITORING METRICS FOR ON-SITE LIVE EVENT EXPERIENCE SERVICE (UC E0.1)

Metric	Format Amount Rate	Source	Collection method	Consumer
vCPU usage in CMS VMs	JSON msg with % value	V(N)F	Notifications on threshold	5GT-SO
RAM usage in CMS VMs	JSON msg with int value	V(N)F	Notifications on threshold	5GT-SO
Storage usage CMS VMs	JSON msg with % value	V(N)F	Notifications on threshold	5GT-SO
Entertainment app SLA	JSON msg from 5GT-VS	5GT-VS	Polling	Vertical
Network load UE<->local CMS	1 JSON msg (variable size) per UE every 10 sec	SDN controller	Polling on SDN controller stat service	5GT-SO
Network load LCMS<-> CCMS	1 JSON msg (variable size)	SDN controller	Polling on SDN controller stat service	5GT-SO
Network latency UE <->LCMS	1 JSON msg (variable) size per UE every 10 seconds	SDN controller	Polling on SDN controller stat service	5GT-SO
Network latency LCMS -> CCMS	1 JSON msg (variable) size per UE every 10 seconds	SDN controller	Polling on SDN controller stat service	5GT-SO
Network load LCMS<-> Scoring Provider	1 JSON msg (variable size)	SDN controller	Polling on SDN controller stat service	5GT-SO
Network latency UE <->Scoring provider	1 JSON msg (variable) size per UE	SDN controller	Polling on SDN controller stat service	5GT-SO

The functional elements for E0.2 are shown in Figure 26, while the corresponding network service is depicted in Figure 27.

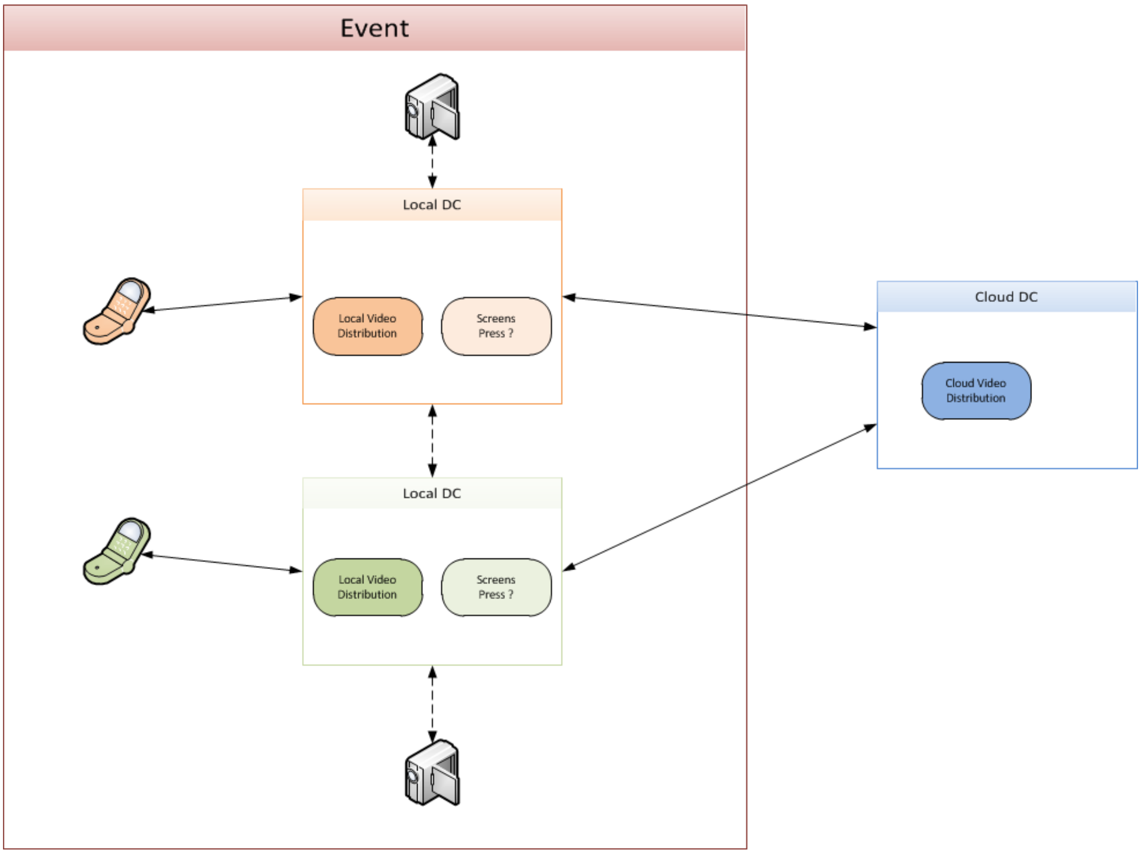


FIGURE 26: ON-SITE LIVE EVENT EXPERIENCE SERVICE (UC E0.2): FUNCTIONAL DECOMPOSITION

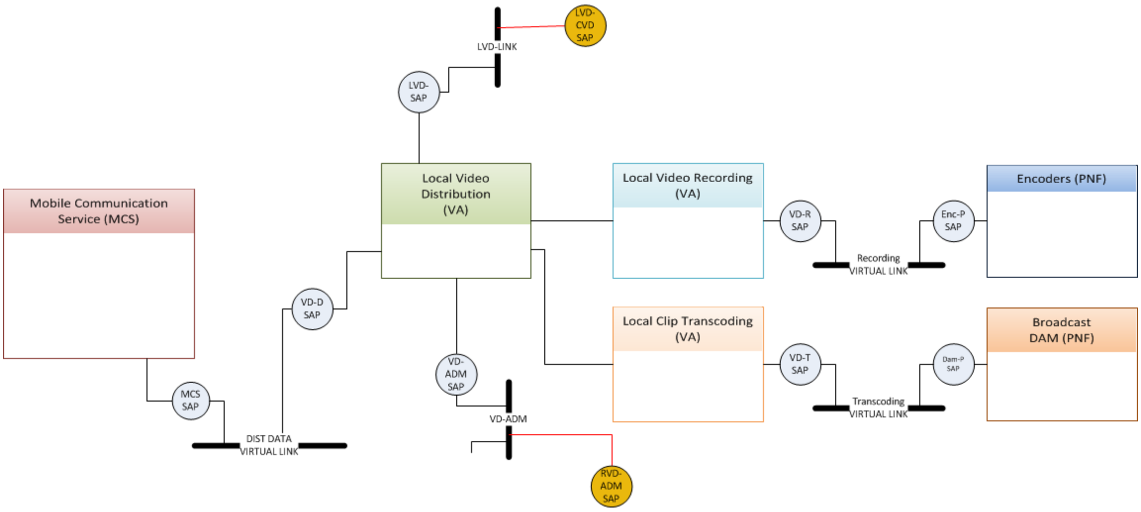


FIGURE 27: ON-SITE LIVE EVENT EXPERIENCE SERVICE (UC E0.2): NETWORK SERVICE MODELLING

In addition to the amount of people attending the event, in this case the dynamicity of the service is also affected by the amount of video feeds, with impact on the bandwidth requirements and on the vCPU, RAM and storage of the local CMS and the local storage service.

The considerations done for the previous use case about SLA monitoring are still valid. Moreover, the following monitoring metrics should be considered for internal decisions at the 5GT-SO or 5GT-VS:

- Network load in the connection UE - LVD (Local Video Distribution), LVD-CVD (Cloud Video Distribution), LVD…
- Network latency in the connection UE - LVD, LVD-CVD, LVD…
- Packet losses UE - LVD.
- vCPU, RAM, storage usage in LVD, LVR (Local Video Recording) and LCT (Local Clip Transcoding).
- Any failure event at the virtual infrastructure level.

The details of the monitoring metrics to be collected for this use case are summarized in Table 16.

TABLE 16: MONITORING METRICS FOR ON-SITE LIVE EVENT EXPERIENCE SERVICE (UC E0.2)

Metric	Format Amount Rate	Source	Collection method	Consumer
vCPU usage in CMS VMs	JSON msg with % value every 10 seconds	V(N)F	Notifications on threshold	5GT-SO
RAM usage in CMS VMs	JSON msg with int value every 10 seconds	V(N)F	Notifications on threshold	5GT-SO
Storage usage CMS VMs	JSON msg with % value	V(N)F	Notifications on threshold	5GT-SO
Entertainment app SLA	JSON msg from 5GT-VS	5GT-VS	Polling	Vertical
Network load UE<->local CMS	1 JSON msg (variable size) per UE every 10 sec	SDN controller	Polling on SDN controller stat service	5GT-SO
Network load LCMS<-> CCMS	1 JSON msg (variable size) every 10 seconds	SDN controller	Polling on SDN controller stat service	5GT-SO
Network latency UE <->LCMS	1 JSON msg (variable size) per UE every 10 seconds	SDN controller	Polling on SDN controller stat service	5GT-SO
Network latency LCMS -> CCMS	1 JSON msg (variable size) per UE every 10 seconds	SDN controller	Polling on SDN controller stat service	5GT-SO
Network load LCMS<-> Scoring Provider	1 JSON msg (variable size)	SDN controller	Polling on SDN controller stat service	5GT-SO

Network latency UE <->Scoring provider	1 JSON msg (variable) size per UE	SDN controller	Polling on SDN controller stat service	5GT-SO
Packet losses UE<-> LVD	1 JSON msg (variable size) per UE	SDN controller	Polling on SDN controller stat service	5GT-SO, Vertical

10.3 Monitoring requirements for the e-Health use case

In this section we analyze the requirements for the e-Health use case, represented in Figure 28.

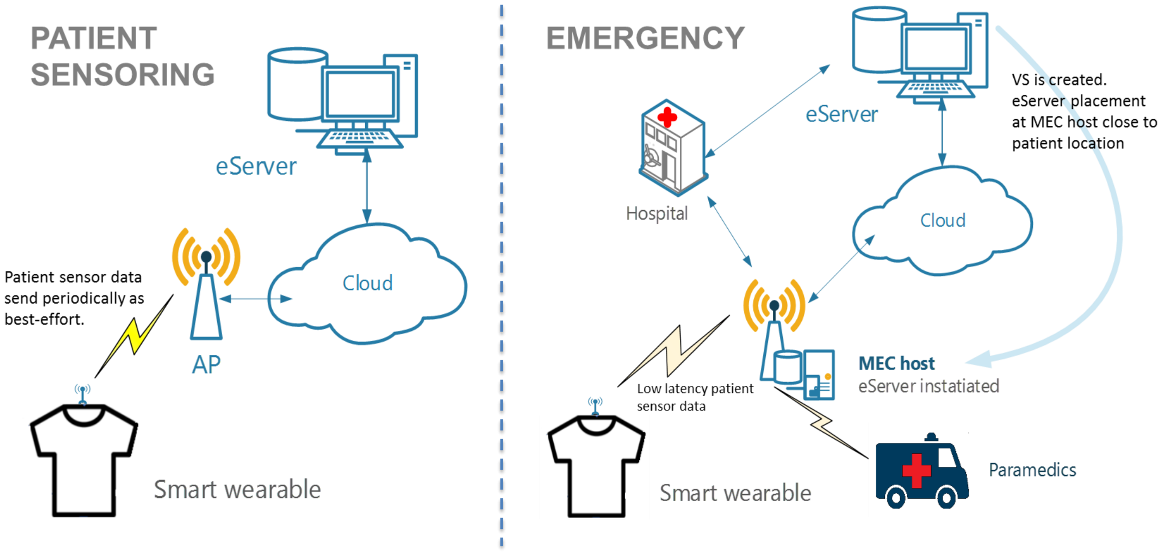

PATIENT SENSORING
eServer
Cloud
AP
Patient sensor data send periodically as best-effort.
Smart wearable
EMERGENCY
eServer
Hospital
Cloud
VS is created. eServer placement at MEC host close to patient location
MEC host
eServer instatiated
Low latency patient sensor data
Smart wearable
Paramedics

FIGURE 28: E-HEALTH SERVICE: FUNCTIONAL DECOMPOSITION

In a patient sensoring slice, the number of users does not impact the usage of the resources, while an increasing number of emergency patients under the coverage of a MEC area has the following effects:

- The increased number of monitoring data from the patient has an Impact on the bandwidth requirements from RRHs to MEC server.
- The increased number of messages towards paramedic units has an impact on bandwidth requirements from MEC server to RRHs and hospitals
- The increased number of messages from paramedic units has an impact on bandwidth requirements (video streaming, proxy monitoring data) from RRHs to MEC server
- The increased capacity and processing of emergency patient data has an impact on vCPU and capacity for processing and sending patient data towards hospitals and/or paramedics

In particular, the following monitoring metrics should be considered for this use case:

- Monitoring data requested by the vertical service:
    - Number of emergency patients in each service covered MEC area, per RRH

  - Number and location of paramedic units in each MEC area
- Monitoring data for internal decisions at the 5GT-SO or 5GT-VS:
  - Network latency in the connection between patient and MEC host.
  - Network latency and bandwidth in the connection between paramedic and MEC host.
  - Network load between MEC host and hospital/eServer.
  - vCPU, RAM, storage usage in MEC host.
  - Any failure event at the virtual infrastructure level.

The details of the monitoring metrics to be collected for this use case are summarized in Table 17.

TABLE 17: MONITORING METRICS FOR E-HEALTH

Metric	Format Amount Rate	Source	Collection method	Consumer
Network latency UE→MEC host	JSON msg with int value	MEC host	Notifications on threshold + polling after alert	5GT-SO
Network latency Paramedic → MEC host	JSON msg with int value	MEC host	Notifications on threshold + polling after alert	5GT-SO
Network bandwidth Paramedic → MEC host	JSON msg with int value	MEC host	Notifications on threshold + polling after alert	5GT-SO
Network load MEC host ←→ hospital/eServer	JSON msg with int value	MEC host	Notifications on threshold + polling after alert	5GT-SO
vCPU usage in MEC host	JSON msg with % value	MEC host	Notifications on threshold + polling after alert	5GT-SO
Storage usage in MEC host	JSON msg with % value	MEC host	Notifications on threshold + polling after alert	5GT-SO
RAM usage in MEC host	JSON msg with % value	MEC host	Notifications on threshold + polling after alert	5GT-SO
# of emergency patients	JSON msg with int value	MEC host	Polling	5GT-VS
# of paramedic units	JSON msg with int value	MEC host	Polling	5GT-VS

10.4 Monitoring requirements for the e-Industry use case

In this section we analyze the requirements for the e-Industry use case, with reference to the Cloud Robotics for Industrial Automation service, represented in Figure 29.

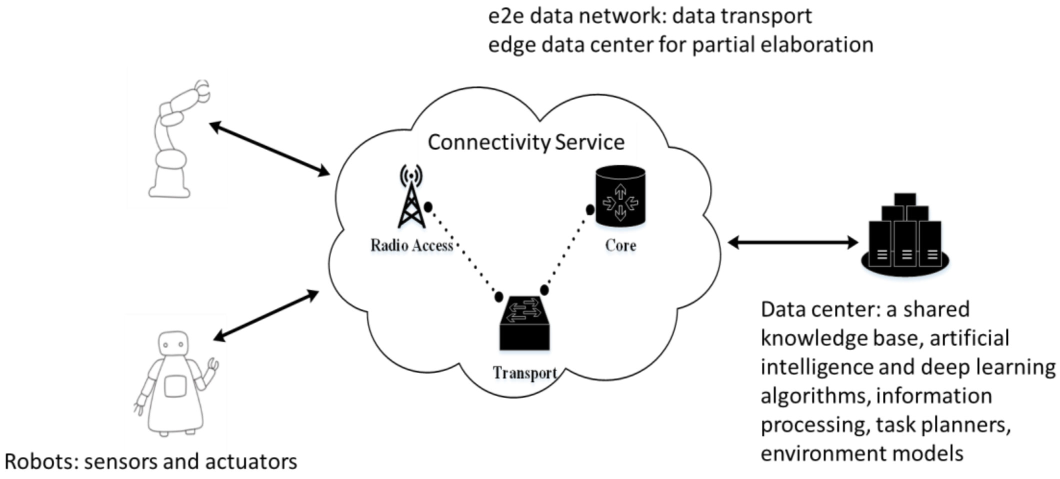

e2e data network: data transport
edge data center for partial elaboration
Connectivity Service
Radio Access
Core
Transport
Data center: a shared knowledge base, artificial intelligence and deep learning algorithms, information processing, task planners, environment models
Robots: sensors and actuators

FIGURE 29: E-INDUSTRY – CLOUD ROBOTICS SERVICE: FUNCTIONAL DECOMPOSITION

In this use case, we can identify the following variables:

- The size of the factory and the number of robots, which affects the number of messages produced by the robots and, consequently, the capacity requirements of the network connectivity from the robot (UE) to the data centre.
- The robot task, which affects the amount of exchanged data, e.g. when images are exchanged between robot and data centre the amount of data increases.
- The robot control task, which requires higher rate for the control messages and lower latency.

In particular, the following monitoring metrics are required for internal decisions at the 5GT-SO or 5GT-VS:

- Network load in the connection from robot (i.e., UE) and data centre (DC).
- Network latency in the connection from robot (i.e., UE) and DC.
- vCPU and RAM usage in VMs for robot control.
- Any failure event at the virtual infrastructure level.

The details of the monitoring metrics to be collected for this use case are summarized in Table 18.

TABLE 18: MONITORING METRICS FOR E-INDUSTRY

Metric	Format Amount Rate	Source	Collection method	Consumer
Network load UE→DC	1 JSON msg (variable size) per robot node every 10 sec	SDN controller	Polling on SDN controller stat service	5GT-SO
Network latency UE → DC	1 JSON msg (variable) size per robot every	SDN controller	Polling on SDN controller stat service	5GT-SO

	x second based on the control cycle			
vCPU usage in robot control algo VMs	JSON msg with % value	V(N)F	Notifications on threshold	5GT-SO
RAM usage in robot contro algo VMs	JSON msg with int value	V(N)F	Notifications on threshold	5GT-SO
Network load DC → UE	1 JSON msg (variable size) per network node every 10 sec	SDN controller	Polling on SDN controller stat service	5GT-SO
Storage usage ReportDB	JSON msg with % value	V(N)F	Notifications on threshold	5GT-SO
Cloud Robotics app SLA	JSON msg from 5GT-VS	5GT-VS	Polling	Vertical

11 Annex III. Notation for Requirements

In this deliverable we follow - with slight adaptions - the notation for requirements used already in [9]. For each requirement, the following fields should be provided:

ID	Requirement	F/NF
ReqX.XX	e.g. The vehicle shall be connected to a 5G router	F/NF

The meanings of the fields are as follows:

- **ID**: is the identifier of the requirement (written in the form ReqX.XX).
- **Requirement**: a complete sentence explaining the requirement.
- **F/NF**: if the requirement is Functional (F) or Non Functional (NF).

NOTE: The requirement field is written following the approach followed by IETF documents, included next. The key words "must", "must not", "required", "shall", "shall not", "should", "should not", "recommended", "may" and "optional" in this document are to be interpreted as described in [61].

1. **MUST** This word, or the terms "**REQUIRED**" or "**SHALL**", mean that the definition is an absolute requirement of the specification.
2. **MUST NOT** This phrase, or the phrase "**SHALL NOT**", means that the definition is an absolute prohibition of the specification.
3. **SHOULD** This word, or the adjective "**RECOMMENDED**", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighted before choosing a different course.
4. **SHOULD NOT** This phrase, or the phrase "**NOT RECOMMENDED**" means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
5. **MAY** This word, or the adjective "**OPTIONAL**", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item.

An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein, an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.).

12 Annex IV. Reference Architectures

A few Standard Development Organizations (SDOs) and fora are contributing to the design of management systems for 5G that have many common design principles: (i) flexibility, (ii) adaptability, and (iii) cost-efficiency. Despite the many common design objectives across these working groups (as presented below), there are various relevant architectural concepts (e.g., slicing, federation/multi-domain, edge computing) that are specific to individual groups. In this context, 5G-TRANSFORMER strives to bridge the gaps across such heterogeneous ecosystem in order to harmonically integrate these concepts under a single architecture.

This section introduces ongoing architectural work at 3GPP, ETSI NFV and MEC that fulfills two objectives:

- Serve as inspiration to define the 5G-TRANSFORMER architecture;
- Set the framework in which 5G- TRANSFORMER must be integrated to maximize its impact, i.e., by seeking as much as possible compliance with what is already defined.

Despite the fact that there are other organizations discussing about slicing and architectural concepts related with 5G-TRANSFORMER, we focus on the ones below because they are the ones with a more complete definition of their architecture and building blocks, and so, they go well beyond requirements and high-level concepts.

12.1 3GPP

The most relevant working groups inside 3GPP related to 5G-TRANSFORMER are SA2 (Architecture) and SA5 (Telecom management).

12.1.1 3GPP SA2

The 3GPP SA2 Working Group (WG), responsible for overall system architecture, is currently working on specifying the 5G Core (5GC) architecture with Network Slicing being a main feature of 5GC. Technical Specification (TS) 23.501 [37] defines Stage-2 System Architecture for the 5G System which includes Network Slicing. Figure 30 depicts an example of network slicing from 3GPP's perspective.

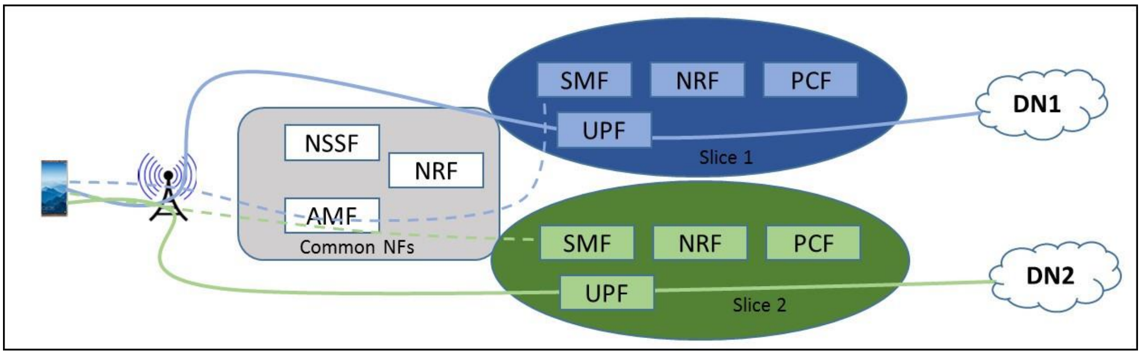


FIGURE 30: EXAMPLE OF NETWORK SLICES FROM 3GPP SA2 PERSPECTIVE

A network slice is viewed as a logical end-to-end network that can be dynamically created. A given User Equipment (UE) may access to multiple slices over the same Access Network (e.g. over the same radio interface). Each slice may serve a particular service type with agreed upon SLA. In the following, we provide highlights of 3GPP Network Slicing as being defined in TS 23.501 [37] in SA2.

A Network Slice is defined within a Public Land Mobile Network (PLMN) and includes the Core Network Control Plane and User Plane Network Functions as well as the 5G Access Network (AN). The 5G Access Network may be a Next Generation (NG) Radio Access Network described in 3GPP TS 38.300 [38] or a non-3GPP Access Network.

TS 23.501 [37] defines Network Function, Slice, and Slice Instance as follows:

- Network Function: A 3GPP adopted or 3GPP defined processing function in a network, which has defined functional behavior and 3GPP defined interfaces. (Note: A network function can be implemented either as a network element on a dedicated hardware, as a software instance running on a dedicated hardware, or as a virtualized function instantiated on an appropriate platform, e.g. on a cloud infrastructure.);
- Network Slice: A logical network that provides specific network capabilities and network characteristics;
- Network Slice instance: A set of Network Function instances and the required resources (e.g. compute, storage and networking resources) which form a deployed Network Slice.

12.1.2 3GPP SA5

3GPP SA5 Working Group (WG) is the 3GPP telecom management working group. 3GPP SA5 specifies the requirements, architecture and solutions for provisioning and management of the network, including Radio Access Network (RAN) and Core Network (CN) and its services.

SA5 has completed a study on management and orchestration on network slicing (3GPP 28.801 [36]) and started the normative specification work for release 15 based on this study. It is expected to be completed by the second quarter of 2018, including:

- Network slice concepts, use cases and requirements (3GPP 28.530 [39]);
- Provisioning of network slicing for 5G networks and services (3GPP 28.531[40]);
- Assurance data and Performance Management for 5G networks and network slicing;
- Fault Supervision for 5G network and network slicing.

The following description highlights management and orchestration aspects of network slicing in 3GPP 28.801 [36]. However, these may be updated in the SA5 normative specifications based on the ongoing development of the SA2 technical specifications.

- General management and orchestration aspects of network slicing defined in 3GPP 28.801 [36]. Based on 3GPP 23.501 [37], SA5 has defined different management aspects for network slices in 3GPP 28.801 [36] as listed below:
  - Managing a complete Network Slice Instance (NSI) is not only managing all the functionalities but also the resources necessary to support certain set of communication services.
  - An NSI not only contains Network Functions (NFs), e.g. belonging to AN and CN, but also the connectivity between the NFs. If the NFs are interconnected, the 3GPP management system contains the information

relevant to connections between these NFs such as topology of connections, individual link requirements (e.g. QoS attributes), etc. For the part of the Transport Network (TN) supporting connectivity between the NFs, the 3GPP management system provides link requirements to the management system that handles the part of the TN supporting connectivity between the NFs.
   - NSI can be composed of network slice subnets of Physical Network Functions and/or Virtualized Network Functions.

- Network Slice Instance lifecycle management. 3GPP 28.801 [36] has introduced the network slice instance lifecycle management as depicted below in Figure 31, considering it independent of the network service instance which is using the network slice instance. Typically, a network slice instance is designed (preparation phase), then it is instantiated (Instantiation, Configuration and Activation phase), then it is operated (Run Time phase) and finally it may be decommissioned when the slice is no longer needed (Decommissioning phase). 3GPP 28.801 [36] introduces 3 management logical functions:
   - Communication Service Management Function (CSMF): Responsible for translating the communication service related requirement to network slice related requirements.
   - Network Slice Management Function (NSMF): Responsible for management and orchestration of NSI and derive network slice subnet related requirements from network slice related requirements.
   - Network Slice Subnet Management Function (NSSMF): Responsible for management and orchestration of network slice subnet instances (NSSI).

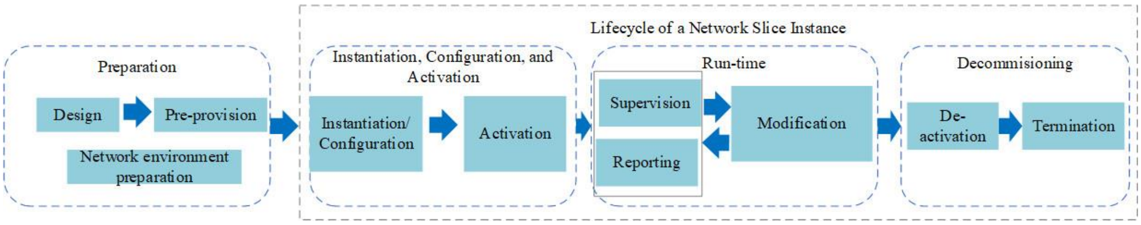

Lifecycle of a Network Slice Instance
Preparation
Design
Pre-provision
Network environment preparation
Instantiation, Configuration, and Activation
Instantiation/ Configuration
Activation
Run-time
Supervision
Reporting
Modification
Decommisioning
De-activation
Termination

FIGURE 31: 3GPP VIEW ON NETWORK SLICE INSTANCE LIFECYCLE

12.2 ETSI

12.2.1 ETSI NFV

Work is also ongoing inside ETSI NFV on how the NFV architecture in general, but more specifically, the ETSI MANO components can support network slicing.

In this respect the Evolution and Ecosystem (EVE) working group has carried out activities that map NFV and 3GPP network slicing concepts. On the one hand, ETSI NFV-EVE 012 [29] establishes the correspondence between a network slice (as defined by 3GPP) and a network service (as defined by ETSI NFV). There, ETSI describes that an NFV Network Service (NFV-NS) can be regarded as a resource-centric view of a network slice, for the cases where an NSI would contain at least one virtualized network function. According to 3GPP 28.801 [36], an NSSI can be shared by multiple NSIs. The virtualized resources for the slice subnet and their connectivity to physical resources can thus be represented by the nested Network Service concept defined in ETSI GS NFV-IFA 014 [24] (right hand side of Figure 32), or one or more VNFs and PNFs directly attached to the Network Service used by the network slice. Figure 32 illustrates the relationship between 3GPP's network slice and ETSI NFV Network Service.

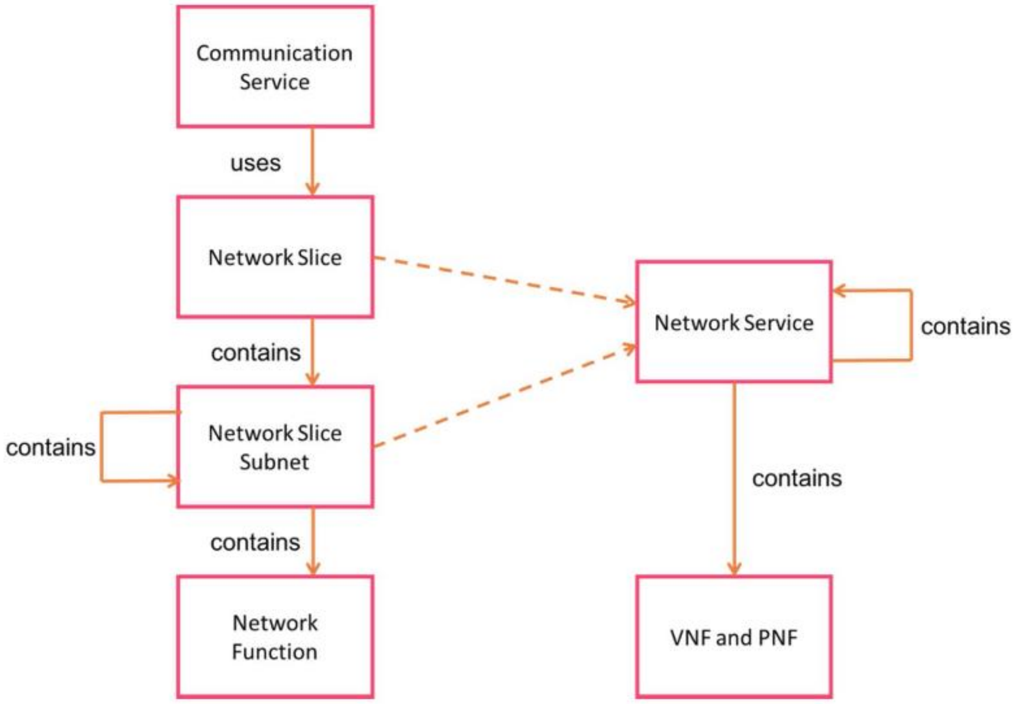


FIGURE 32: RELATION BETWEEN 3GPP AND ETSI INFORMATION MODELS (FROM [29])

As mentioned before, 3GPP 28.801 [36] identifies three management functions related to network slicing management: Communication Service Management Function (CSMF), Network Slice Management Function (NSMF), and Network Slice Subnet Management Function (NSSMF).

As shown in Figure 33, the Os-Ma-nfvo reference point can be used for the interaction between 3GPP slicing related management functions and NFV MANO. To properly interface with NFV MANO, the NSMF and/or NSSMF need to determine the type of Network Service or set of Network Services, VNF and PNF that can support the resource requirements for a NSI or NSSI. In addition, they need to determine whether new instances of these Network services, VNFs and the connectivity to the PNFs need to be created or existing instances can be reused.

From a resource management viewpoint, there are different approaches to map NSIs to Network Services. In the first approach, NSI can be mapped to an instance of a simple or composite network service or to a concatenation of such Network Service instances. From a resource management viewpoint, different NSIs can use instances of the same type of Network Service (i.e. they are instantiated from the same Network Service Descriptor or NSD) with the same or different deployment flavors. In the second approach, different NSIs can use instances of different types of Network Services. The first approach can be used if the NSIs share the same types of network functions (or a large common subset) but differ in terms of the performance expected from these network functions (and from the virtual links connecting them) and/or the number of instances to be deployed for each of them. If slices differ more significantly, mapping to different Network Services, each with its own NSD can be considered. The same mapping principles might apply to NSSIs.

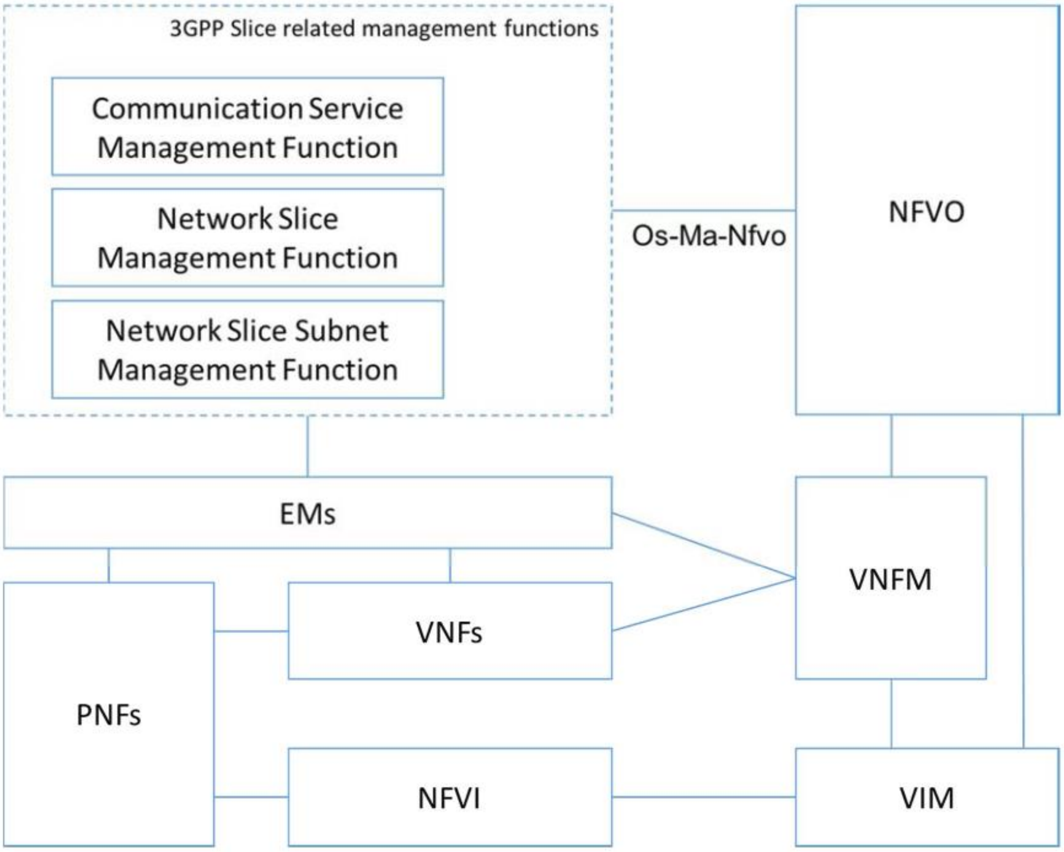

3GPP Slice related management functions
Communication Service Management Function
Network Slice Management Function
Network Slice Subnet Management Function
Os-Ma-Nfvo
NFVO
EMs
VNFM
VNFs
PNFs
NFVI
VIM

FIGURE 33: NETWORK SLICE MANAGEMENT IN AN NFV FRAMEWORK (FROM [29])

Also, as described before, 3GPP 28.801 [36] describes the lifecycle of a network slice, which is comprised of the four following phases: (i) Preparation; (ii) Instantiation, configuration and activation; (iii) Run-time; and (iv) Decommissioning.

The preparation phase includes the creation and verification of Network Slice Template(s) (NST(s)). From an NFV perspective, the resource requirement for a NST can be realized by one or more existing NSDs that have been previously on-boarded on the NFVO. The creation of a new NST can lead to requiring update of an existing NSD or generation of a new NSD followed by on-boarding the new NSD if the slice requirements do not map to an already on-boarded NSD. Indeed, the NFV-NS for the multiple NSIs may be instantiated with the same NSD, in order to deliver exactly the same optimizations and features but dedicated to different enterprise customers. On the other hand, a network slice intended to support totally new customer facing services is likely to require a new NFV-NS and thus the generation of a new NSD. The network slice instantiation step in the second phase triggers the instantiation of the underlying NSs. NFV-MANO functions are only involved in the network slice configuration phase if the configuration of virtualisation-related parameters is required on one or more of the constituent VNF instances. Configuration of the network applications embedded in the constituent network functions involves the NSMF or NSSMF and/or other parts of the OSS/BSS, and the element managers (if any) associated to these functions. NFV-

MANO functions can be triggered during the network slice activation step. If explicit activation of VNFs is required, the NSMF or the NSSMF can change the operational state of those VNFs through an Update NFV-NS operation defined in ETSI GS NFV-IFA 013 [23]. The involvement of NFV-MANO in the run-time phase is limited to the operations related to the performance management, fault management, and lifecycle management of virtualised resources (e.g. scaling an underlying NFV-NS to expand a NSI). The decommissioning phase triggers the termination of the underlying network service instances.

Additionally, and given the multiple administrative boundaries of the 5G-TRANSFORMER architecture, the Interfaces and Architecture (IFA) working group is of particular interest for our project. ETSI GS NFV-IFA 028 [26] reports on potential architecture options to support the offering of NFV MANO services across multiple administrative domain. NFV-MANO services can be offered and consumed by different organizations, e.g. by different network operators or by different departments within the same network operator. Administrative domains as defined in ETSI GS NFV-IFA 010 [21] can be mapped to such different organizations. Examples of use cases for NFV-MANO service offerings across multiple administrative domains are described in ETSI NFV 001 [16]. Furthermore, ETSI GS NFV-IFA 022 [25] reports on the functional architecture needed to provision and manage multi-site network services. To this end, a set of multi-site use cases are studied.

Furthermore, compliance with widely accepted standards of the 5G-TRANSFORMER architecture is also relevant to maximize its impact. Therefore, in a more general architectural context than that defined by the previous documents (which focus on specific issues) the interfaces already defined in ETSI NFV MANO are also relevant:

- ETSI GS NFV-IFA 013 [23] defines the interfaces supported over the Os-Ma-nfvo reference point of the NFV MANO architectural framework as well as the information elements exchanged over those interfaces;
- ETSI GS NFV-IFA 005 [17] defines the interfaces supported over the Or-Vi reference point of the NFV MANO architectural framework as well as the information elements exchanged over those interfaces;

ETSI GS NFV-IFA 006 [18] defines the interfaces supported over the Vi-Vnfm reference point of the NFV MANO architectural framework as well as the information elements exchanged over those interfaces;

12.2.2 ETSI MEC

Multi-Access Edge Computing (MEC) is one of the key concepts for fulfilling some of the requirements of vertical services, and therefore its integration in the 5G-TRANSFORMER architecture is nexus in its design. MEC and its integration in an NFV context was studied in ETSI MEC017 [28] document and a reference architecture is provided with the following key observations:

- The mobile edge platform is deployed as a VNF and therefore the procedures defined by ETSI NFV for this means are used;
- ETSI NFV MANO sees mobile edge applications as regular VNFs allowing for reuse of ETSI MANO functionality (with perhaps some extensions);
- The virtualization infrastructure is deployed as a NFVI and its virtualized resources are managed by the VIM. For this purpose, the procedures defined

by ETSI NFV Infrastructure specifications, i.e. ETSI NFV INF003 [30], ETSI NFV INF004 [31] and ETSI NFV INF005 [32] can be used.

13 Annex V. State of the Art Solutions for 5GT-SO

Orchestration for networks, services and applications emerged as a technology domain several years ago. Originally, it was focused on an application deployment and lifecycle management where it is tightly coupled with Continuous Integration/Continuous Delivery (CI/CD) principles. By recognizing benefits of the automated and non-interrupted service delivery on an application layer, orchestration moved down the stack to the infrastructure and network layers. Multiple software implementations of the orchestration platforms for the NFV domain are already available both as open source and proprietary solutions.

In the scope of 5G, network softwarization, service orchestration and automated lifecycle management are mandatory components of the 5G infrastructure. Thus, the 5G-PPP Software Networks working group analyses and attempts to unify the research results of the various 5G-PPP research projects. In [45], the contributions of the research projects in the area of SDN and NFV are collected and an overall architecture for SDN/NFV integration and management is proposed. Several network features and scenarios are presented in this white paper and their impact on SDN and NFV are investigated. Multi-domain and multi-provider orchestration as well as network slicing are those network features that are most relevant for the work described in this deliverable. The Software Networks working group has been classifying assets of phase 1 projects for reuse in phase 2 projects. Several projects such as SONATA, SelfNet, and 5GEx have created assets related to orchestration.

Besides 5G-PPP R&D projects, there are number of open source industry-driven communities focused on the development of NFV orchestration platforms for Telcos. In the following section we review these projects in detail, focusing on the main features and functions of these platforms, including internal software architecture, building blocks and service deployment approach. Essential part of this review is an analysis of the applicability of each platform for the 5G-TRANSFORMER architecture with the purpose of selecting candidate schemes for the Service Orchestrator role.

13.1 Industry-driven projects

In this section we review several industry-driven NFV orchestration projects. All the platforms below are available in the form of open source code which can be downloaded and deployed at local environment. Additionally, it should be stated that some of the projects below are available as commercially-supported software. However, this aspect is out of scope for the current analysis.

13.1.1 Open Source MANO (OSM)

During 2016 ETSI launched the Open Source Mano (OSM) initiative [46]. OSM intends to develop an Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV specifications. This kind of Open Source software initiative can facilitate the implementation of NFV architectures aligned to ETSI NFV specifications, increasing and ensuring the interoperability among NFV implementations.

The baseline implementation took as references RIFT.ware [47], OpenMANO [46] and Juju [48] which were provided by RIFT.io, Telefonica and Canonical respectively. The first release of OSM was announced in May 2016. The third release of OSM [46] was released in October 2017 with incremental additions with respect to the two other releases.

The architectural principles followed during the development of Open Source MANO can be summarized as follows:

- Layering: OSM provides a clear delineation between the layers and modules while broadly aligning with ETSI NFV ISG architectural framework.
- Abstraction: clear differentiation in the levels of abstraction/detail presented between layers to facilitate independent evolution of the components, facilitating integration.
- Modularity: clear modularity enabled with a plugin model preferred to facilitate module replacements as OSM community develops.
- Simplicity: solution focused on orchestration just incorporating the minimal complexity necessary to be successful, with clear yet simple map of components and modules, as well as lightweight implementation.

Figure 34 shows the main components for OSM.

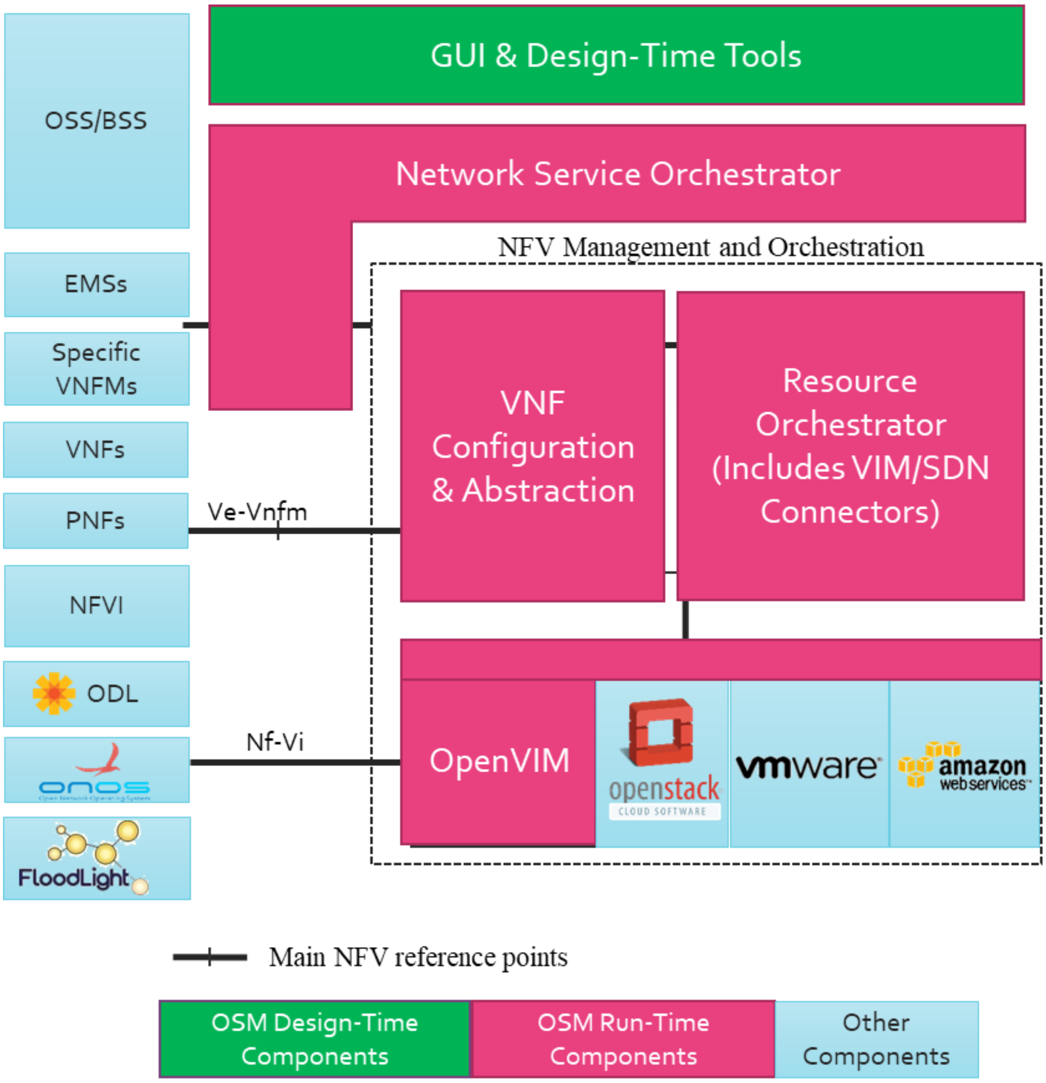

OSS/BSS
GUI & Design-Time Tools
Network Service Orchestrator
NFV Management and Orchestration
EMSs
Specific VNFMs
VNFs
PNFs
NFVI
ODL
onos
FloodLight
VNF Configuration & Abstraction
Resource Orchestrator (Includes VIM/SDN Connectors)
Ve-Vnfm
Nf-Vi
OpenVIM
openstack CLOUD SOFTWARE
vmware
amazon webservices
Main NFV reference points
OSM Design-Time Components
OSM Run-Time Components
Other Components

FIGURE 34: OSM MAPPING TO ETSI NFV MANO

The main blocks of OSM cover the following functionalities:

- GUI and Design-Time Tools (UI): This is a graphical interface which allows users to manage the Network Service (NS) and VNF catalogues and lifecycle of NSs. Also, through this interface users can perform NFV-NS and VNF level configurations and compose new NSs. This component is realized with Launchpad which is part of RIFT.ware.
- Network Service Orchestrator (SO): This is the component that provides the end-to-end service orchestration by managing resources through the RO and using configuration plugins such as Juju to manage VNFs. RIFT.ware implements this component.
- Resource Orchestrator (RO): The RO is realized with OpenMANO being responsible for the creation and management of compute and network resources necessary for the NFV-NS instantiation.
- VNF Configuration & Abstraction (CM or VCA): This component uses Juju to realize the interface with the applications deployed on top of the virtual resources. That is, Juju provides VNF configuration management capabilities to the SO.
- OpenVIM: It is the reference Virtual Infrastructure Manager (VIM) implementation in OSM for all-in-one installations.

The main objective of the first release of OSM was only to provide a first integration of the various components and data models to provide a single-entry point to end-users.

The second release included relevant steps forward, among others:

- Agnostic Data Model, i.e., the creation of a unified data model that better reflects ETSI’s specification while abstracting specific technology used for implementing OSM.
- Plugin Framework, that allows to specify and implement more generic interfaces between components which will lead to a plugin framework that eases the integration with third party components (e.g. SDN controllers, VIMs).
- Multi-VIM support in order to allow the instantiation of network services across multiple VIMs.

Finally, the recently delivered release THREE presents the following capabilities:

- Security: OSM includes comprehensive Role-Based Access Control and Multi-Tenancy/Project to the interface model.
- Service assurance: support for network scaling events to add and remove full VNF instances from a running Network Service.
- Resiliency: improvements on the recovery on single component failure, supporting multiple VCA instances and by offering improved scalability of the OSM platform.
- Usability: continuous work for facilitating usability of OSM code: Python based OSM client, VNF consoles accessible via the GUI, etc.
- Interoperability:
  - At VIM level: extension of the Amazon Web Services EC2 plugin, improvements to the VMware vCloud Director VIM plugin compatible with vCloud NFV 1.5 and 2.0, and support for VMware Integrated OpenStack (VIO).

- At SDN controller level: support of ONOS, OpenDaylight (ODL) and Floodlight.

At monitoring level: new monitoring plugin framework enabled Amazon CloudWatch, VMware vRealize Operations Manager, OpenStack Aodh and OpenStack Gnocchi monitoring tools.

ETSI OSM includes by design some of the functionalities expected in 5G-TRANSFORMER while some other are not yet present.

Basically, OSM implements now both service and resource orchestration in a differentiated manner. This can facilitate the usage of OSM components in the project. Additionally, some experiences have been already developed for service chain and 5G network slices.

However, the following aspects are not yet covered:

- No component similar to the Vertical Slicer for generating NSD from slice templates.
- No multi-domain approach yet covered.

With respect to this last issue, however, OSM considers already a multi-PoP infrastructure emulation platform, as presented in Figure 35 that could be leveraged as starting point for multi-domain extensions.

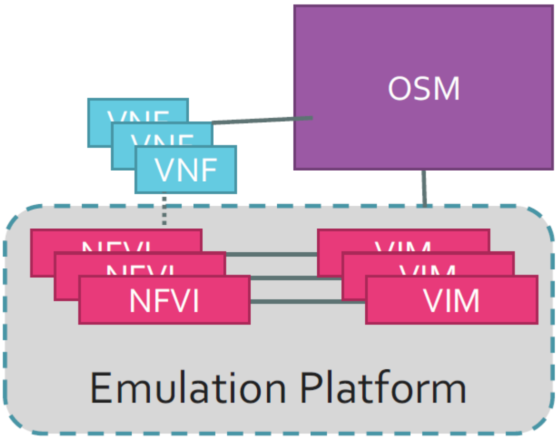


FIGURE 35: OSM MULTI-POP NFVI+VIM EMULATION PLATFORM

13.1.2 Open Network Automation Platform (ONAP)

ONAP (Open Network Automation Platform) was founded in 2017 by merging the AT&T driven OpenECOMP project and another NFV orchestration project called Open-O. According to [49], the project appeared as a response to a rising need for a common platform for telecommunication, cable, and cloud operators and their solution providers to deliver differentiated network services on demand, profitably and competitively, while leveraging existing investments. The first official platform release, called Amsterdam, was delivered in November 2017. It consists of the integrated software artifacts originated both from the OpenECOMP and Open-O projects.

From an architecture standpoint, ONAP consists of multiple software components. These components are part of two major architectural frameworks:

- Design-time framework - used to design, define and program the platform and services
- Execution-time framework - used to execute service logic and deploy services, defined during the design phase.

In Figure 36 the general architecture of the ONAP platform, Amsterdam release is presented.

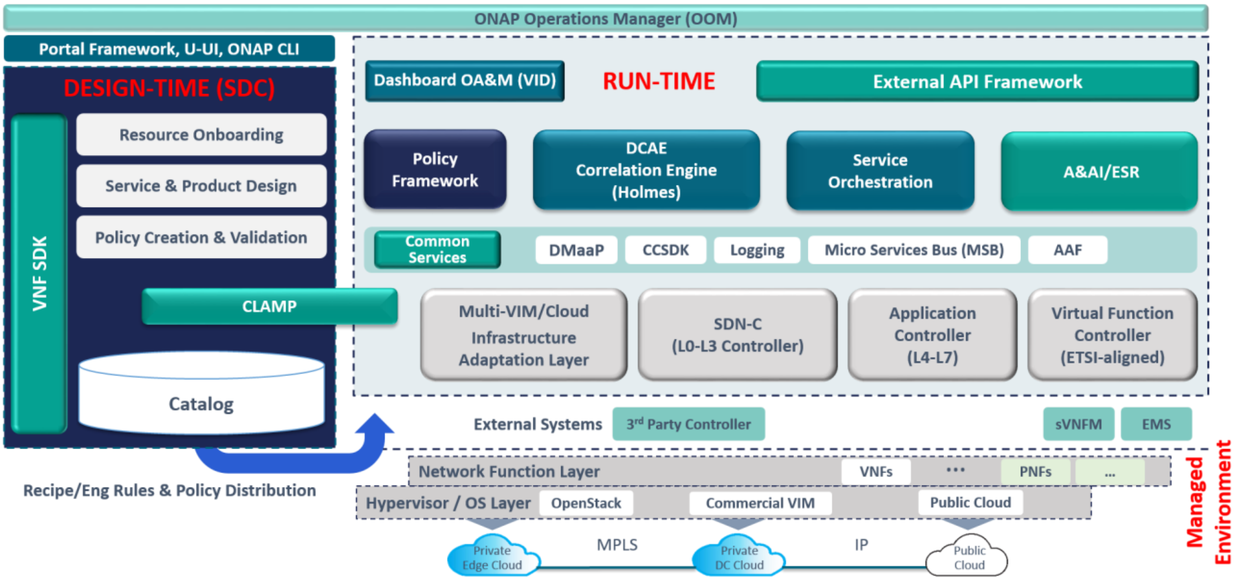


FIGURE 36: ARCHITECTURE OF THE ONAP PLATFORM, AMSTERDAM RELEASE

The Service Design and Creation (SDC) component provides instruments and repositories to define, simulate and certify system assets together with a set of processes and policies associated with particular assets. Other components, like Policy creation engine and VNF SDK provide necessary building blocks for service/product design and on-boarding.

The core blocks of the runtime framework are Service Orchestrator (SO), set of specialized Controllers, Data Collection, Analytics and Events (DCAE), Active and Available Inventory (A&AI) components together with a number of auxiliary services, like dashboard, API service and so on. These components jointly used to instantiate a particular service on top of the heterogeneous infrastructure, manage service lifecycle and expose run-time analytics regarding service state.

Amsterdam release of the ONAP platform is capable to demonstrate just two use cases:

- Voice over LTE: This consists of a set of proprietary VNFs and demonstrates the platform’s capability to deploy VoLTE services over SDN/NFV infrastructure. This is one of the use cases introduced in the Open-O project. However, due to the proprietary nature of the VNFs used in this scenario there is limited capability to rebuild this use case at a local environment.
- Residential vCPE: This demonstrates the applicability of the NFV concept to common network functions, like NAT, firewall, and parental controls and their implementation in a virtualized form. Primary ONAP components used for this use case are SDN-C, which enables interconnection among VNFs and APP-C,

which manages virtualization services. In this case, ONAP provides a common service orchestration layer for the end-to-end service provisioning and deployment. Open source VNFs and applications were used for this use case, thus technically it is possible to reproduce this scenario at a local environment.

In the next, Beijing release support for additional use cases is planned. Among them is the 5G use case which includes a number of subcases, like 5G slicing, 5G RAN Deployment, and 5G Optimization. At the moment of writing, this use case is still under development and its exact scope and deliverables will be clarified later upon their development.

Initial analysis shows that certain components of the ONAP v1 might be considered as a generic framework for a service orchestration engine of the 5G-TRANSFORMER platform, as multiple software components, like SO, Controllers, DCAE are already available and provide the required functionality. However, the overall amount of the resources required for platform deployment is huge, which makes the integration and extension process of this platform complicated.

13.1.3 Cloudify

Cloudify is the open source cloud orchestration software platform developed by Gigaspaces. This platform allows to model applications and services and automate their entire life cycle management, including service deployment on the top of VIM and WIM environments, monitoring of the deployed service or particular application, detecting issues and failures, manually or automatically remediating such issues. Originally introduced as a cloud/application deployment orchestration solution, the platform was further focused on the NFV domain.

Cloudify has an extensible architecture and can interact with numerous infrastructure providers, both public (AWS, Microsoft Azure, etc.) or private (OpenStack, Kubernetes, etc.). Overall, it is a quite mature platform widely adopted in production environments as an orchestration engine. At the moment of writing, version 4.2 of the Cloudify platform is considered as stable and it will be used as a reference in this review.

From the architecture perspective, Cloudify implements an asynchronous, agent-based approach to orchestrate the service deployment process. As a software platform, Cloudify integrates a number of open source components, like InfluxDB for time-series data storage or Riemann as a monitoring and alerting platform. The general platform architecture is presented in Figure 37.

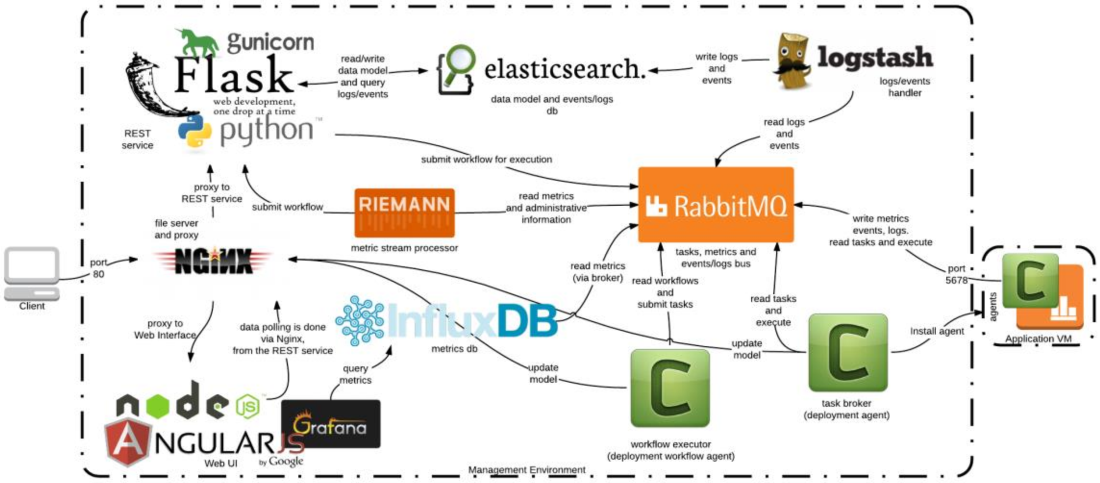

gunicorn
Flask
web development, one drop at a time
python
REST service
read/write data model and query logs/events
elasticsearch.
data model and events/logs db
write logs and events
logstash
logs/events handler
read logs and events
submit workflow for execution
proxy to REST service
submit workflow
RIEMANN
metric stream processor
read metrics and administrative information
RabbitMQ
tasks, metrics and events/logs bus
write metrics events, logs. read tasks and execute
file server and proxy
port 80
Client
NGINX
read metrics (via broker)
read workflows and submit tasks
read tasks and execute
port 5678
agents
Application VM
Install agent
update model
proxy to Web Interface
data polling is done via Nginx, from the REST service
InfluxDB
metrics db
query metrics
update model
task broker (deployment agent)
workflow executor (deployment workflow agent)
node JS
ANGULARJS by Google
Web UI
Grafana
Management Environment

FIGURE 37: CLOUDIFY ARCHITECTURE

To model the networking service or application deployment process, Cloudify uses a text template called a “Blueprint” which is effectively an orchestration plan. To describe blueprints, a special domain specific language (DSL) is used, which is based on a TOSCA [50] specification. These blueprints describe service topology as a set of nodes and associated resources, relationships and dependencies among nodes, their states and appropriate actions. Cloudify integrates the Apache ARIA library [51] to process TOSCA blueprints and execute sequence actions specified in a blueprint for a particular state.

From the perspective of the underlying infrastructure management, Cloudify heavily relies on a plugin mechanism, which specifies custom data types used for service modelling and used to implement appropriate actions, specified in these models. Multiple plugins are already publicly available for common infrastructure providers (AWS, OpenStack etc.) at the project github repository [52]. So, Cloudify has the required capabilities for Multi-VIM support and any further arbitrary VIM/WIM support might be added via a custom plugin.

As Cloudify is a quite mature platform, it can be considered as a candidate for 5G-TRANSFORMER’s generic service orchestration component. In the meantime, even if its TOSCA-based orchestration engine might be easily customised, full integration of this platform into the 5G-TRANSFORMER project requires multiple extensions and customisation in areas of the multidomain and federation specific use cases.

13.1.4 OpenBaton

OpenBaton [53] is an open source project, led by Fraunhofer FOKUS and released under the Apache 2.0 License, which implements an NFVO with an integrated and general-purpose VNFM. OpenBaton is compliant with the ETSI MANO specifications and workflows for VNF lifecycle management, even if not fully aligned with the ETSI GS NFV-IFA interfaces and information models, and it can be easily integrated with existing cloud platforms and adapted to different types of VNFs and Network Services. In particular, at the VIM level, OpenBaton supports the integration with (multi-site) OpenStack environments and it also provides an SDK to implement VIM-specific drivers. On the VNF side, it implements a generic VNFM, but it can also interoperate with external VNF-specific VNFM via REST APIs or AMQP and with a Juju-based VNFM. The interaction with the user is handled through REST APIs, a command-line interface (CLI) or a web-based dashboard.

In terms of features, OpenBaton supports (i) multi-tenancy at the infrastructure level, making use of SDN technologies for ensuring isolation between multiple network services sharing the same physical resources, (ii) auto-scaling and (iii) fault management based on monitoring information collected from the NFVI.

OpenBaton architecture (see Figure 38) is built around two main components: the NFVO and the Generic VNFM. The **NFVO** implements the main orchestrator features defined in the ETSI MANO. It keeps an overview of the underlying multi-site infrastructure, offering mechanisms to dynamically register several NFV Point of Presence (PoP) VIMs (e.g. several OpenStack instances). It also offers mechanisms for loading VNF packages (including VM images) and VNF descriptors, maintaining the associated catalogues. The VNF Descriptor templates are in JSON format and they are based on a detailed information model which includes most of the attributes defined in the ETSI MANO specification. The NFVO also offers mechanisms for on-demand deployment of VNFs based on the loaded descriptors, with the possibility to specify the VM placement and implement multi-tenancy, even though just in a limited scope.

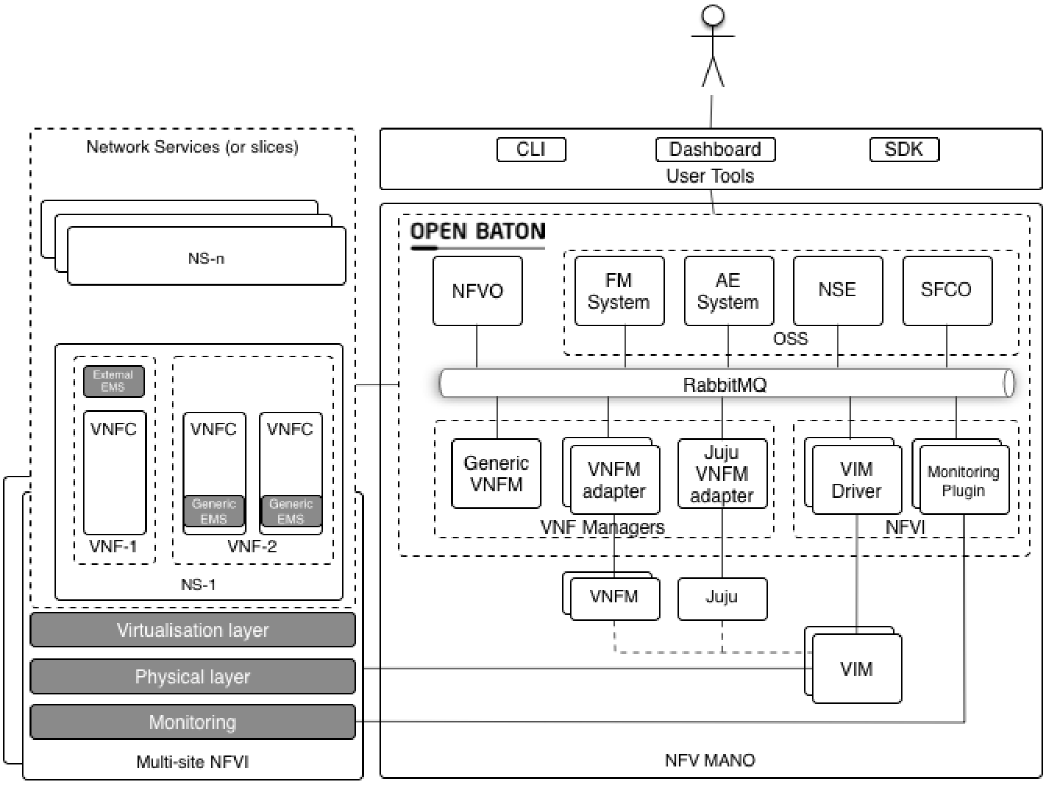


FIGURE 38: OPENBATON ARCHITECTURE

On the other hand, the **Generic VNFM** offers a basic support for the management of VNFs lifecycle and interacts with the NFVO and the VIM to request instantiation, modification, starting and stopping of VNFs. The interaction with a VNF is mediated through the generic OpenBaton Element Management System, that is a thin software application that runs on the VNF itself and enables a communication based on message queues. This interaction is mostly used for VNF configuration purposes and to trigger the execution of lifecycle management scripts on the VNF side.

13.1.5 OpenStack Tacker

Tacker[13] is a Generic VNF Manager (VNFM) and a NFV Orchestrator (NFVO) from the OpenStack project, allowing to deploy an end-to-end network service in terms of the decomposition of Virtual Network Functions (VNFs). Tacker is based on the ETSI MANO architecture framework to orchestrate an end-to-end network service. Figure 39 shows the Tacker architecture. From Figure 39, three main components inside Tacker can be identified: NFV catalogue, VNFM, and NFVO.

The NFV catalogue contains VNF Descriptors (VNFD), Network Service Descriptors (NSD), VNF Forwarding Graph Descriptors (VNFFGD) and a TOSCA Template Validation component. VNFD is a template to define the behavioral and deployment information of a VNF in Tacker. The template is based on TOSCA standards and uses a YAML model. NSD specifies the network services to be created while VNF FG can be regarded as Service Function Chaining (SFC) and Classifiers. SFC makes an ordered list of VNFs for data traffic to go through and the classifier describes which type of traffic should pass those VNFs.

The VNFM in Tacker includes Management and Monitoring Driver Frameworks, an Alarm Monitor, and an Event Audit Log. The Management Driver Framework relates to the basic lifecycle of VNF instances in terms of create, update and delete actions, and uses Enhanced Platform Awareness (EPA) for optimized VM placement, performance and operation. The Monitoring Driver Framework, Alarm Monitor and Event Audit Log allow for the development of customized modules for VNF-specific monitoring and, in turn, policy-based auto-healing and auto-scaling. Note that the initial configuration of VNFs, monitoring, self-healing and scaling are facilitated by TOSCA Workflows.

The NFVO has Multi-Site support to utilize multi-VIM installations with a unified view of resource control and management. Furthermore, the NFVO can analyze end-to-end network services by decomposing the network services into related VNFs. Using TOSCA Workflows, the NFVO can ensure efficient placement of VNFs based on a VNF placement policy. A VNF Forwarding Graph makes use of SFC to describe the connections among VNFs. Also, the NFVO can check and allocate multi-VIM resources. Network service instances and SFC can orchestrate VNFs across multi-VIM and multi-sites.

Based on the above introduction, we can summarize Tacker as having the following properties:

1) An end-to-end workflow management for the complete lifecycle of VNFs.
2) An API following the ETSI NFV-MANO specifications.
3) A framework for health monitoring and healing capabilities for VNFs.
4) Parameterized TOSCA VNF model.
5) Via plug-in drivers, it can retrieve monitoring data from and inject data to VNFs.
6) Initial configuration for VNFs.

[13] https://wiki.openstack.org/wiki/Tacker

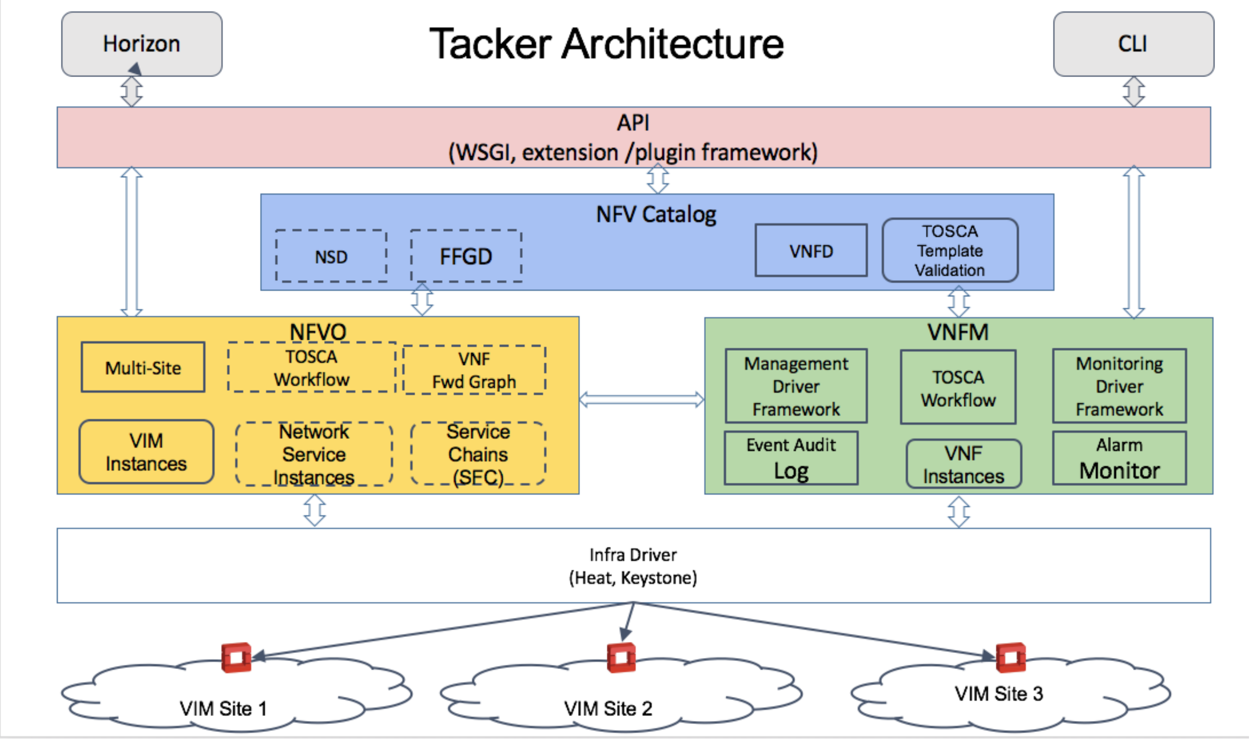


FIGURE 39: TACKER ARCHITECTURE

13.2 5G-PPP and FP7 Projects

A number of projects have been funded in the scope of EU R&D programs like FP7 and 5G-PPP. In this section we review several projects whose tangible outputs are relevant to the scope of the 5G-TRANSFORMER project.

13.2.1 T-NOVA

T-NOVA is a FP7 project that implements the migration of network functionalities, deployed as software, to virtualized network infrastructures developing NFV. Network operators can deploy VNFs and offer them to their customers as a value-added service by the implementation of an innovative framework. The project developed a MANO platform for the automated management of NFaaS over virtualized network infrastructures and the extension of SDN platforms to increase efficiency.

The formation of a NFV Marketplace allows developers to publish and trade network services as well as customers to select the services that cover their needs.

T-NOVA develops an orchestrator called TeNOR which complies with ETSI-NFV architecture and includes Network Services (NSs) and VNFs lifecycle management operations.

TeNOR interacts with the Marketplace, the VIM (for managing the datacenter network infrastructure resources) and with the WAN Infrastructure Connection Management (for the WAN elements connectivity management). TeNOR orchestrator also interfaces with the VNF itself to provide its lifecycle management.

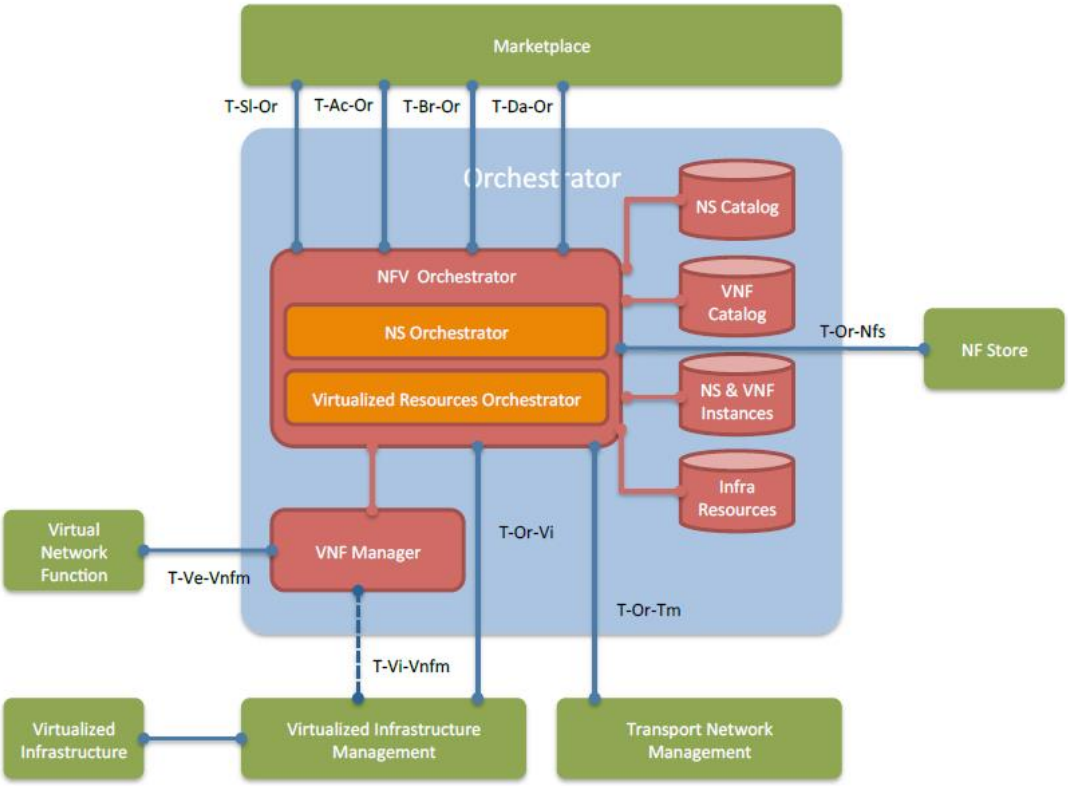


FIGURE 40: TENOR REFERENCE ARCHITECTURE AND INTERACTIONS

TeNOR is composed internally of two main elements: NFV Orchestrator and VNF Manager, as seen in Figure 40.

The NFV Orchestrator acts as the front-end with the Marketplace, orchestrates the incoming requests towards the other elements of the architecture and has the support of a set of repositories to describe the available resources, VNFs and NSs. The VNF Manager is responsible for the VNF-specific lifecycle management procedures.

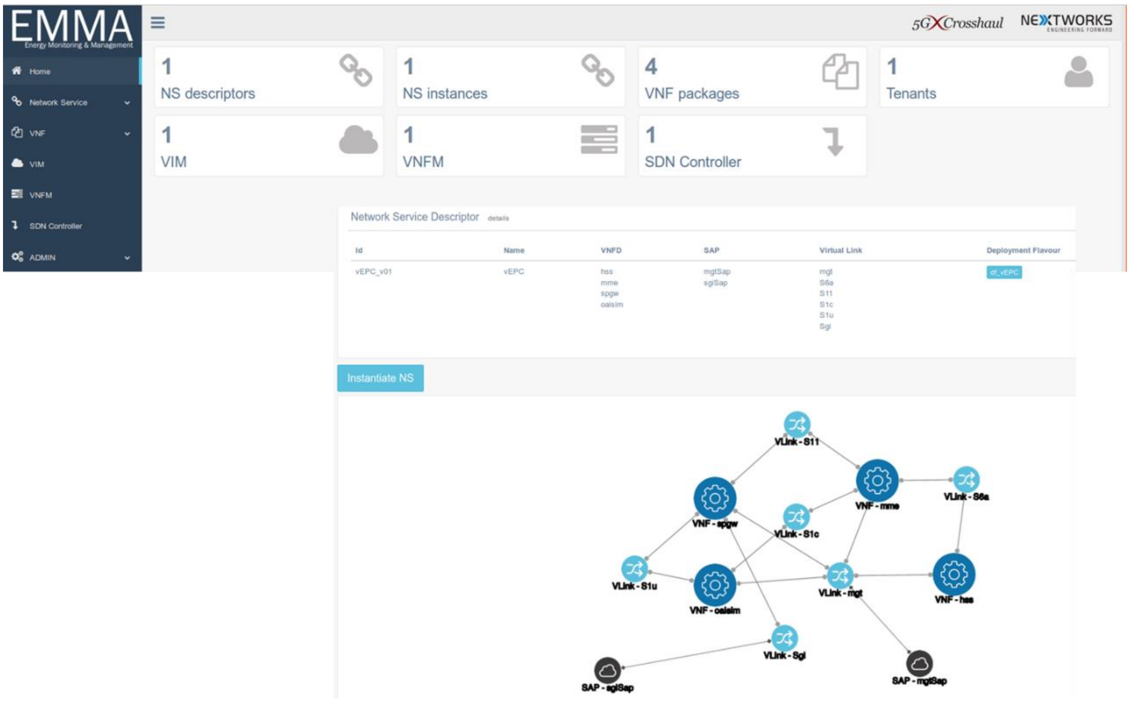
13.2.2 5G-Crosshaul

5G-Crosshaul has delivered an NFVO software prototype able to support the instantiation and termination of Network Services composed of VNFs over a unified Crosshaul network infrastructure including Crosshaul Processing Units (XPUs), i.e. computing nodes, and Crosshaul Packet Forwarding Elements (XPFEs), i.e. packet switches able to support PBB-TE based MAC-in-MAC Crosshaul Common Frames (XCF). In particular, this NFVO operates over an OpenStack-based VIM controlling the XPUs and configures the XPFEs that interconnect the XPUs interacting with an OpenDaylight-based SDN controller.

The features and functionalities currently supported by the NFVO developed in 5G-Crosshaul are the following:

- NS lifecycle management for Network Service instantiation (including cloud-init based configuration), termination, and post-init configuration.
- Resource orchestration with pluggable algorithms for joint computation of VNFs placement and network paths.
- Management of VNF packages and NSDs catalogues.

- REST APIs aligned with ETSI GS NFV-IFA information models (json format) for VNF packages (ETSI GS NFV-IFA 011), NSD (ETSI GS NFV-IFA 014) and Os-Ma-Nfvo (ETSI GS NFV-IFA 013) [22][23][24].
- Basic multi-tenancy support, with isolation between Network Service instances.
- Integration of an embedded VNFM specialized for OAI vEPC instantiation and configuration (the interface between NFVO and VNFM is a java interface based on ETSI GS NFV-IFA 007 [19]).
- Management of multiple VIMs through dedicated plugins, with a generalized interface based on ETSI GS NFV-IFA 006 [18]. The NFVO includes a plugin for OpenStack and a plugin for a dummy VIM (used for testing). However, Network Service instantiation across different VIMs is not yet supported.
- Integrated provisioning of underlying network connectivity for a QoS-enabled communication between VMs placed in different XPUs. The interaction with the SDN controller managing the XPFEs network (i.e. the WIM) is based on a plugin that exposes a proprietary java interface towards the NFVO, implementing methods for advertisement of the network topology, provisioning of end-to-end network paths (with and without explicit specification of the path hops) and power state management.
- Power consumption management through energy efficient resource allocation algorithms and automated adjustment of power states for XPUs and XPFEs.
- Web based GUI (Figure 41) for:
    - Management of tenants, VIMs, WIMs (SDN controllers), and VNFMs.
    - On-boarding and visualization of VNF packages and NSDs.
    - Instantiation, termination and visualization of Network Service instances.



Network Service Descriptor details

Id	Name	VNFD	SAP	Virtual Link	Deployment Flavour
vEPC_v01	vEPC	hss mme spgw oaisim	mgtSap sgiSap	mgt S6a S11 S1c S1u Sgi	df_vEPC

FIGURE 41: NFVO DEVELOPED IN 5G-CROSSHAUL – WEB GUI

The high-level software architecture of the 5G-Crosshaul NFVO is shown in Figure 42. The software is written in Java, using the Spring framework [54], adopts PostgreSQL [55] as SQL DB and RabbitMQ [56] as internal message broker.

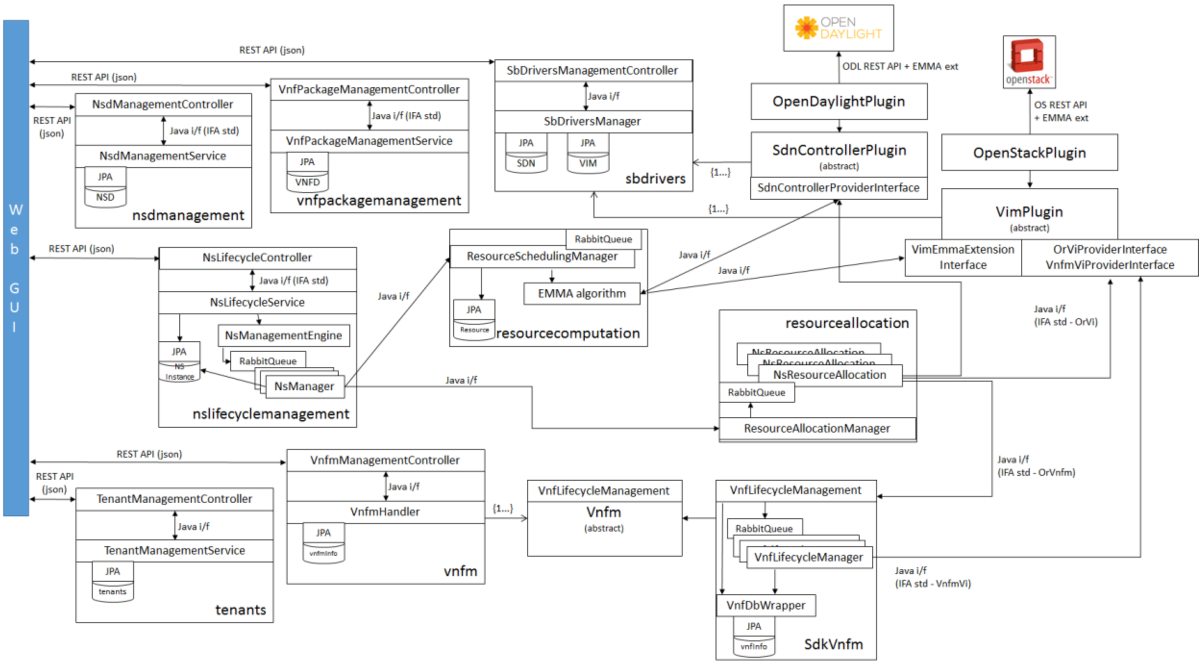


FIGURE 42: NFVO DEVELOPED IN 5G-CROSSHAUL – HIGH-LEVEL SOFTWARE ARCHITECTURE

In the following, the major components are listed:

- The **VNF package** and **NSD catalogues** are managed through the *nsdmanagement* and the *vnfpackagemanagement* packages, which expose REST APIs aligned with ETSI GS NFV-IFA 013 [23], adopting a JSON based encoding for VNFDs and NSDs.
- The **NSO functionalities**, i.e. the management of the lifecycle of Network Service instances, are implemented in the *nslifecyclemanagement* package, which offers a REST API (again based on ETSI GS NFV-IFA 013 [23]) for the instantiation and tear-down of network services. The lifecycle of each Network Service is handled through a finite state machine that evolves based on the external commands received through the REST APIs or the events coming from other components, e.g. the successful allocation of a Network Service instance resources or the correct configuration of its VNFs. Network services are persisted in an internal repository, in order to keep trace of their current status, their associated VNF instances and virtual resources.
- The **RO functionalities**, i.e. the coordination of the resource allocation on the VIM and WIM, as well as the instantiation of the VNFs through the VNFM, are implemented in the *resourceallocation* package, while the management of the pluggable resource allocation algorithms is handled in the *resourcecomputation* package. Also in this case, the resource allocation for each Network Service is handled through a finite state machine, which invokes commands on the south-bound plugins (i.e. the VIM plugin and the WIM plugin) or on the VNFM and

evolves based on events like the result of a resource allocation command on the VIM, the correct setup of a network path on the underlying XPFE domain or the result of the VNFs configuration from a VNFM.

13.2.3 SONATA

SONATA [57] is a 5G-PPP project that aims to increase the flexibility and programmability of 5G networks by reducing time to market for networked services shortening service development, optimizing resource utilization, accelerating the adoption of software networks in industry and standardizing project results.

The project develops an SDK written in the Python programming language that supports network developers in building network service and VNF descriptors by automatically validating them. These descriptors are then packaged and submitted to the Catalogue.

The Catalogue stores package files and packages, services and functions descriptors like code, configuration data and specific management requirements and preferences. There are three types of catalogues in SONATA: Private catalogues of service developers, Service platform catalogues and Public catalogues.

The responsible component for processing the different requests is a gatekeeper module in the Service Platform. The SDK of SONATA implements the service packages, and the Service Platform receives them and is responsible for managing them on existing cloud infrastructures. The Service Platform provides flexibility and control to operators and developers.

The system design of SONATA is based on the DevOps workflow involving continuous deployment and integration during service deployment. The Orchestrator of SONATA is an extension of the ETSI MANO including more flexibility and modularity to the MANO Framework by the addition of Service Specific Managers and Function Specific Managers, dividing lifecycle management operations and modifying the provided default managers to a specific service or function needs. The Gatekeeper controls the overall access to the Orchestrator and the use of OpenStack as VIM makes production monitoring data available to developers during limited time slots.

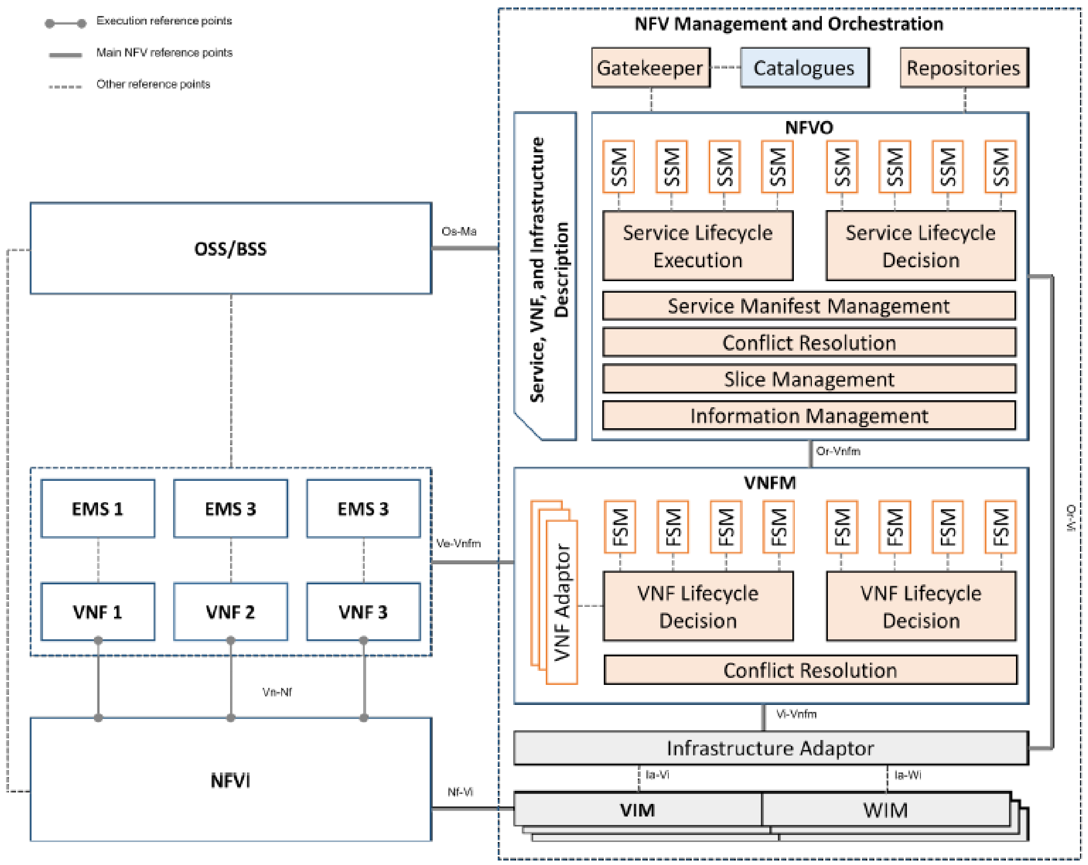


FIGURE 43: MAPPING FUNCTIONAL ARCHITECTURE OF THE SONATA SYSTEM TO ETSI REFERENCE ARCHITECTURE

Figure 43 shows the high level architecture of SONATA and how it maps into the ETSI model, the Service Specific Managers (SSMs) and Function Specific Managers (FSMs) point out one of the innovations of SONATA. At the NFVO level, a default manager for every network service is provided by the Orchestrator, as well as VNF at the VNFM level.

The Orchestrator provides a default manager for every network service, at the NFVO level and VNF, at the VNFM level, but allows this generic behavior to be adapted for each network service or VNF by their developers.

13.2.4 5G-TANGO

5G-TANGO [58] builds upon the SONATA project to improve and extend the MANO platform/service platform. It simplifies the creation of network service packages and descriptors providing an extensive SDK that supports the creation of network service descriptors and packages and even recursion in network services.

A main innovation is the Validation and Verification platform (V&V), a standalone platform that enables automated testing of VNFs and network services. The test results are implemented as digitally signed certificates and provide feedback for further improvements of the network service. They also allow comparing different versions and implementations of network services to select the most suitable ones.

Network services are managed and orchestrated by the service platform while abstracting the infrastructure details and enabling network slicing.

5G-TANGO introduces a complete DevOps approach and assists the progress of development, deployment, management and orchestration through the lifecycle of network services.

13.2.5 5G-EX

The goal of the 5G Exchange (5GEx) project is to enable cross-domain orchestration of services over multiple administrations or over multi-domain single administrations. This will allow end-to-end network and service elements to mix in multi-vendor, heterogeneous technology and resource environments thereby overcoming market fragmentation in a multitude of network operators each focused on different countries and regions. Indeed, 5GEx aims to enable collaboration between operators, regarding 5G infrastructure services, with the view to introducing unification via NFV/SDN compatible multi-domain orchestration by producing an open platform enabling cross-domain orchestration of services over these multiple domains, i.e., Multi-Domain Orchestrator (MdO).

The MdO acts as a key element for governing the orchestration of services and resources across different domains. As shown in Figure 44, the MdO coordinates resource and/or service orchestration at multi-domain level, where multi-domain may refer to multi-technology (orchestrating resources and/or services using multiple Domain Orchestrators) or multi-operator (orchestrating resources and/or services using Domain Orchestrators belonging to multiple administrative domains). The MdO interacts with Domain Orchestrators via I3 interface APIs to orchestrate resources and services within the same administrative domains. The MdO interacts with other MdOs via I2 interface APIs (business-to-business, B2B) to request and orchestrate resources and services across administrative domains. Finally, the MdO exposes on interface I1 service specification APIs (Business-to-Customer, B2C) that allow business customers to specify their requirements for a service. 5GEx scenario also considers third party MdO service providers, which does not own resource domains but operate a multi-domain orchestrator level to trade resources and services from other providers (the ones actually owning such resources) [66].

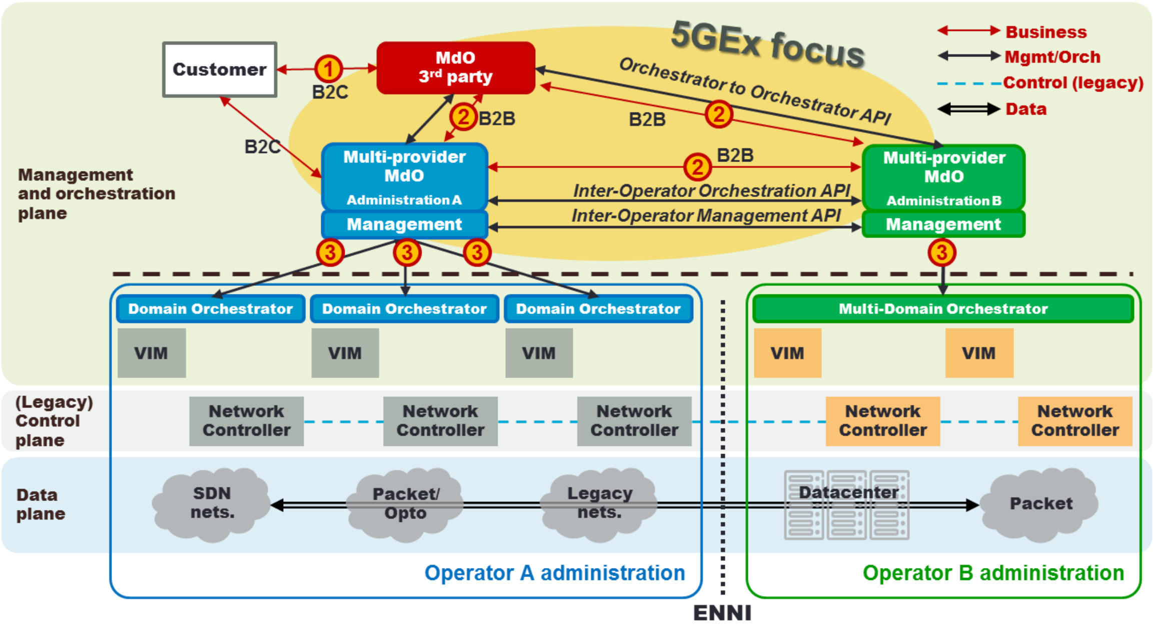


FIGURE 44: 5GEX REFERENCE ARCHITECTURAL FRAMEWORK

As an Innovation Action, 5GEx aims at delivering a Multi-domain Orchestrator (MdO) prototype that is the composition of T-NOVA NFV Marketplace offering Service Orchestration functions and the Unify ESCAPE Resource Orchestrator functions[14], both extended with multi-domain features. 5GT-SO leverages the general framework of orchestration of resources across multiple providers and domains from 5GEx. However, 5GT-SO extends the 5GEx MdO in the terms of more integrated and comprehensive support to vertical platforms. In this way, by exploiting the results of 5GEx, 5G-TRANSFORMER addresses both horizontal integration across domains as well as vertical integration thereby demonstrating the dynamic creation of slices per various vertical industries presenting different service requirements in real life scenarios. A joint standardization effort [5] has been started between 5G-TRANSFORMER and 5GEx, with the goal of specifying the main interfaces and protocols between domains. 5G-TRANSFORMER will specially leverage and extend the federation approach followed by 5GEx.

In terms of software components, 5GEx extended FP7 UNIFY project modules, with the goal of achieving proof-of-concept implementations, some of which have been released as Open Source components. Due to this nature, 5GEx components are not directly re-usable as such, but we plan to base on the concepts and develop some of them in 5G-TRANSFORMER, where we aim at contributing to main Open Source efforts in the area of NFV.

[14] ESCAPE (Extensible Service ChAin Prototyping Environment) is a general prototyping framework which supports the development of several parts of the service chaining architecture including VNF implementation, traffic steering, virtual network embedding, etc. (see [65])